User's Manual

Model 707741
WE Control API

YOKOGAWA
Yokogawa Meters & Instruments Corporation

IM 707741-61E
14th Edition

Thank you for purchasing the WE Control API (Model 707741).
This User's Manual contains information about the installation, programming model, and functions of the WE Control API. To ensure correct use, please read this manual thoroughly before operation.

Notes

- **The contents of this manual describe the WE Control API Ver. 5.2.0.0. If you are using another version of the API, the information given in this manual may differ from that of API that you are using.**
- The contents of this manual are subject to change without prior notice as a result of continuing improvements to the instrument's performance and functions.
- Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA dealer.
- Copying or reproducing any or all of the contents of this manual without YOKOGAWA's permission is strictly prohibited.

Trademarks

- Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are trademarks of Adobe Systems Incorporated.
- All other company and product names used in this manual are trademarks or registered trademarks of their respective companies.

Revisions

1st Edition: Febuary 1999
2nd Edition: September 1999
3rd Edition: March 2000
4th Edition: January 2001
5th Edition: April 2001
6th Edition: May 2001
7th Edition: October 2001
8th Edition: July 2002
9th Edition: September 2002
10th Edition: March 2003
11th Edition: September 2003
12th Edition: January 2004
13th Edition: July 2004
14th Edition: February 2007

14th Edition: February 2007 (YK)

How to Read This Document

How to read the “Parameter”

The words (IN) and (OUT) that follow the parameter indicate whether the parameter is an input parameter or an output parameter.

How to read the “Example (Visual Basic)”

Variable “ret”

Indicates a variable that contains the returned value (Dim ret As Integer).

“value → parameter value”

Indicates that the variable “value” is set to the parameter value.

Contents

1.1 Overview

This document covers the interface functions (WE Control API) that can be used to control the WE7000. The WE Control API is in the form of a dynamic-link library (DLL) for Microsoft Windows 95/98/Me, NT 4.0, 2000 Pro or XP Professional/Home Edition. Thus, it can be used under a WIN32 programming environment such as Microsoft Visual C++ and Microsoft Visual Basic.

The main features of the WE Control API are as follows:

Handle-based access
Just as with the file and window handles, a handle representing the connection to a station or module is obtained and used to control it. The independence of each station or module is realized in this manner.

Use of Plug&Play functionality
The Plug&Play functionality that is achieved by the control software can be utilized through the API, simplifying the programming work.

Displays the WE Control Software module operation panel
The operation panels and trigger setting dialog boxes for each module, that the WE Control Software displays, can be called through the API. This interface can be used to easily confirm setup information or make changes to the setup.

Independent from the communication medium
The WE7000 currently supports optical communication, Ethernet (100BASE-T), and serial communication and is expected to support other communication media as well in the future. By absorbing the differences between these different media within the API, the application program is independent of the communication media.

Supports Microsoft Windows
The API supports the most popular operating systems on the PC today, Microsoft Windows 95/98/Me, NT 4.0, 2000 Pro and XP Professional/Home Edition.

Additional files

File name	Description
WeAscii.bas	WE Control API definition file for Visual Basic
WeDP.bas	API definition file for file operation functions for Visual Basic
WeEvent.ocx	Control used to process asynchronous messages (events)
*.dll	DLL files for WE control API
WeAscii.h	WE Control API definition file (C++ include file)
WeDP.h	API definition file for file operation functions (C++ include file)
WeAPI.pdf	WE Control API Online Manual (Adobe Acrobat Reader 3.0 is needed to open this file.)
WeAPINet.pdf	WE Control API for .Net Online Manual (Adobe Acrobat Reader 3.0 or later is needed to open this file.)
Sample programs	Sample programs for various modules (included in the Samples folder)

The definition files contain symbols, data structures, and etc. that are used by the API. Make sure to include the appropriate file to the user application program.

With the exception of the DLL files listed above, all files are copied to the "C:\ProgramFiles\WE7000\API" directory as default. The DLL files are copied to the following directories.
For Windows 95/98/Me: Windows\System
For Windows NT/2000 Pro: Winnt\System32
For Windows XP Professional/Home Edition: Windows\System32

OS

Microsoft Windows 95/98/Me, NT 4.0, 2000 Pro or Windows XP Professional/Home Edition

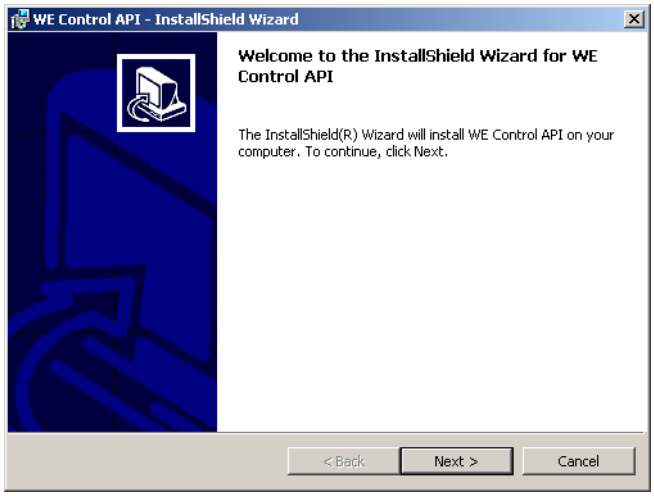
Development environment

Microsoft Visual Basic 5.0, 6.0, and .Net
Microsoft Visual C++ 4.1, 5.0, 6.0, and .Net
Microsoft Visual C# .Net

2.1 Installation

This chapter describes the installation procedures for the WE Control API. The WE7000 Control Software must be installed before installing the WE Control API.

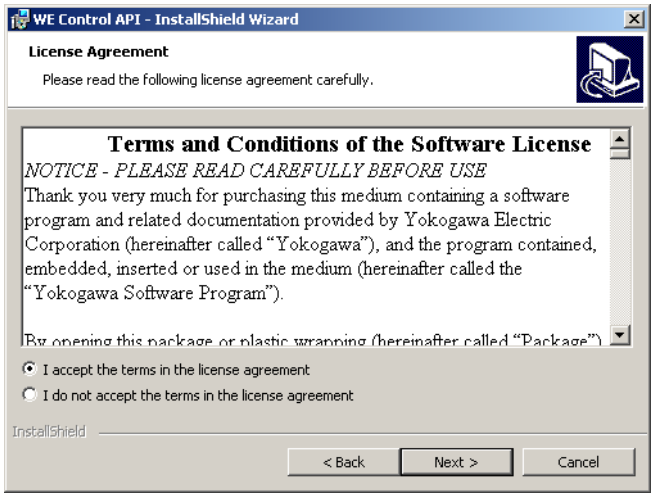
1. Start Windows.
2. Insert the WE Control API Setup Disk into your PC's CD-ROM drive. An installer automatically starts and the following dialog box opens. Click [Next].



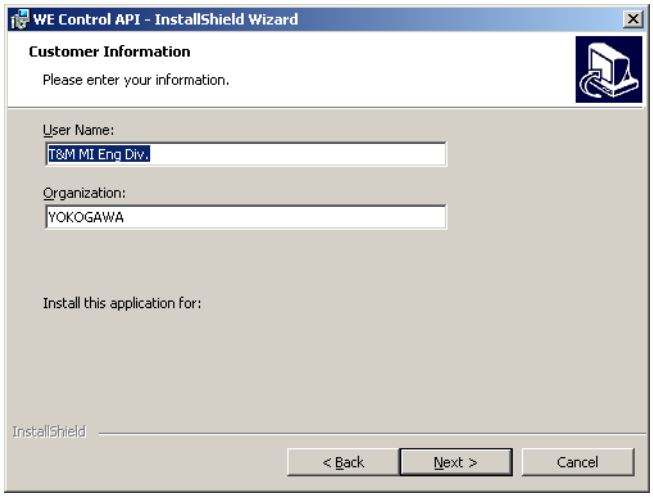
Note

If you attempt to install the WE Control API when the WE7000 Control Software is not installed, the following dialog box appears. Install the WE7000 Control Software first, and then install the WE Control API.

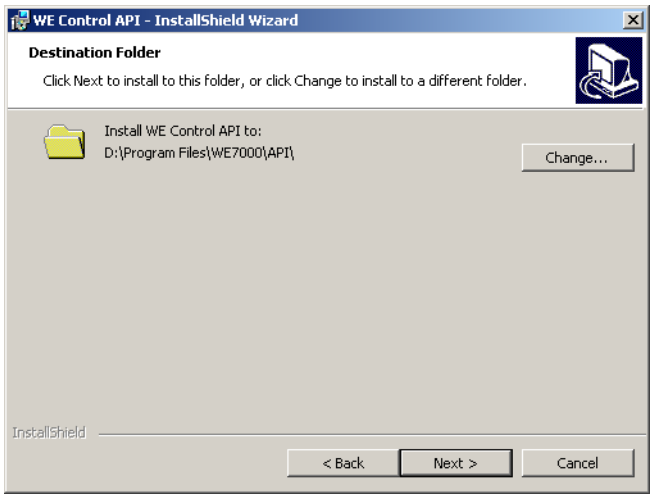
3. The following dialog box appears containing license agreement information. Confirm the license agreement, click the [I accept the terms in the license agreement] option button, and click [Next].



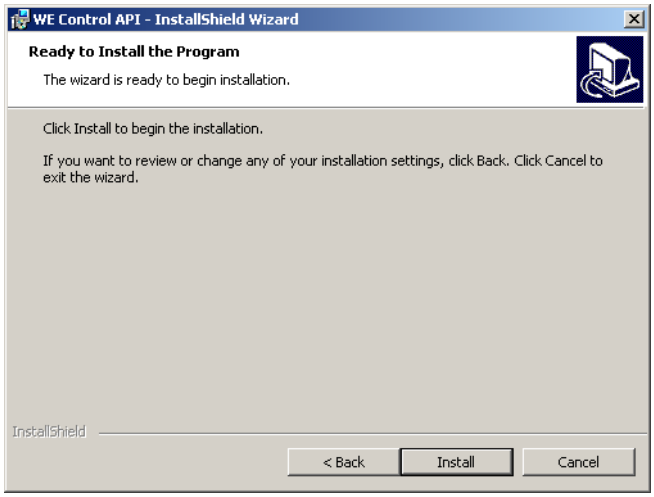
4. The following dialog box appears for registering the name and the organization of the user. After entering the appropriate information into each box, click [Next].



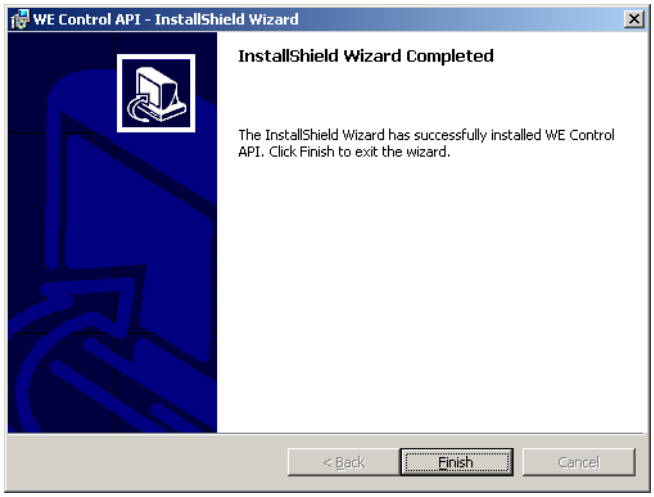
5. Specify the installation destination. The default installation destination is set to “C:\Program Files\WE7000\API\.” If this is OK, click [Next]. To change the installation directory, click [Change], select the directory, and click [Next].



6. A dialog box appears for you to confirm the start of the installation. Click [Install].
   The installation starts and a dialog box appears indicating the progress of the installation.



7. A dialog box appears notifying you that the installation has been completed. Click [Finish].



3.1 Basic Model

Handle

The WE7000 Control API is a handle-based API. Generally, when accessing a file, a program opens the file by specifying the file name and then uses the file handle that is returned for reading and writing the data. The WE7000 Control API is used in a similar way. The program opens (connects to) the station or module by specifying its name and then uses the handle that is returned to get information and to set up and execute measurements.

The handle of the Control API has the following features that simplify the access to the station or module.

- One handle can correspond to multiple stations. As a result, one function can be called to access multiple stations.
- There is a broadcast handle that can be used to access all stations.
- One handle can correspond to multiple modules of same type existing at different stations. As a result, one function can be called to set and operate multiple modules in the same way.
- Since the handle is obtained by specifying the name, it is unaffected by the configuration of the measurement system such as the position of the station on the network or the slot position of the module. This improves the maintainability of the program.
- One handle can be used to control multiple modules of the same type that are linked.

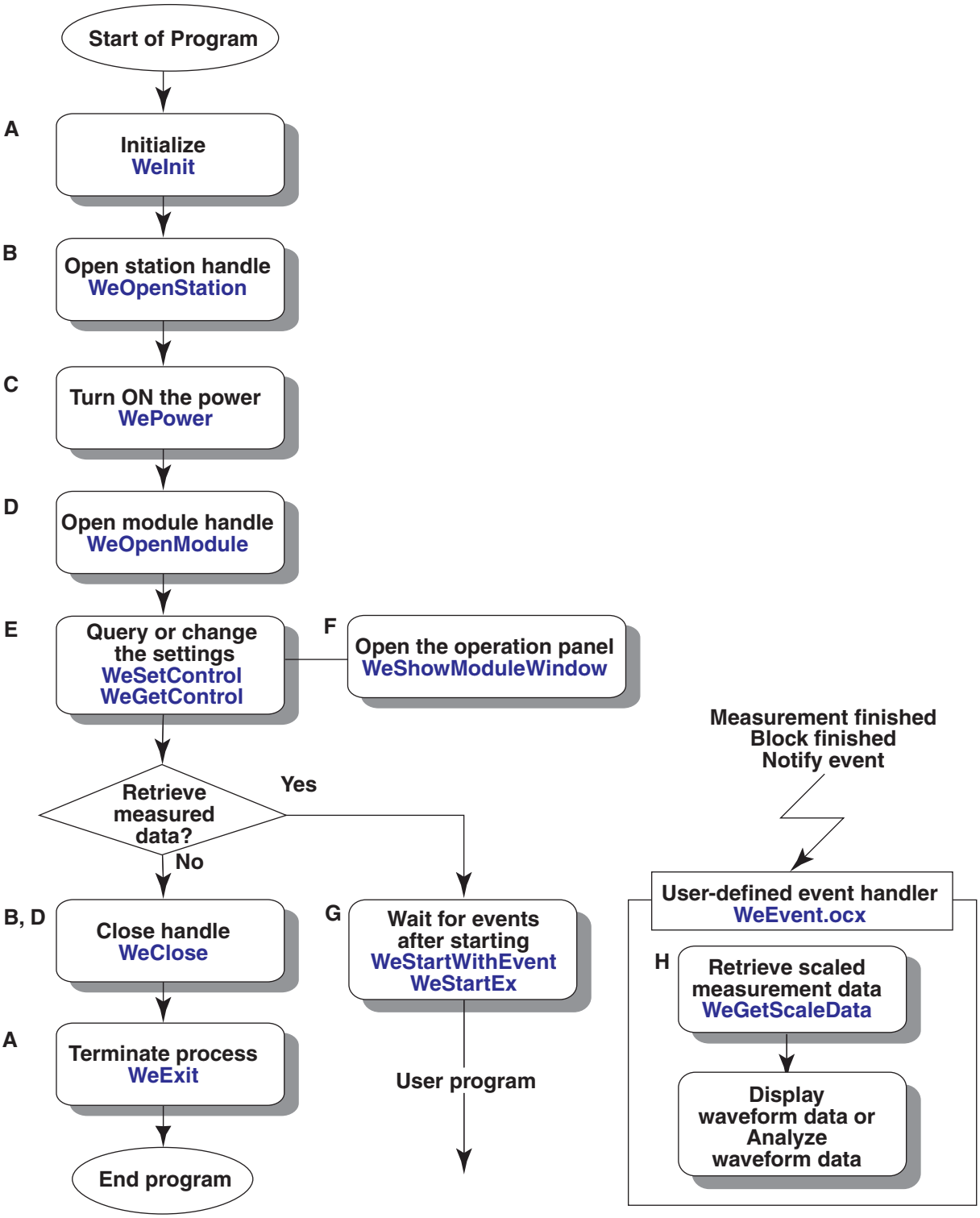
Asynchronous Messages (WE Event)

To improve the performance of the application program, the API has a function in which the stations and modules can be controlled asynchronously. This function makes use of the Windows Messaging. When an event occurs, the function sends a message that is defined by the WE7000 to the user application. Thus, the user can create event-driven programs.

The events include end of data acquisition, end of the specified number of data points of acquisition, alarm detection, stopping of the station fan, and other events that are defined for each type of module. In addition, to make the handling of these messages easier, OCX for Visual Basic is provided.
For details, see **chapter 5**.

3.2 Basic Flow of a Program

In a general application program for the WE7000, the program and network environment are initialized first. Then, the handle of the station and module that you wish to control is obtained. The following flow chart indicates the basic flow of the program when asynchronous messages are used. For information on measurement control, see also **chapter 4**.



A. Initialize **WeInit**, Terminate **WeExit**
Call WeInit to initialize the API environment and to automatically detect the station. The call is also used to select the communication interface and register the event handler. Be sure to call WeExit for every call to WeInit.

B. Open station **WeOpenStation**, close **WeCloseHandle**
Call WeOpenStation by specifying the desired station name. Be sure to call WeCloseHandle for every call to WeOpenStation. Another API, WeGetStationList, that can be used to read the list of names of all stations that are connected, is also provided

C. Turn On standby power **WePower**
Using the station handle obtained in step B, call WePower to turn ON the standby power switch of the desired station.

D. Open module **WeOpenModule**, close **WeCloseHandle**
Using the station handle obtained in step B, call WeOpenModule to select the measurement module. You can specify the module by the slot position or by the module name. For modules of same type, linked modules can be specified. Be sure to call WeCloseHandle for every call to WeOpenModule. The measuring station's standby power switch must be turned ON to call this API.

E. Set measurement conditions **WeSetControl**, query **WeGetControl**
Using the module handle obtained in step D, the measurement conditions of the module can be changed or queried. ASCII commands are provided for each module (see the section on ASCII Commands).

F. Open operation panel **WeShowModuleWindow**
An operation panel similar to that of the WE Control Software can be opened. This operation panel can be used to change the measurement conditions of the module as in Step E. Call WeCloseModuleWindow to close the operation panel.

G. Wait for an event after start **WeStartEx** (**WeStopEx**) or **WeStartWithEvent**
Call WeStartEx or WeStartWithEvent to start the data acquisition operation. The block complete or data acquisition complete event is notified. Be sure to call WeStopEx for every call to WeStartEx. The following additional APIs are provided for retrieving the measured data.
- Poll to see whether the measured data have been acquired — **WeStart** and **WeIsRun**
- Wait for the measured data to be acquired — **WeStartSingle**

For details, see the next chapter “Data Acquisition Model.”

H. Retrieve measured data that have been scaled **WeGetScaleData**
Retrieves the measured data that is scaled.
The APIs indicated on the next page are provided for retrieving the measured data acquired in the module and saving the acquired data to files.
When retrieving the waveform information, use the WeGetWaveInfo API.

Data Type	Retrieve Instantaneous Data	Retrieve Block Data	Save Block Data (Binary (wvf) format)	Save Block Data (ASCII (csv) format)
Raw data	–	WeGetAcqData, WeGetAcqDataEx	WeSaveAcqData	–
Physical values*1	WeGetCurrentData	*4	*5	WeSaveAsciiData
Scaled values*2	WeGetScaleCurrentData	WeGetScaleData	WeSaveScaleData	WeSaveScaleAsciiData
Scaled values*3	WeGetScaleCurrentDataEx	WeGetScaleDataEx	WeSaveScaleDataEx	WeSaveScaleAsciiDataEx

*1 Physical values are values in the unit of measurement of the module, not the scaled values. For example, it is voltage for the digitizer module.
*2 Data obtained by scaling the acquired data by specifying the scaling parameter through the API parameter.
*3 Data obtained by scaling the acquired data by specifying the scaling parameter through the scale conversion window or the WeSetScaleInfo API.
*4 Can be achieved by setting “a” and “b” of “ax+b,” the scaling parameters of WeGetScaleData, to “1” and “0,” respectively.
*5 Can be achieved by setting “a” and “b” of “ax+b,” the scaling parameters of WeSaveScaleData, to “1” and “0,” respectively.

4.1 Parameters Required for Data Acquisition

In general, three parameters (number of blocks, data size per block, number of acquisitions) must be specified to acquire data using the data acquisition module such as a digital oscilloscope module or a digitizer module, as described below:

Number of blocks

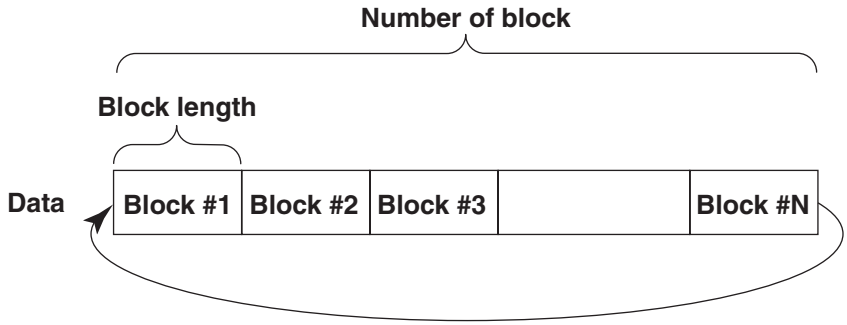
The acquisition memory (specific memory provided for acquiring measured data) can be divided into multiple blocks so that multiple instances of phenomena relating to the measured item can be acquired. The number of blocks specifies the number of partitions. The maximum number of blocks is equal to "total record length of acquisition memory/data size per block." The record length and the selectable data length per block vary depending on the type of module. For details, see section "Acquisition Restrictions" in the ASCII command description for each module in chapter 8, "ASCII Commands."

Data size per block

The maximum value is equal to "total record length of acquisition memory/number of blocks." In the trigger/gate (level) mode, you must also specify the record length indicating the number of data points that will actually be acquired.

Number of acquisitions

This number specifies how many times to acquire the data before stopping the measurement. Specifying zero for this number is equivalent to specifying infinity. The data acquisition continues until the user stops the operation. In other words, for each data acquisition, the "data size per block" of data are acquired. This action is repeated the number of times specified by the "number of acquisitions," after which the acquisition sequence stops. As shown in the figure below, if the number of blocks is less than the number of acquisitions, the oldest data block is overwritten, in a circular fashion, once all blocks have been used.



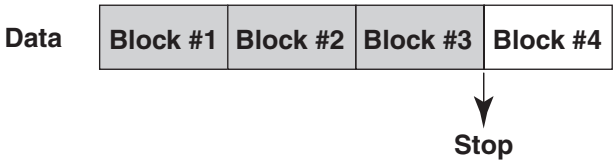
When the number of acquisitions > the number of blocks
or
When the number of acquisition is infinity

There are two ways to retrieve the acquired data using the WE Control API:

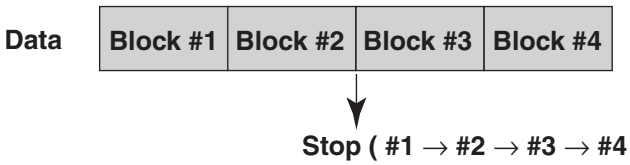
1. Retrieve the data after all acquisitions are finished (after the number of acquisitions of data has been acquired).

   In this case, specifying zero for the number of acquisitions (infinite acquisitions) would be useless.

- **When the number of acquisitions ≤ the number of blocks**
  For example, if the number of acquisitions = 3, the number of blocks = 4



- **When the number of acquisitions > the number of blocks**
  For example, if the number of acquisitions = 6, the number of blocks = 4



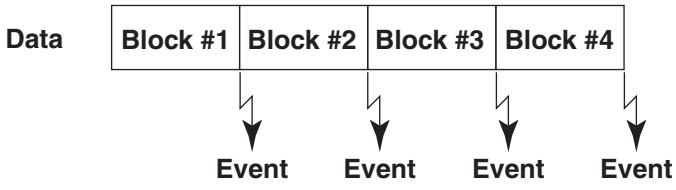
In this case, the following three methods can be used to detect the sequence stop condition.

- Poll to see if the acquisition sequence has stopped (use WeStart and WelsRun).
- Use a synchronous API function that does not complete (return) until the acquisition sequence stops (use WeStartSingle). See section 4.3.

- Use the WeStartWithEvent function that triggers an event when the acquisition sequence stops or use WeStartEx and set the fifth parameter, acquisition operation mode, to WE_ACQ_STOP_EVENT (generate events after acquiring all blocks and the operation stops). See section 4.3.

2. Using the event function, retrieve the data every time a block of data is acquired.

   In this case, continuous measurement becomes possible and the number of acquisitions becomes meaningless. In short, the data acquisition operation continues until the stop command is issued.



To use the event function, use the WeStartEx API and set the fifth parameter, acquisition operation mode, to WE_ACQ_BLOCK_EVENT (Generate events for each block). See section 4.3.

Note

The combination of WeCreateEvent, WeStart, and WeSetControl can be used to achieve the same effects as the WeStartEx API.

4.2 Specifying the Data Block that You Wish to Retrieve

To retrieve the data, specify the data block that you wish to retrieve. There are two ways to specify the block, relatively and absolutely, as described below (a figure is provided on the next page.):

Relative specification

The block is specified as follows: The newest block is –1, the next newest is –2, the next one after that is –3, and so on.

Absolute specification

The block is specified by the acquisition count. In other words, the first acquisition is 0, the next acquisition is 1, and the acquisition after that is 2, and so on. If the acquisition parameters are specified so that the number of blocks is less than the number of acquisitions, the valid block numbers are from (the current acquisition number – the number of blocks + 1) to the current acquisition number. Specifying any other block returns the following error, "Data does not exist," because the data block has been overwritten.

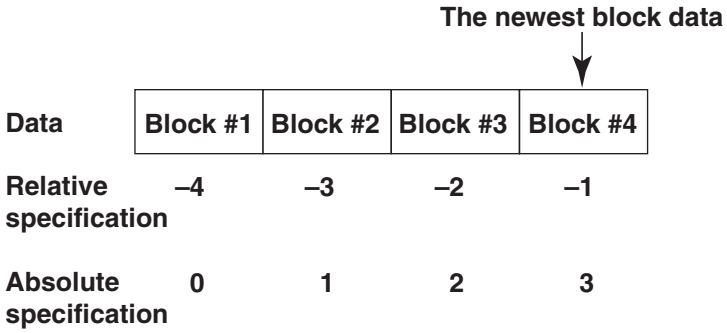
Note

If the block size is small and the sampling rate is high, the process of retrieving the data cannot keep up with the high frequency of events that occur, which may cause the acquired data to be overwritten (data acquisition overrun). If this occurs, the acquisition sequence automatically stops. The WeSetOverRun API is provided so that operation at the time of an overrun detection (stop/not stop the acquisition sequence) can be switched. For details, see chapter 6, "Details of Functions."

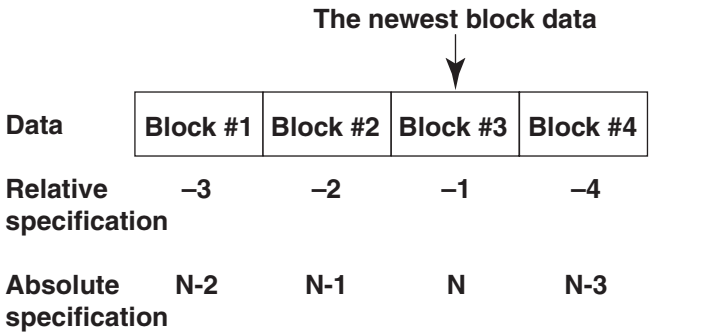
The following measures can be taken to prevent data acquisition overrun.

- Lower the sampling rate.
- Make the data length per block longer.
- Lower the trigger generation frequency.

- **When the number of acquisitions ≤ the number of blocks**
  For example, if the number of blocks = 4



- **When the number of acquisitions > the number of blocks**
  For example, if the number of blocks = 4, the newest acquisition count = N (with zero as the origin)



Note

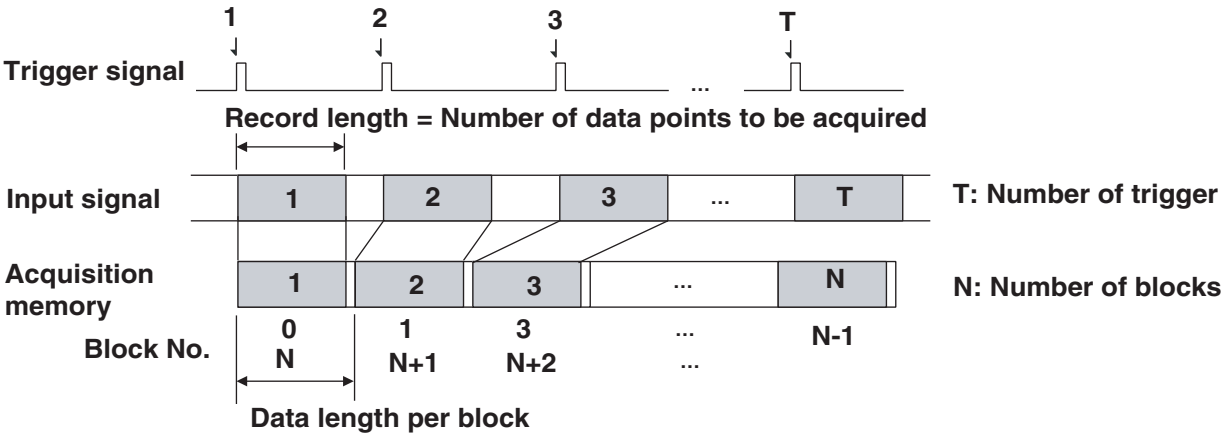
When using the event function to retrieve the block data, the block data corresponding to the event can be retrieved by specifying the block number relatively with a -1. **However, if the event indicating the end of the next data block occurs before the current data block is retrieved, the newest data block is actually retrieved; the data block no longer corresponds to the correct event in this case** (depends on the block size, sampling rate, and data retrieval interval). In this case, specify the block number using absolute values (the acquisition count). The acquisition count can be obtained from the parameter of the block end event handler.

4.3 Relationship between the Acquisition Mode and the Data Collection Method

The WE7000 data acquisition modules have a maximum of four data acquisition modes (methods): trigger, free run, gate (level), and gate (edge). This section describes the data acquisition operation of each mode, and the flow of operations when the WE control API is used.

Trigger mode

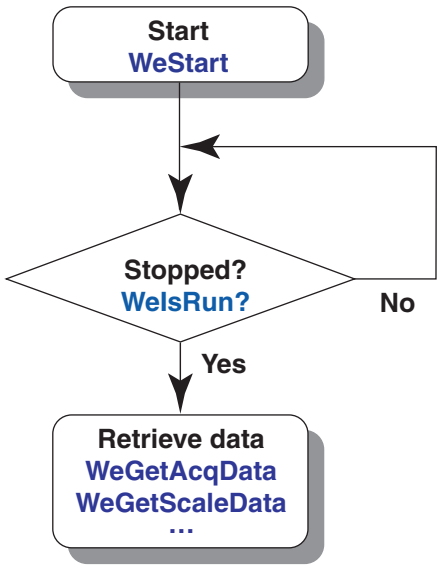
As shown in the figure below, the acquisition memory is partitioned, and measured data are acquired to each memory block every time a trigger occurs. This is exactly the same as the acquisition model that was described in the earlier section.



For the trigger mode, the following data collection methods are available.

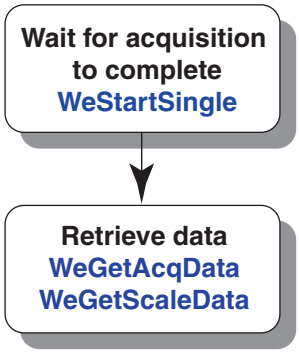
1. By polling

After starting the data acquisition operation (with WeStart), the program polls to see whether or not the data acquisition has been completed (with WelsRun). When the trigger occurs and the data acquisition is complete (the second parameter of WelsRun becomes zero (stop condition)), requests for measured data, waveform information, and other information (WeGetAcqData and WeGetAcqDataInfo) become effective. This method is advantageous in that the program can be written in a sequential fashion, making it easy to understand. In the trigger mode, the module automatically stops the data acquisition operation when it is complete. Thus, there is no need to call an API to stop the measurement (WeStop).



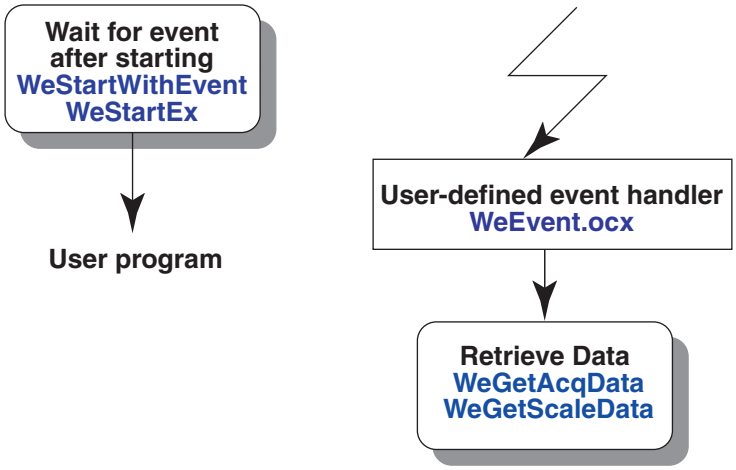
2. By waiting for data acquisition to complete

In this method, the program starts the data acquisition and waits until it is complete (WeStartSingle). When the trigger occurs and the data acquisition is complete, the program returns from the API. At this point, requests for measured data, waveform information, and other information (WeGetAcqData and WeGetAcqDataInfo) become effective. When the data acquisition does not terminate, the program returns from the API after a timeout. This method is advantageous in that the program can be written in a sequential fashion, making it easy to understand. As with the first method, in the trigger mode, the module automatically stops the data acquisition operation when it is complete. Thus, there is no need to call an API to stop the measurement (WeStop).



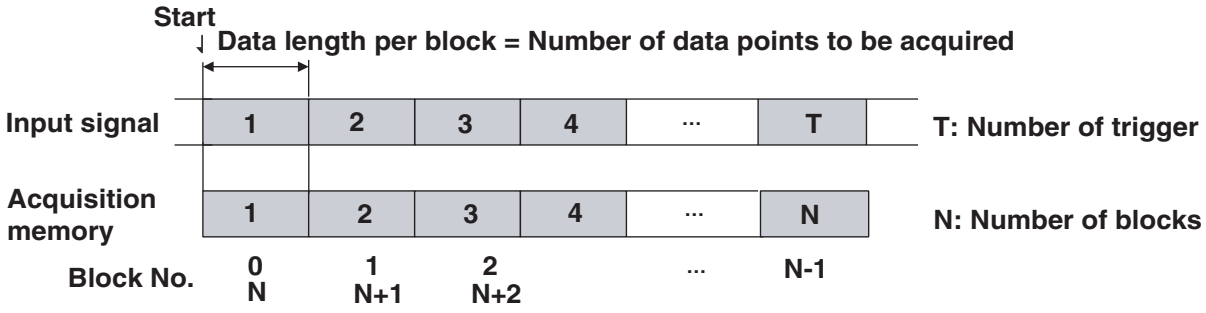
3. By event notification

After starting the data acquisition operation, an event occurs notifying the end of the block or the completion of the data acquisition operation. Using the acquisition model described earlier, three parameters are specified. To facilitate event handling, WeEvent.ocx is provided. In this method, the completion of the data acquisition operation can be detected clearly and other processes can be performed while waiting for the event (asynchronous programming). WeStopEx must be called to clear the event for every call to WeStartEx.



Free run mode

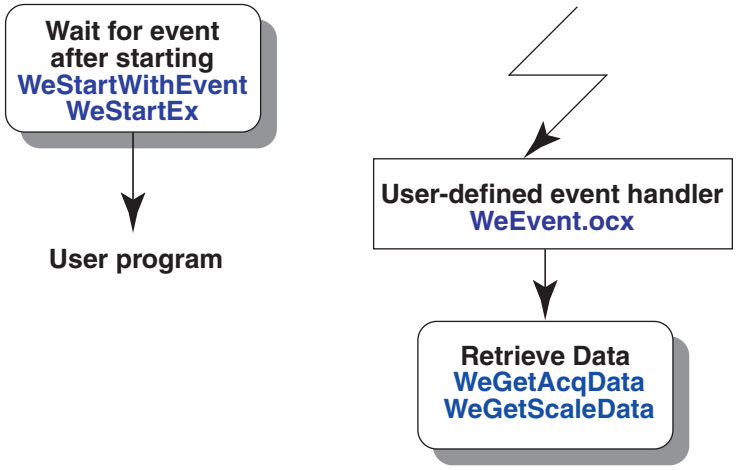
As shown in the figure below, measured data are acquired continuously according to the specified sampling rate (sampling interval) after the acquisition is started. Triggers are not detected in this mode. Because the acquisition memory is used in a circular fashion, measured data are acquired until the operation is stopped.



For the free run mode, the following two data collection methods are available.

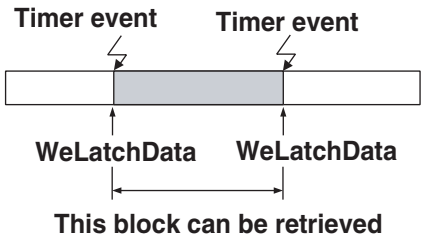
1. By event notification

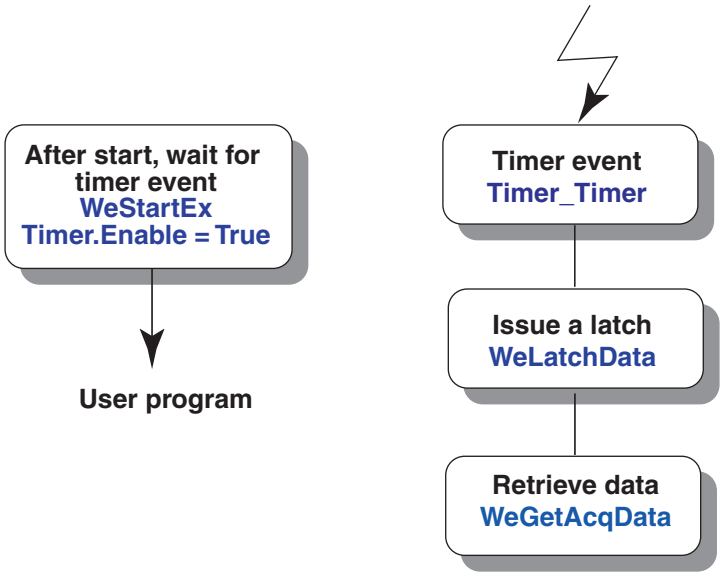
Measured data are acquired using the WeStartEx API by specifying the three parameters that were explained in section 4.1 (the block number is ignored). The blocks, in this case, are logical blocks just as in the latch operation. When retreaving the measured data, absolute or relative values can be used to specify the blocks. After the acquisition has stopped, you can specify "0x80000000" for the block number in order to retrieve all the data from the start of the acquisition to the end.



2. By latching at arbitrary times

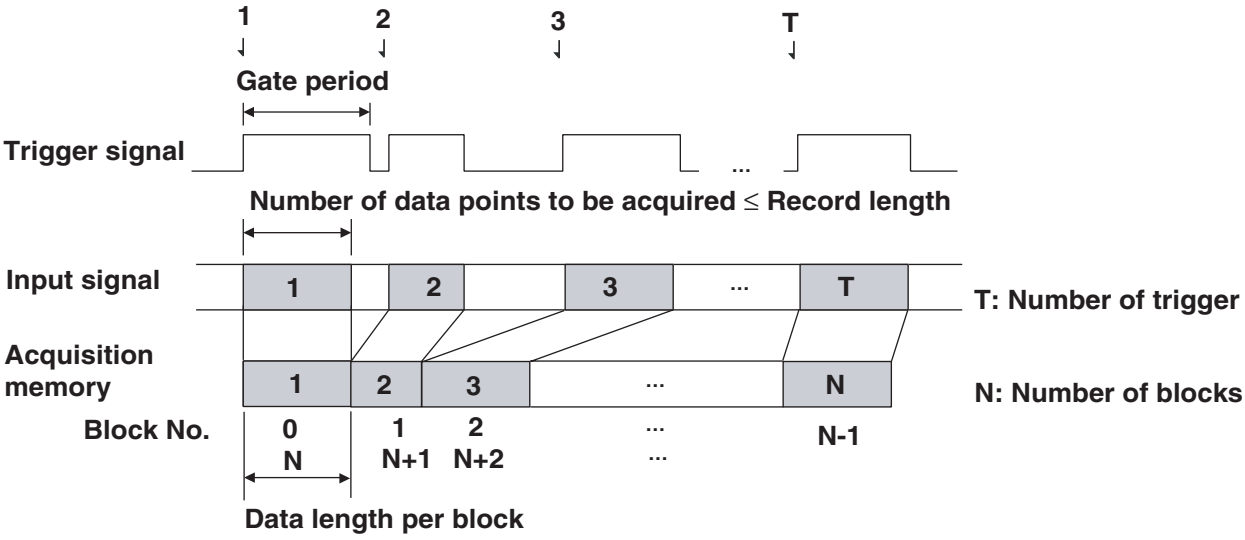
Latch operation is an operation in which a logical block is assigned to an array of data hat has been acquired continuously. One data block is considered to be the data between the previous latch operation and the current latch operation. This block can be retrieved. However, the only valid logical block is the newest block. Thus, block numbers are ignored.





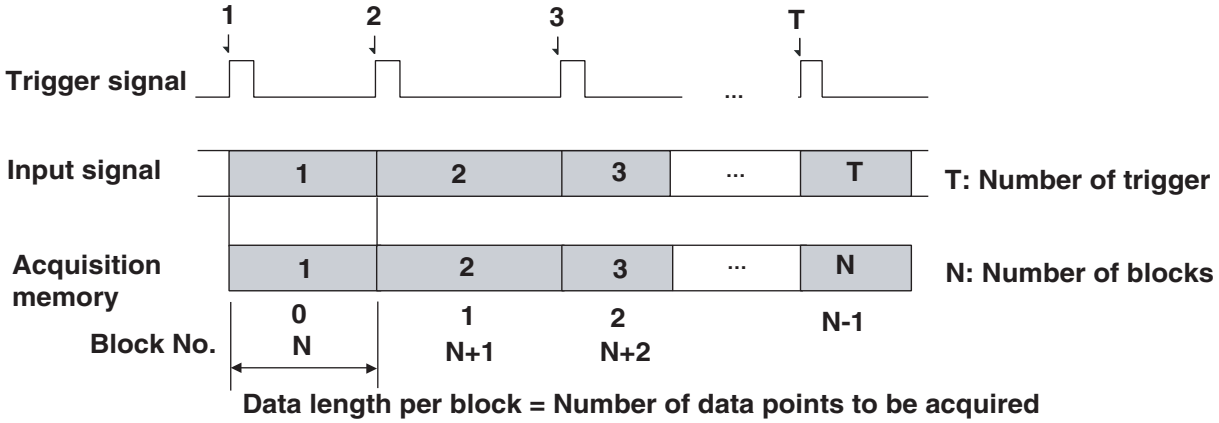
Gate (level) mode

As shown in the figure below, measured data are acquired only during the gate period. When the specified number of measured data are acquired before reaching the end of the gate period, the acquisition stops at that point. The data collection method in the gate (level) mode is the same as in the trigger mode. However, the number of blocks described in section 4.1 is void.



Gate (edge) mode

As shown in the figure on the next page, data acquisition starts when the specified trigger condition is satisfied and pauses when the trigger condition is no longer satisfied. When the trigger condition is satisfied again, data acquisition starts at a new memory block. The operation repeats the number of times specified by the number of acquisitions. However, the number of blocks and the record length described in section 4.1 are void.



5.1 About Events

To improve the performance of a user application, the WE Control API uses the Windows Messaging function to notify of phenomena occurring on the WE7000 as event. Currently, the following events are defined:

Events generated by the station

Remote power switching event	WE_EV_POWER
Event when the cooling fan stops	WE_EV_FANSTOP
Event when the DIO terminals change	WE_EV_DIO0 to WE_EV_DIO3

Events generated by a module that are common to all modules

The following events are available. However, not all the events are valid on all modules. For details regarding the validity of the events on each module, see section “Valid Common Events” for each module in chapter 8, “ASCII Commands.”

Event when the acquisition is complete and the module is in a stopped condition	WE_EV_MEASEND
Event when a block data has been acquired	WE_EV_BLOCKEND
Event when the measurement is aborted	WE_EV_MEASABORT
Event when a level trigger occurs	WE_EV_TRIG_HIGH WE_EV_TRIG_LOW
Event when a pulse trigger occurs	WE_EV_TRIG_PULSE
Even when the module's operation panel is closed	WE_EV_CLOSE_MODULE_GUI
Event when an alarm occurred/released	WE_EV_ALARM

Module-specific events

For details regarding these events, see section “Module-specific Events” for each module in chapter 8, “ASCII Commands.”

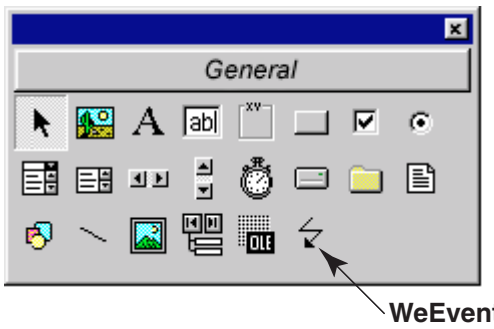
By using this capability, an event-driven program that effectively uses the Windows programming environment can be created allowing asynchronous control of the stations and modules.

In addition, the WE Control API provides an OCX file (WeEvent.ocx) to easily handled events using Microsoft Visual Basic. The following **section** describes how to use this file.

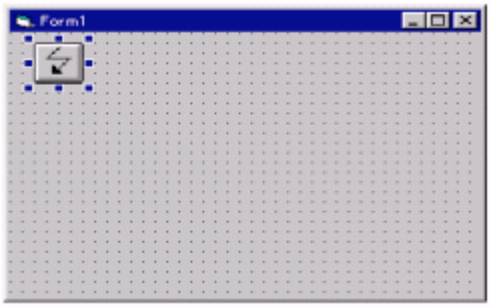
5.2 Example of Using Events

Example using VB

Start VB and open a project. Select “WeEvent OLE Control Module” from the [Project] > [Component] menu to add the control module to the project. The following figure shows the control after it is added.



Next, double click the WeEvent **control** icon and paste it to the form that will receive the events. The following figure shows the form after the control has been added.



Double click the WeEvent **control** icon above to create the event handler.
There are two types of event handlers, “WeEvent” and “WeBlock.”

WeEvent

This handler is used to handle all events except the block end event. The first parameter holds the event information and the second parameter holds a number representing the source of the phenomenon (specifically the station number in the upper 16 bits and the module number in the lower 16 bits). In order to get the module handle, the WeGetHandle API is called using the second parameter.

```
Private Sub WeEvent1_WeEvent(ByVal event As Long, ByVal handle As
Long)

End Sub
```

Note

When WE_EV_POWER is received, the second parameter is always “0.” To confirm to which station this event applies, you must make a query to the measuring stations that are connected using the WeGetPower API.

WeBlock

This handler is used to handle the block end event (WE_EV_BLOCKEND). The first parameter holds the current block number and the second parameter holds a number representing the source of the phenomenon (specifically the station number in the upper 16 bits and the module number in the lower 16 bits). In order to get the module handle, the WeGetHandle API is called using the second parameter.

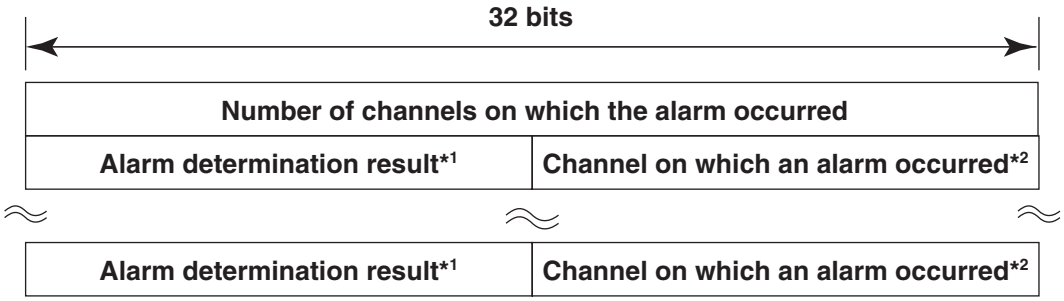
```
Private Sub WeEvent1_WeBlock (ByVal blockNo As Long, ByVal handle As
Long)

End Sub
```

WeAlarm

This handler is used to handle the alarm occurrence/release event (WE_EV_ALARM). As opposed to WE_EV_TRIG_HIGH, WE_EV_TRIG_LOW, and WE_EV_TRIG_PULSE events does not allow you to identify the channel on which the alarm occurred, WE_EV_ALARM allows you to do so.

The first parameter holds the alarm information (see below), and the second parameter holds a number representing the source of the phenomenon (specifically the station number in the upper 16 bits and the module number in the lower 16 bits). In order to get the module handle, the WeGetHandle API is called using the second parameter.

The first parameter holds a pointer to the following data structure (alarm information).



*1 The alarm determination result is defined as follows:
bit 31: WE_ALARM_PULSE (0x80000000)
bit 30: WE_ALARM_TRUE (0x40000000)
bit 29: WE_ALARM_FALSE (0x20000000)

*2 The channels on which alarms occur are counted starting from 1.
The number of “alarm determination result” and “channel on which an alarm occurred” that exists in the data structure is equal to the “number of channels on which the alarm occurred.” On the WE7231, the maximum is 240 channels.

The memory area that is allocated by the API is passed to the first parameter. Retrieve the alarm information by using the WEGetAlarmInfo API to copy and release the application area.

```
Private Sub WeEvent1_WeAlarm(ByVal dp As Long, ByVal handle As Long)

End Sub
```

• Example in which data are collected continuously by using the block end event

In order for the WeEvent.ocx to receive the event, call the initialization API (WeInit) to set the window handle.

```
' Initialization
ret = WeInit (WeEvent1.hWnd, "Optical", WE_CONTROLLER)
```

After starting the measurement station's operation, force an event notification every time a block of data (100 points) is acquired.

```
Dim hSt As Long
Dim hMo7251 As Long
Dim evHandle As Long
' Get the station handle for the station named "Station 1."
ret = WeOpenStation ("Station1",hSt)
' Open module WE7251.
ret = WeOpenModule (hSt,"WE7251",1,hMo7251)
' Set the acquisition mode to Free Run.
ret = WeSetControl (hMo7251, "Acquisition Mode", "Free Run")
' Set the sampling interval to 0.01 s.
ret = WeSetControl (hMo7251, "Sampling Interval", "0.01")
' Force an event notification every time 100 data points are
' acquired (in other words, 100 points x 0.01 = 1 s).
ret = WeStartEx (hWE7251,100,0,0,WE_ACQ_BLOCK_EVENT,evHandle)
```

At this point, the WeEvent control is ready to receive the events. The appropriate program codes are now written for the handler.

```
Private Sub WeEvent1_WeBlock (ByVal blockNo As Long, ByVal handle As
Long)
  Dim hMo As Long
  ret = WeGetHandle (handle,hMo) ' Get handle.
  if hMo = hMo7251 Then    ' If the handle is that of the WE7251
                           ' module,
    ' save the acquired data (raw data) to a file.
    ret = WeSaveAcqData (hMo,1,blockNo,"c:\tmp\WeSaveAcqData",1)
  End If
End Sub
```

To stop the measurement, call the WeStopEX API that releases the event that was obtained using WeStartEx.

```
ret = WeStopEX (evHandle)
```

• Example in which an alarm occurrence is detected on the WE7241

In order for the WeEvent.ocx to receive the event, call the initialization API (WeInit) to set the window handle.

```
' Initialization
ret = WeInit (WeEvent1.hWnd, "Optical",WE_CONTROLLER)
```

Force an event notification when the measured value exceeds 1000 °C.

```
Dim hSt As Long
Dim hMo7241 As Long
Dim evHandle As Long
' Get the station handle for the station named "Station 1."
ret = WeOpenStation ("Station1",hSt)
' Open module WE7241.
ret = WeOpenModule (hSt,"WE7241",1,hMo7241)
' Set the range of CH1 to type K.
ret = WeSetControl (hMo7241, "CH1:Range", "Type K")
' Set the alarm condition to High.
ret = WeSetControl (hMo7241, "CH1:Alarm", "High")
' Set the upper limit of the alarm to 1000°C.
ret = WeSetControl (hMo7241, "CH1:AlarmHigh", "1000")
' Force an event notification when the measured value exceeds 1000
' °C.
ret = WeCreateEvent (hMo7241, WE_EV_TRIG_HIGH, WE_EV_CONT, evHandle)
' Start measurement.
ret = WeStart (hMo7241)
```

The program is ready to receive the events. Next, we will write the program inside the handlers.

```
Private Sub WeEvent1_WeEvent (ByVal event As Long, ByVal handle As
Long)
  Dim hMo As Long
  ' Get handle.
  ret = WeGetHandle (handle, hMo)
  ' If a level trigger (HIGH) occurs on the WE7241,
  if hMo = hMo7241 Then
    if event = WE_EV_TRIG_HIGH Then
      ' Diplay Massage.
      MsgBox ("an alarm occurrence is detected on the WE7241.")
    End If
  End If
End Sub
```

To stop the event notification, call WeReleaseEvent.

```
' Clear the event.
ret = WeReleaseEvent (hMo7241, evHandle)
```

• Example in which an alarm occurrence is detected on the WE7231 module

In order for the WeEvent.ocx to receive the event, call the initialization API (WeInit) to set the window handle.

```
ret = WeInit(WeEvent1.hWnd, “Optical”, WE_CONTROLLER)    ‘ Initialize
```

Force an event notification when the measured value of CH1 exceeds 100 ++ !!!.

```
Dim hSt As Long
Dim hMo7231 As Long
Dim evHandle As Long
‘ Get the station handle for the station named “Station 1.”
ret = WeOpenStation(“Station1”, hSt )
‘ Open module WE7231.
ret = WeOpenModule(hSt, “WE7231”, 1, hMo7231)
‘ Set the thermocouple type to Type K.
ret = WeSetControl(hMo7231, “CH1:Range”, “Type K”)
‘ Set the alarm condition to High.
ret = WeSetControl(hMo7231, “CH1:Alarm:Type”, “High”)
‘ Set the alarm high level to 100 ++.
ret = WeSetControl(hMo7231, “CH1:Alarm:High”, “100”)
‘ Force a WE_EV_ALARM notification when the measured value of CH1
exceeds 100 ++.
ret = WeCreateEvent(hMo7231,WE_EV_ALARM,WE_EV_CONT,evHandle)
ret = WeStart(hMo7231)
```

The program is ready to receive the events. Next, we will write the program inside the handlers.

```
‘ Alarm event handler
Private Sub WeEvent1_WeAlarm(ByVal dp As Long, ByVal handle As Long)
    Dim hMo As Long
    Dim recSize As Long
    Dim i As Long
    ret = WeGetHandle(handle, hMo)                          ‘ Get
handle
    If hMo = hMo7231 Then
        RtlMoveMemory recSize, dp, 4                        ‘ Get
the data size
        If recSize <> 0 Then
            ReDim buf(recSize) As Long                      ‘
Allocate a buffer
            ret = WeGetAlarmInfo(dp, buf(0), recSize, handle) ‘ Get
alarm information
            For i = 0 To recSize - 1
            ‘ Display the alarm determination result and alarm
determination channel information on the debug window.
                Debug.Print “Alarm information = “ & (Hex(buf(i)))
            Next i
        End If
    End If
End Sub
```

Example using VC++

For VC, WE events are sent to the main thread. Thus, we override the PreTrancelateMessage of CWinApp.

```
BOOL WeApiTestApp::PreTranslateMessage(MSG* pMsg)
{
  return CWinApp::PreTranslateMessage(pMsg);
}
```

To retransmit the WeAPI event to the window that wishes to handle it, do the following. The WeAPI event is sent to the CMainFrame in the following case. As explained earlier, there are three types of events:
WM_WE_EVENT: The message number for all events except the block end event.
WM_WE_ALARM: The message number for alarm occurrence/release event.
WM_WE_EVENT: The message number for other events.
The handling of the parameters is the same as for VB.

```
BOOL WeApiTestApp::PreTranslateMessage(MSG* pMsg)
{
  UINT msg = pMsg->message;
  if (msg == WM_WE_EVENT|| msg == WM_WE_BLOCK|| msg == WM_WE_ALARM)
return m_pMainWnd->SendMessage(msg, pMsg->wParam, pMsg->lParam);
  else
return CWinApp::PreTranslateMessage(pMsg);
}
```

The event handler is defined in the CMainFrame class.

```
BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
  //{AFX_MSG_MAP(CGUIFrame)
  ON_WM_CREATE()
  ON_WM_CLOSE()
  //}AFX_MSG_MAP
  // Global help command
  ON_MESSAGE(WM_WE_EVENT, OnWeEvent)
  ON_MESSAGE(WM_WE_BLOCK, OnWeBlock)
  ON_MESSAGE(WM_WE_ALARM, OnWeAlarm)
END_MESSAGE_MAP()

// Handler for other events
LRESULT CMainFrame::OnWeEvent(WPARAM event, LPARAM handle)
{
    return TRUE ;
}

// Handler for the block end event.
LRESULT CMainFrame::OnWeBlock(WPARAM blockNo, LPARAM handle)
{
    return TRUE ;
}

//
LRESULT CMainFrame::OnWeAlarm(WPARAM info, LPARAM handle)
{
    return TRUE ;
}
```

6.1 The List of Functions

Initialization

Function name	Description
WeInit	Carry out initialization procedures.
WeExit	Carry out termination procedures.

Handle/Link

Function name	Description
WeOpenStation	Get the station handle.
WeOpenModule	Get the module handle.
WeLinkStation	Get the station link handle.
WeLinkModule	Get the module link handle.
WeCloseHandle	Release handles.

Measuring station control

Function name	Description
WeGetStationList	Get the list of station names.
WeGetStationInfo	Get station information.
WePower	Turn ON/OFF the power of the measuring station.
WeRestart	Restart the power of the measuring station.
WeSetStationName	Set the station name and the comment.
WeGetStationName	Get the station name and the comment.
WeIdetifyStation	Identify the measuring station.
WeGetPower	Get the power's ON/OFF state of the measuring station.
WeSetStatusLED	Set the STATUS LED.
WeGetStatusLED	Get the STATUS LED.
WeSetDIOConfig	Set the DIO configuration.
WeGetDIOConfig	Get the DIO configuration.
WeSetDIO	Set the DIO.
WeGetDIO	Get the DIO.

Module control

Function name	Description
WeGetModuleInfo	Get module information.

Settings

Function name	Description
WeInitSetup	Set the current setup data to factory default.
WeInitPreset	Set the preset value to the factory default.
WeSaveSetup	Save current setup data or update the preset values using the current settings.
WeLoadSetup	Load the setup data from a file or update the current setup data with preset values.
WeCopySetup	Copy the setup data of a module.
WeCopyChSetup	Copy the setup data between channels.
WeCopyChSetupEx	Copy the setup data between channels (extended).
WeSetControl	Set control parameters using a variant variable.
WeGetControl	Get control parameters using a variant variable.
WeSetControlEx	Set control parameters.
WeGetControlEx	Get control parameters.
WeSetQuryControl	Set control parameters then get the parameters.
WeSetScaleInfo	Set scale conversion information.
WeGetScaleInfo	Get scale conversion information.

Synchronization between Modules

Function name	Description
WeExecManualTrig	Generate a trigger signal on the trigger bus.
WeSetModuleBus	Set the I/O setting of the trigger and clock bus.
WeGetModuleBus	Get the I/O setting of the trigger and clock bus.
WeSetTrigBusLogic	Set the trigger bus logic.
WeGetTrigBusLogic	Get the trigger bus logic.
WeSetEXTIO	Set the trigger time base signal I/O setting of the EXT I/O terminal.
WeGetEXTIO	Get the trigger time base signal I/O setting of the EXT I/O terminal.
WeSetTRIG	Set the destination and polarity of the trigger bus signal entering the TRIG terminal.
WeGetTRIG	Get the destination and polarity of the trigger bus signal entering the TRIG terminal.
WeSetTRIGIN	Set the destination and polarity of the trigger bus signal entering the TRIG IN terminal.
WeGetTRIGIN	Get the destination and polarity of the trigger bus signal entering the TRIG IN terminal.
WeSetClockBusSource	Set the time base source.
WeGetClockBusSource	Get the time base source.
WeSetRcvTrigPacket	Set the receive trigger packet.
WeSetSndTrigPacket	Set the transmit trigger packet.
WeFireTrigPacket	Generate trigger packet.
WeSetSndClockPacket	Set the transmit time base packet.
WeSetRcvClockPacket	Set the receive time base packet.
WeFireClockPacket	Generate time base packet.
WeOutputEXTIOEvent	Output EXT I/O terminal event.
WeExecManualArming	Generate arming signal.
WeSetArmingSource	Set the arming source.
WeGetArmingSource	Get the arming source.

GUI control

Function name	Description
WeShowModuleWindow	Display the module window.
WeCloseModuleWindow	Close the module window.
WeIsModuleWindow	Query the open/close condition of the module window.
WeShowTrigWindow	Display the trigger source/time base source/arming setting dialog box.
WeCloseTrigWindow	Close the trigger source/time base source/arming setting dialog box.
WeIsTrigWindow	Query the open/close condition of the trigger source/time base source/arming setting dialog box.
WeShowLinearScaleWindow	Displays the convert scale dialog box.
WeCloseLinearScaleWindow	Closes the convert scale dialog box.
WeIsLinearScaleWindow	Queries the open/close condition of the convert scale dialog box.

Measurement control

Function name	Description
WeStart	Start operation.
WeStop	Stop operation.
WeStartEx	Start operation (extended).
WeStopEx	Stop operation (extended).
WeStartSingle	Start single acquisition.
WeStartWithEvent	Start with end notify event.
WeIsRun	Get the Run/Stop condition.
WeLatchData	Latch data.
WeGetAcqDataInfo	Get data information.
WeGetAcqDataSize	Get the number of points and the total number of bytes of the acquisition data.
WeGetAcqData	Get raw acquisition data.
WeGetAcqDataEx	Get acquisition data (extended).
WeGetCurrentData	Get instantaneous acquisition data.
WeGetScaleCurrentData	Get the data obtained after scaling the instantaneous values.
WeGetScaleCurrentDataEx	Get the data obtained after scaling the instantaneous values (extended).

Function name	Description
WeGetScaleData	Get physical value data.
WeGetScaleDataEx	Get physical value data (extended).
WeGetMeasureParam	Get automated measurement values.
WeSaveAcqData	Save raw acquisition data to file.
WeSaveScaleData	Save physical value data to file.
WeSaveScaleDataEx	Save physical value data to file (extended).
WeSaveAsciiData	Save ASCII data.
WeSaveScaleAsciiData	Save ASCII data of the scaled acquisition data.
WeSaveScaleAsciiDataEx	Save ASCII data of the scaled acquisition data (extended).
WeSaveAcqHeader	Save the waveform information file.
WeSavePatternData	Save the pattern data (arbitrary waveform data) file.
WeLoadPatternData	Load the pattern data (arbitrary waveform data) file.
WeLoadPatternDataEx	Load the ipattern data (arbitrary waveform data) file (extended).
WeSetOverRun	Switch the overrun detection.
WeGetOverRun	Query the overrun detection setting.

Events

Function name	Description
WeCreateEvent	Request generation of an event.
WeSetEventPattern	Set the factor that triggers the event.
WeResetEventPattern	Clear the factor that triggers the event.
WeSetEventMode	Set the event mode.
WeReleaseEvent	Release event handle.

Waveform parameter computation

Function name	Description
WeExecMeasureParam	Compute waveform parameters.
WeExecMeasureParamAcqData	Compute waveform parameters using raw data.

Filter API for the 4-CH, 100 kS/s D/A module WE7281

Function name	Description
WeWvf2S16GetSize	Get the byte size when the waveform data in the specified file is converted to s16 format.
WeWvf2W32GetSize	Get the byte size when the waveform data in the specified file is converted to w32 format.
WeWvf2W7281GetSize	Get the byte size when the waveform data in the specified file is converted to W7281 format.
WeWvf2S16	Convert the specified file to s16 format.
WeWvf2W32	Convert the specified file to w32 format.
WeWvf2W7281	Convert the specified file to W7281 format.

Others

Function name	Description
WeGetHandle	Get the handle from the second parameter of the event.
WeIsNan	Check whether or not the data are non-numeric.
WeLoadHostsFile	Load the host file.
WeTransAcqData	Transfer the acquisition data.

6.2 Initialization Functions

WeInit

Description

This is the function that carries out the initialization procedures. This function must be called first when using the WE Control API in an application program. Some of the activities that take place when this function is called include: the initialization of the network, the automatic detection of connected stations, and the initialization of the API environment.

Syntax

WeInit (ByVal *hWnd* As Long, ByVal *comm* As String, ByVal *ptype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hWnd (IN) Window handle for receiving the WE7000 defined events (described later) that are generated by the station or module. For Visual Basic, specify the window handle of the WeEvent control (WeEvent.ocx). For other languages (Visual C++), specify 0 (NULL).

comm (IN) The type of communication interface between the PC and meauring station.

"optical devicename = we7034" when using Optical Interface card WE7033/WE7034

"optical devicename = we7036" when using Optical Interface card WE7035/WE7036

"serial [option]" when using serial port (only for WE400/WE800)

Enter [option] settings as shown below. Be sure to specify an IP address and subnet mask.IP address: IP = IP address for the PC (ex. 192.168.21.128)

Subnet Mask: NETMASK = subnet mask (ex. 255.255.255.0)

Port Number (optional): PORTNO = port number (default is 34191)

Group number (optional): GROUPNO = group number (default is 0)

Transmission Mode (optional): COMPATIBLE = ON / OFF (default is ON)

[option] example: IP=192.168.21.128 NETMASK=255.255.255.0 PORTNO=34191 GROUPNO=1 COMPATIBLE=OFF

"ethernet95 [option]" when using Windows 95

You can specify the IP address, subnet mask, and port number with this option. The option can be omitted. Example: IP = 192.168.21.128 NETMASK = 255.255.255.0 PORTNO = 34191

"ethernet [option]" in Windows 98 or higher
The option can be omitted. If you wish to enter settings, follow the instructions above. If you only have one Ethernet interface, it is not necessary to specify an IP address or subnet mask. When using more than one Ethernet interface, specify which Ethernet interface you are using in the IP address. It's also necessary to enter the subnet mask.
"USB" when using USB interface (only for WE500/WE900)
For detailed information regarding this option, see "Start Option" in the WE7000 PC-Based Measurement Instruments and WE7051 Ethernet Module*/WE7052 Fast Ethernet Module* User's Manuals.
* Only for WE400/WE800.

ptype (IN) Execution mode of the application program.
There are two execution modes: the controller mode in which the network and station can be controlled, and the server mode in which certain functions can be executed (store or monitor data, for example). The current version only supports the controller mode.

```
WE_CONTROLLER          ' Controller
WE_SERVER              ' Server
```

Note:

Specify the window that will receive the WE7000 defined events in the hWnd parameter. **For languages such as Visual C++ that completely support multi-threaded environments, the event messages are sent directly to the main thread of the application.** Therefore, the hWnd parameter does not need to be specified in this case.
Regardless of the version of Windows, if more than one Ethernet card is installed you must specify an IP address and subnet mask.

Example (Visual Basic)

```
' Receive the events with the WeEvent control (WeEvent.ocx).
' Initialize by specifying optical interface (WE7035/WE7036) and
' controller mode.
Dim ret As Integer
ret = WeInit (WeEvent1.hWnd, "optical devicename = we7036",
WE_CONTROLLER)
```

WeExit

Description

This is the function that carries out the termination procedures. Some of the activities that take place when this function is called include: termination of the communication driver and API environment. Make sure to execute this function at the end of the application program.

Syntax

WeExit () As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

None

Example (Visual Basic)

```
' Carry out termination procedures.
Dim ret As Integer
ret = WeExit ()
```

6.3 Handle/Link Functions

WeOpenStation

Description

Gets the station handle for controlling the station, by specifying the station name. You can also get the handle for controlling all the measuring stations on the network by using a broadcast handle.

Syntax

WeOpenStation (ByVal *name* As String, ByRef *hSt* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

name (IN)	Station name. To get the handle that can be used to control all the measuring stations on the network, specify “BROADCAST.”
hSt (OUT)	Station handle. Returns NULL if unsuccessful.

Note:

The broadcast handle cannot be specified in some of the functions that allow normal station handles (see the explanation for each function). In addition, if a network contains multiple measuring stations with the same name, only the handle of the closest measuring station can be obtained. Therefore, make sure to assign different names to measuring stations on the same network.

Example (Visual Basic)

```
' Get the station handle for the measuring station named "Station 1."
Dim stHandle As Long
Dim ret As Integer
ret = WeOpenStation ("Station 1", stHandle)
```

WeOpenModule

Description

Gets the module handle for controlling the module.

Syntax

WeOpenModule (ByVal *hSt* As Long, ByVal *name* As String, ByVal *connection* As Integer, ByRef hMo As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

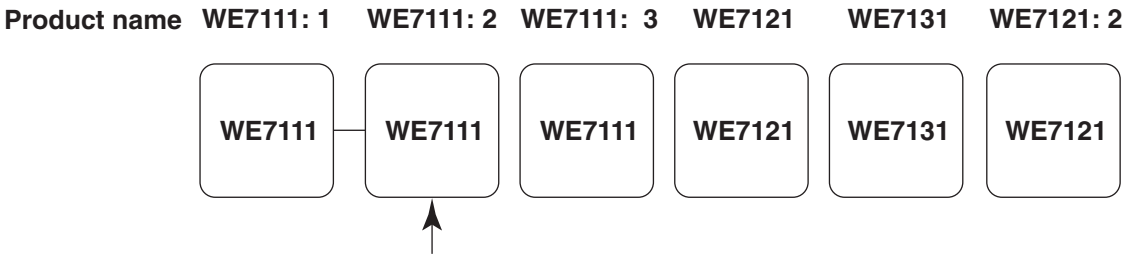
Parameters

hSt (IN)	Station handle
name (IN)	Module's product name [:number] or slot number **Using product name eliminates the need to know the slot number or position of the module in the station.** The number inside the brackets indicates the position of the module counting the modules of the same type starting from the left most module of the same type (1 origin). (See the example below on how to specify the module's product name.) The brackets ( ) can be omitted in which case it is considered to be [:1]. To specify the 4-CH, 100 kS/s Isolated Digitizer Module WE7272, use "WE7271."
connection (IN)	The number of modules you wish to link. Specify 1 if you do not wish to link the modules.
hMo (OUT)	Module handle. (Returns NULL if unsuccessful.)

Note:

By specifying the name parameter with the model's product name instead of the slot number, the program will be independent of the actual slot positions of the modules. Therefore, using the module's product name is encouraged. If the station is turned OFF, the module handles that have been obtained remains effective. This means that the same module handles can be used to control the modules when the measuring station is turned back ON. However, if the modules are switched or moved to different slot positions while the power is OFF, the behavior afterwards is not guaranteed.

Example of specifying the module's product name



Attempting to get the module handle of a child module that is linked to an opened module results in an error. Opening a child module that is linked to an unopened module clears the link.

Example (Visual Basic)

- Open module WE7111 with a link number of 2

```
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
```

- Open the module in slot 2 with a link number of 1
  (When the "WE7111: 1" is not opened.)

```
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeOpenModule (stHandle, "2", 1, dsoHandle)
```

WeLinkStation

Description

Gets the station link handle. The station link handle can be used to simultaneously control multiple stations.

Syntax

WeLinkStation (ByVal *num* As Integer, ByRef *hSt* As Long, ByRef *hlSt* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

num (IN)	The number of stations to link
hSt (IN)	An array of the station handles of the stations to link. The number of station handles in the array must equal the num parameter.
hlSt (OUT)	Station link handle

Example (Visual Basic)

```
' Get the link handle of two stations, then turn ON the power.
Dim stLinkHandle As Long
Dim stHandle (2) As Long
Dim hSt1 As Long, hSt2 As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt1)
' Get the station handle of the station named "Station 2."
ret = WeOpenStation ("Station 2", hSt2)
stHandle (0) = hSt1
stHanlde (1) = hSt2
ret = WeLinkStation (2, stHandle (0), stLinkHandle)
ret = WePower (stLinkHandle, WE_ON)
```

WeLinkModule

Description

Gets the module link handle. The module link handle can be used to the control multiple modules simultaneously.

Syntax

WeLinkModule (ByVal *num* As Integer, ByRef *hMo* As Long, ByRef *hlMo* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. Attempting to get a link handle for modules of a different type results in an error.

Parameters

num (IN)	The number of modules to link
hMo (IN)	An array of the module handles of the modules to be linked. The number of module handles in the array must equal the num parameter.
hlMo (OUT)	Module link handle

Example (Visual Basic)

```
' Get the module link handle of two modules, then set the offset
' voltage to 1.234 V.
Dim mdLinkHandle As Long
Dim moHandle (2) As Long
Dim hSt1 As Long
Dim hMo1 As Long, hMo2 As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1"
ret = WeOpenStation ("Station 1", hSt1)
' Open the module at slot 2 with a link number of 1.
ret = WeOpenModule (stHandle, "2", 1, hMo1)
' Open the module at slot 4 with a link number of 1.
ret = WeOpenModule (stHandle, "4", 1, hMo2)
moHandle (0) = hMo1
moHanlde (1) = hMo2
ret = WeLinkModule (2, moHandle (0), mdLinkHandle)
ret = WeSetControl (mdLinkHandle, "Offset", "1.234")
```

WeCloseHandle

Description

Releases the measuring station, module, and link handles. Releasing the station handle also releases the module handles and module link handles within the measuring station.

Syntax

WeCloseHandle (ByVal *hHandle* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN) Station handle, station link handle, module handle, or module link handle

Example (Visual Basic)

```
' Release the station handle of the station named "Station 1."
Dim hSt As Long
Dim hMo As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt)
' Open the module at slot 2 with a link number of 1.
ret = WeOpenModule (hSt, "2", 1, hMo)
ret = WeCloseHandle (hSt)
```

6.4 Station Control

WeGetStationList

Description

Gets the list of station names on a network.

Syntax

WeGetStationList (ByRef *num* As Integer, *list* As StationListArray) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

num (OUT) The number of measuring stations (includes the controller). The maximum number of measuring stations is defined by MaxStationNum.

list (OUT) The pointer to the structure containing the station name list.
The first member of the list contains controller information. The structure containing the station name list is as follows:

```
Type StationList                          ' Structure containing the station
                                          ' name list
  name As String * MaxStationName         ' Station name
  addr As Integer                         ' Station's logical address
End Type
```

Example (Visual Basic)

```
' Get the list of station names.
Dim list As StationListArray
Dim num As Integer
Dim ret As Integer
ret = WeGetStationList (num, list)
```

WeGetStationInfo

Description

Gets the station information.

Syntax

WeGetStationInfo (ByVal *hst* As Long, ByRef *info* As StationInfoEx) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN) Station handle of the measuring station you wish to get information on

info (OUT) The pointer to the station information structure
The station information structure is as follows:

```
Type ModuleInfo                 ' Module information structure
  productId As String * 8       ' Product name (A string expressed as
                                ' WExxxx.
                                '  (Example: WE7111)
  chNum As Integer              ' Number of channels per module
  connect As Byte               ' Number of links.  This number is 1
                                ' if the module is not linked.
  connectType As Byte           ' Link condition
        ' WE_PARENT_MODULE The parent module of the link
        ' (The left most module of the link)
        ' WE_CHILD_MODULE The child module of the link
        ' WE_SINGLE_MODULE Module that is not linked.
  version As Long               ' Software version of  the module driver
End Type
Type StationInfoEx              ' Station information structure
  addr As Integer               ' Station's logical address
  state As Byte                 ' Power ON/OFF state of the remote
station.
                                ' WE_OFF: Off
                                ' WE_ON: On
  moduleNum As Byte             ' Number of modules that are inserted
  name As String * MaxStationName                    ' Station name
  comment As String * MaxStationComment              ' Comment
string
  mdInfo (9) As ModuleInfo      ' Module information.  The slot number
                                ' is specified in the parenthesis ().
End Type
```

Example (Visual Basic)

```
' Get the station information of Station 1.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt)
Dim inf As StationInfoEx
ret = WeGetStationInfo (hSt, inf)
```

WePower

Description

Turns ON/OFF the measuring station's power.

Syntax

WePower (ByVal *hst* As Long, ByVal *sw* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle, station link handle, or broadcast handle
sw (IN)	Switch state WE_OFF turns OFF the power. WE_ON turns ON the power.

Example (Visual Basic)

```
' Turn ON Station 1.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt)
ret = WePower (hSt, WE_ON)
```

WeRestart

Description

Restarts the station. The operation is similar to turning ON the power. However, the communication module does not restart.

Syntax

WeRestart (ByVal *hst* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle, station link handle, or broadcast handle

Example (Visual Basic)

```
' Restarts Station 1.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt)
ret = WeRestart (hSt)
```

WeSetStationName

Description

Sets the station name and the comments for the measuring station.

Syntax

WeSetStationName (ByVal *hSt* As Long, ByVal *name* As String, ByVal *comment* As String) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. Attempting to set a station name that already exists results in an error.

Parameters

hSt (IN)	Station handle
name (IN)	A string containing the station name (up to 256 characters) Use " " if you are not specifying a name. The default station name is comprised of the string "Station" + the logical address number (for example: Station 1).
comment (IN)	A string containing a comment (up to 256 characters) Use " " if you are not specifying a comment.

Example (Visual Basic)

```
' Set "Plant 1" for the station name and "Measure Sample" for the '
' comment.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt)
ret = WeSetStationName (hSt, "Plant 1", "Measure Sample")
```

Note:

Executing this command overwrites the flash ROM of the measuring station. There is a limitation on the number of times the flash ROM can be overwritten (approx. 100,000 times). Therefore, minimize the execution of this command.

WeGetStationName

Description

Gets the station name and the comment for the measuring station.

Syntax

WeGetStationName (ByVal *hSt* As Long, ByRef *name* As String, ByRef *comment* As String) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle
name (OUT)	Station name (up to 256 characters)
comment (OUT)	A string containing a comment (up to 256 characters)

Example (Visual Basic)

```
' Gets the station name and the comment.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt)
Dim strName As String
' Allocate an area for retrieving the station name.
strName = String (MaxStationName, " ")
Dim strComment As String
' Allocate an area for retrieving the comment.
strComment = String (MaxStationComment, " ")
ret = WeGetStationName (hSt, strName, strComment)
```

WeIdentifyStation

Description

For measuring station identification, the LED of the optical interface module of the specified measuring station blinks. This is valid only when the optical interface module is being used as the communication interface.

Syntax

WeIdentifyStation (ByVal *hSt* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle or station link handle

Example (Visual Basic)

```
' Identify Station 1.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt)
ret = WeIdentifyStation (hSt)
```

WeGetPower

Description

Gets the power ON/OFF state of the station.

Syntax

WeGetPower (ByVal *hSt* As Long, ByRef *sw* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle	
sw (OUT)	Power ON/OFF state of the remote station	
	WE_OFF	' The measuring station's power is OFF.
	WE_ON	' The measuring station's power is ON.

Example (Visual Basic)

```
' Get the power's ON/OFF state of Station 1.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", hSt)
Dim sw As Byte
ret = WeGetPower (hSt, sw)
```

WeSetStatusLED

Description

Turns ON/OFF the measuring station's STATUS LED.

Syntax

WeSetStatusLED (ByVal *hSt* As Long, ByVal *LEDNo* As Byte, ByVal *status* As Byte) As Integer

Integer

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle	
LEDNo (IN)	LED number	
	WE_LEDA	' LED A
	WE_LEDB	' LED B
status (IN)	LED setting	
	WE_OFF	' OFF
	WE_RED	' Illuminate in red
	WE_GRN	' Illuminate in green
	WE_ORG	' Illuminate in orange

Example (Visual Basic)

```
' Illuminate LED A of Station 1 in green.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
ret = WeSetStatusLED (hSt, 1, WE_GRN)
```

Note:

This function can be used only on the WE500 and WE900.

WeGetStatusLED

Description

Gets the illumination status of the measuring station's STATUS LED.

Syntax

WeGetStatusLEDA (ByVal *hSt* As Long, ByVal *LEDNo* As Byte, ByRef *status* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle	
LEDNo (IN)	LED number	
	WE_LEDA	' LED A
	WE_LEDB	' LED B

status (OUT) LED A status
WE_OFF ' OFF
WE_RED ' Illuminated in red
WE_GRN ' Illuminated in green
WE_ORG ' Illuminated in orange

Example (Visual Basic)

```
' Get the status of LED A on Station 1.
Dim hSt As Long
Dim ret As Integer
Dim status As Byte
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
ret = WeGetStatusLED (hSt, 1, status)
```

Note:

This function can be used only on the WE500 and WE900.

WeSetDIOConfig

Description

Sets the input/output direction, the change detection polarity, and the change detection mask of the DIO pin of the measuring station's EXT IO.

Syntax

WeSetDIOConfig (ByVal *hSt* As Long, ByVal *Pin* As Integer, ByVal *Direct* As Byte, ByVal *Pol* As Byte, ByVal *Mask* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN) Station handle
Pin(IN) Pin number (0 to 3)
Direct (IN) Input/output direction setting
WE_KEEP ' Keep current setting
WE_IN ' Input
WE_OUT ' Output
Pol (IN) Change detection polarity setting
WE_KEEP ' Keep current setting
WE_RISE ' Rise
WE_FALL ' Fall
Mask (IN) Change detection mask setting
WE_KEEP ' Keep current setting
WE_MASK ' Not detect (if this is specified, WE_EV_DIO# events will not be activated.)
WE_DETECT ' Detect (if this is specified, WE_EV_DIO# (where # is the pin number) events will be activated. To generate an event, the event must also be defined using Create Event.)

Example (Visual Basic)

```
' Set conditions for pin number 1 on Station 1.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
ret = WePower (hSt, WE_ON)
ret = WeCreateEvent (hSt, WE_EV_DIO1, evHandle)
ret = WeSetDIOConfig (hSt, 1, WE_IN, WE_RISE, WE_DETECT)
```

Note:

This function can be used only on the WE500 and WE900.

WeGetDIOConfig

Description

Gets the input/output direction, the change detection polarity, and the change detection mask of the DIO pin of the measuring station's EXT IO.

Syntax

WeGetDIOConfig (ByVal *hSt* As Long, ByVal *Pin* As Integer, ByRef *Direct* As Byte, ByRef *Pol* As Byte, ByRef *Mask* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle
Pin (IN)	Pin number (0 to 3)
Direct (OUT)	Input/Output direction
	WE_IN ' Input
	WE_OUT ' Output
Pol (OUT)	Change detection polarity
	WE_RISE ' Rise
	WE_FALL ' Fall
Mask (OUT)	Change detection mask
	WE_MASK ' Not detect
	WE_DETECT ' Detect

Example (Visual Basic)

```
' Get conditions for pin number 1 on Station 1.
Dim hSt As Long
Dim ret As Integer
Dim Direct As Byte
Dim Pol As Byte
Dim Mask As Byte
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
ret = WeGetDIOConfig(hSt, 1, Direct, Pol, Mask)
```

Note:

This function can be used only on the WE500 and WE900.

WeSetDIO

Description

Sets the output on the DIO pin of the measuring station's EXT IO.

Syntax

WeSetDIO (ByVal *hSt* As Long, ByVal *Pin* As Integer, ByVal *Val* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle
Pin (IN)	Pin number (0 to 3)
Val (IN)	DIO output setting
	WE_KEEP ' Keep current setting
	WE_DIO_OFF ' Low status
	WE_DIO_ON ' High status

Example (Visual Basic)

```
' Output DIO pin number 1 on Station 1.
Dim hSt As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
ret = WeSetDIO (hSt, 1, WE_DIO_ON)
```

Note:

This function can be used only on the WE500 and WE900.

WeGetDIO

Description

Gets the status of the DIO pin of the measuring station's EXT IO.

Syntax

WeGetDIO (ByVal *hSt* As Long, ByVal *Pin* As Integer, ByRef *Val* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle
Pin (IN)	Pin number (0 to 3)
Val (OUT)	DIO input/output status
	WE_DIO_OFF ' Low status
	WE_DIO_ON ' High status

Example (Visual Basic)

```
' Get the input/output status of DIO pin number 1 on Station 1.
Dim hSt As Long
Dim ret As Integer
Dim Val As Byte
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
ret = WeGetDIO (hSt, 1, Val)
```

Note:

This function can be used only on the WE500 and WE900.

6.5 Module Control

WeGetModuleInfo

Description

Gets the module information.

Syntax

WeGetModuleInfo (ByVal *hMo* As Long, ByVal *no* As Integer, ByRef *info* As ModuleInfoEx) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
no (IN)	Link position of the module. The relative position counted starting from the parent module (0 origin). Specify 0 if the module is not linked.
info (OUT)	Pointer to the module extended information structure that is returned

```
Type ModuleInfoEx                          ' Module extended  Information
  mdInfo As ModuleInfo                     ' Module information
  maker As String* MaxMakerName            ' Name of the manufacturer
  productName As Stringq* MaxProductName   ' Product name
End Type
```

Note:

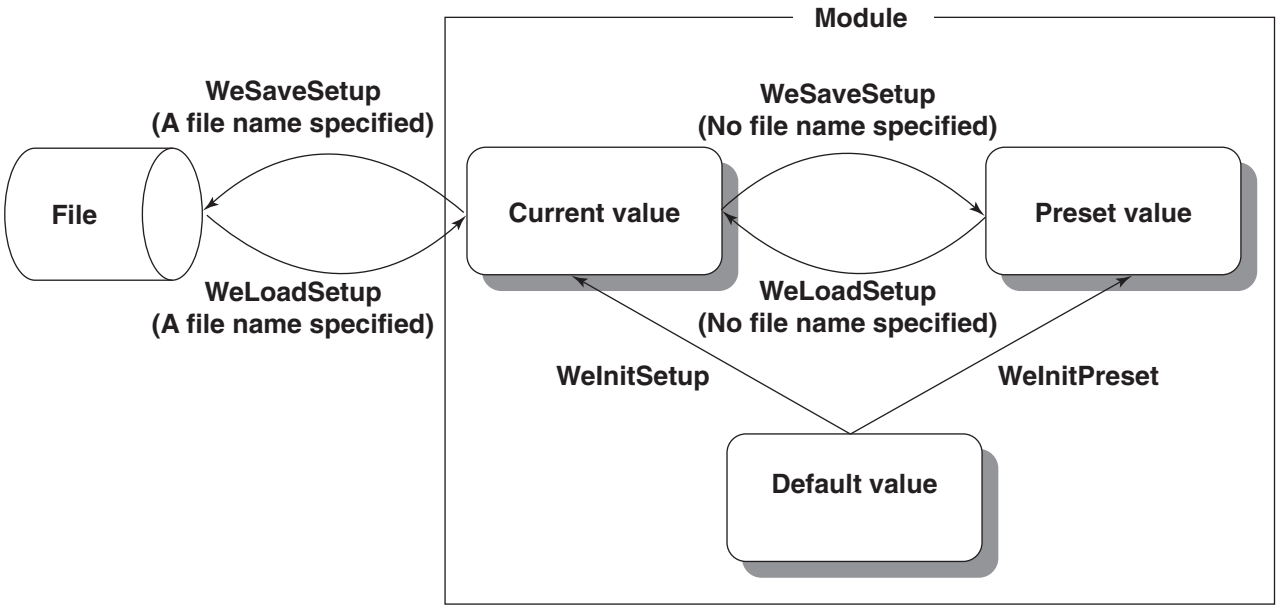
When the module is operating under a link, the module handle that was used to open the module is that of the parent module. Thus, the no parameter is necessary to specify the child module.

Example (Visual Basic)

```
' Get the parent module information of the linked modules.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
Dim inf As ModuleInfoEx
ret = WeGetModuleInfo (dsoHandle, 0, inf)
```

6.6 Settings

These functions are used to update and save the setup data of the modules. At the time of shipment, the default values (factory default) and the preset values (setup data used when the module is turned ON or when it is reset) are the same.



WeInitSetup

Description

Updates the current module's setup data with the default values.

Syntax

WeInitSetup (ByVal *hHandle* As Long, ByVal *no* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN)	Module handle, module link handle, station handle or station link handle If a station handle is specified, all modules in the station are affected.
no (IN)	Link position of the module. The relative position counted from the parent module (0 origin). Specify 0 if the module is not linked. This parameter is ignored if a station handle or station link handle is specified for the handle parameter.

Example (Visual Basic)

```
' Update the setup data of the parent module with the default values.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeInitSetup (dsoHandle, 0)
```

WeInitPreset

Description

Updates the preset values of the module with the default values.

Syntax

WeInitPreset (ByVal *hHandle* As Long, ByVal *no* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN)	Module hanle, module link handle, station handle or station link handle If a station handle is specified, all modules in the station are affected.
no (IN)	Link position of the module. The relative position counted starting from the parent module (0 origin). Specify 0 if the module is not linked. This parameter is ignored if a station handle or station link handle is specified for the handle parameter.

Example (Visual Basic)

```
' Update the preset values of the parent module with default values.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeInitPreset (dsoHandle, 0)
```

Note:

Executing this command overwrites the flash ROM of the measuring station. There is a limitation on the number of times the flash ROM can be overwritten (approx. 100,000 times). Therefore, minimize the execution of this command.

WeSaveSetup

Description

Saves the current setup data of the module or updates the preset values. When saving to a file, the file is saved with a file extension ".set."

Syntax

WeSaveSetup (ByVal *hHandle* As Long, ByVal *no* As Integer, ByVal *filename* As String) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN)	Module handle or station handle. If a station handle is specified, all modules in the station are affected.
no (IN)	Link position of the module. The relative position counted starting from the parent module (0 origin). Specify 0 if the module is not linked. This parameter is ignored if a station handle or station link handle is specified for the handle parameter.
filename (IN)	File name. Specify the file without the file extension ".set." Specifying " " for the file name updates the preset values.

Example (Visual Basic)

```
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Save the current setup data to the file, c:\param.set.
ret = WeSaveSetup (dsoHandle, 0, "c:\param")
' Update the preset values with the current setup data.
ret = WeSaveSetup (dsoHandle, 0, "")
```

Note:

Executing this command when only "" is specified for the file name overwrites the flash ROM of the measuring station. There is a limitation on the number of times the flash ROM can be overwritten (approx. 100,000 times). Therefore, minimize the execution of this command.

WeLoadSetup

Description

Updates the current setup data of the station or module using the setup data file or preset values.

Syntax

WeLoadSetup (ByVal *hHandle* As Long, ByVal *no* As Integer, ByVal *filename* As String) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the station handle is specified, the module configuration of the station is compared with that of the setup data file (includes the modules' slot positions). If they match, the setup data are updated (successful). If not, an error is returned.

Parameters

hHandle (IN)	Module handle or station handle
no (IN)	Link position of the module. The relative position counted starting from the parent module (0 origin). Specify 0 if the module is not linked. This parameter is ignored if a station handle or station link handle is specified for the handle parameter.
filename (IN)	File name. Specify the file without the file extension ".set." Specifying " " for the file name updates the current settings using preset values.

Example (Visual Basic)

```
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Update the current setup data with the file, c:\param.set.
ret = WeLoadSetup (dsoHandle, 0, "c:\param")
' Update the current setup data with the preset values.
ret = WeLoadSetup (dsoHandle, 0, "")
```

WeCopySetup

Description

Copies the current setup data of the module to a specified module of the same type.

Syntax

WeCopySetup (ByVal *hSrcMo* As Long, ByVal *hDesMo* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the module type of the copy source differs from that of the copy destination, an error is returned.

Parameters

hSrcMo (IN)	Module handle of the copy source
hDesMo (IN)	Module handle of the copy destination

Example (Visual Basic)

```
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle1 As Long, dsoHandle2 As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7111 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7111: 1", 1, dsoHandle1)
' Open the second WE7111 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7111: 2", 1, dsoHandle2)
' Copy the setup data.
ret = WeCopySetup (dsoHandle1, dsoHandle2)
```

WeCopyChSetup

Description

Copies the current setup data of a channel of a module to the specified channel. This function is valid only when the modules are linked.

Syntax

WeCopyChSetup (ByVal *hMo* As Long, ByVal *srcCh* As Integer, ByVal *desCh* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
srcCh (IN)	Channel number of the copy source
desCh (IN)	Channel number of the copy destination

Example (Visual Basic)

```
' Copy the setup data of the WE7111 module at channel 1 to the
' module at channel 3.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7111 module with a link number of 3.
ret = WeOpenModule (stHandle, "WE7111: 1", 3, dsoHandle)
ret = WeCopyChSetup (dsoHandle, 1, 3)
```

WeCopyChSetupEx

Description

Copies the current setup data of a channel of a module to the specified channel. This function can be used even when the modules are not linked.

Syntax

WeCopyChSetupEx (ByVal *hMo* As Long, ByVal *srcCh* As Long, ByVal *desStartCh* As Long, ByVal *desEndCh* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
srcCh (IN)	Channel number of the copy source. One origin. If the modules are linked, it is the link number.
desStartCh (IN)	Channel number of the copy destination. One origin. If the modules are linked, it is the link number.
desEndCh (IN)	Last channel number of the copy destination. One origin. If the modules are linked, it is the link number.

Example (Visual Basic)

```
Dim hSt As Long, hMo As Long
Dim ret As Integer
Dim sw As Byte
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
' Open the first WE7251 module with a link number of 3.
ret = WeOpenModule (hSt, "WE7251:1", 3, hMo)
' Copy the setup data of CH1 to the destination channels of CH2
through CH30.
ret = WeCopyChSetupEx (hMo, 1, 2, 30)
```

WeSetControl

Description

Sets the individual control parameters of a module.

Syntax

WeSetControl (ByVal *hHandle* As Long, ByVal *command* As String, ByVal *param* As Variant) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN)	Module handle or module link handle
command (IN)	ASCII command name that is defined for each module. For details, see the command table of each measurement module.
param (IN)	Parameter dependent on the command

Note:

ASCII commands are defined for each module. See chapter 8 for the command table of each module.

Example (Visual Basic)

```
' Set the offset value of the WE7111 module to 1.234 V.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7111 module with a link number of 3.
ret = WeOpenModule (stHandle, "WE7111: 1", 3, dsoHandle)
ret = WeSetControl (dsoHandle, "Offset", "1.234")
```

WeGetControl

Description

Gets the current values of the individual control parameters of the module.

Syntax

WeGetControl (ByVal *hHandle* As Long, ByVal *command* As String, ByRef *param* As Variant) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN)	Module handle
command (IN)	ASCII command name that is defined for each module. For details, see the command table of each measurement module.
param (OUT)	Parameter dependent on the command. In most cases, string data is returned. (There are exceptions.)

Note:

ASCII commands are defined for each module. See chapter 8 for the command table of each module.

Example (Visual Basic)

```
' Get the offset value of the WE7111 module.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7111 module with a link number of 3.
ret = WeOpenModule (stHandle, "WE7111: 1", 3, dsoHandle)
' Variable used to retrieve data
Dim value As Variant
ret = WeGetControl (dsoHandle, "Offset", value)
```

WeSetControlEx

Description

Sets the individual control parameters of the module. You can set the control parameters by specifying the data type.

Syntax

WeSetControlEx (ByVal *hHandle* As Long, ByVal *command* As String, ByVal *ptype* As Integer, ByVal *paramNum* As Long, ByRef *param* As Any) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN)	Module handle
command (IN)	ASCII command name that is defined for each module
type (IN)	Data type. The data types that can specified are defined for each command of the measurement modules. For details, see the command table for each module.

```
WE_NULL      ' No parameter
WE_UBYTE     ' 8-bit unsigned integer
WE_SBYTE     ' 8-bit signed integer
WE_UWORD     ' 16-bit unsigned integer
WE_SWORD     ' 16-bit signed integer
WE_ULONG     ' 32-bit unsigned integer
WE_SLONG     ' 32-bit signed integer
WE_FLOAT     ' 32-bit real number
WE_DOUBLE    ' 64-bit real number
```

paramNum (IN)	Number of data
param (IN)	data pointer

Example (Visual Basic)

```
' Set the offset value of the WE7111 module to 1.24 V.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7111 module with a link number of 3.
ret = WeOpenModule (stHandle, "WE7111: 1", 3, dsoHandle)
Dim param As Double
param = 1.24
ret = WeSetControlEx (dsoHandle, "Offset", WE_DOUBLE, 1, data)
```

WeGetControlEx

Description

Gets the current values of the individual control parameters of the module. You can get the control parameters by specifying the data type.

Syntax

WeGetControlEx (ByVal *hHandle* As Long, ByVal *command* As String, ByVal *ptype* As Integer, ByVal *paramNum* As Long, ByRef *param* As Any) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN)	Module handle
command (IN)	ASCII command name that is defined for each module
ptype (IN)	Data type. The data types that can specified are defined for each command of the measurement modules. For details, see the command table for each module.

```
WE_NULL       ' No parameter
WE_UBYTE      ' 8-bit unsigned integer
WE_SBYTE      ' 8-bit signed integer
WE_UWORD      ' 16-bit unsigned integer
WE_SWORD      ' 16-bit signed integer
WE_ULONG      ' 32-bit unsigned integer
WE_SLONG      ' 32-bit signed integer
WE_FLOAT      ' 32-bit real number
WE_DOUBLE     ' 64-bit real number
```

paramNum (IN)	Number of data to retrieve
param (OUT)	Data pointer

Example (Visual Basic)

```
' Get the offset value of the WE7111 module.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7111 module with a link number of 3.
ret = WeOpenModule (stHandle, "WE7111: 1", 3, dsoHandle)
Dim param As Double
ret = WeGetControlEx (dsoHandle, "Offset", WE_DOUBLE, 1, data)
```

WeSetQueryControl

Description

Sets the control parameter using the module handle and command, and then gets the current value of the control parameter.

Syntax

WINAPI WeSetQueryControl (ByVal *hMo* As Long, ByVal *command* As String, ByVal *stype* As Integer, ByVal *paramNum* As Long, ByRef *setParam* As Any, ByVal *rtype* As Integer, ByRef *getParam* As Variant) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
command (IN)	ASCII command name that is defined for each module
stype (IN)	Data type. The following symbols are defined:

```
WE_NULL      ' No parameter
WE_UBYTE     ' 8-bit unsigned integer
WE_SBYTE     ' 8-bit signed integer
WE_UWORD     ' 16-bit unsigned integer
WE_SWORD     ' 16-bit signed integer
WE_ULONG     ' 32-bit unsigned integer
WE_SLONG     ' 32-bit signed integer
WE_FLOAT     ' 32-bit real number
WE_DOUBLE    ' 64-bit real number
```

paramNum (IN)	Number of data to send
setParam (IN)	Pointer to the data to send
rtype (IN)	Data type of the data to receive (same as type)
getParam (OUT)	Pointer to the data to receive

Note:

Use this function only with the WE7021 module.

Example (Visual Basic)

```
' Using the WE7021 module, retrieve the block data from the
' connected device.
Dim stHandle As Long
Dim pBuf As Variant
Dim rSize As Long
rSize = 2002
Dim ret As Integer
Dim gpibHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7021 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7021", 1, gpibHandle)
ret = WeSetQueryControl (gpibHandle, "DATA", WE_ULONG, 1, rSize,
WE_UWORD, pBuf)
```

WeSetScaleInfo

Description

Sets scale conversion information to the measurement module. This function is equivalent to the settings in the convert scale dialog box. The information is stored to the module through operations such as update preset.

Syntax

WeSetScaleInfo (ByVal *hMo* As Long, ByVal *ch* As Integer, ByRef *info* As LinearScaleInfoArray) As Integer

Parameters

hMo (IN) Module handle

ch (IN) Channel number
One origin. If the modules are linked, it is the link number. -1 represents all channels.

info (IN) Scale conversion table information
If -1 is specified for ch, the number of info parameters that needs to be specified is equal to the number of channels.

```
Type LinearScaleInfo            ' Whether or not to enable scaling.
        scaling As Long         ' (1: enable scaling
                                ' 0: disable scaling)
        rsrv  As Long           ' Reserved.
        a As Double             ' The value a of scaling parameter ax+b.
        b As Double             ' The value b of scaling parameter ax+b.
        label As String * 32    ' Label name.
        unit As String * 16     ' Name of the unit.
End Type
Type LinearScaleInfoArray
        list(80) As LinearScaleInfo      ' Scaling info.
End Type
```

Example (Visual Basic)

```
Dim hSt As Long, hMo As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
' Open the first WE7275 module with a link number of 1.
ret = WeOpenModule (hSt, "WE7275:1", 1, hMo)
Dim info As LinearScaleInfoArray
info.list(0).scaling = 1
info.list(0).a = 1.0
info.list(0).b = 0.0
info.list(0).label = "CH1" + Chr$(0)
info.list(0).unit = "unit1" + Chr$(0)
info.list(1).scaling = 1
info.list(1).a = 2.0
info.list(1).b = 0.0
info.list(1).label = "CH2" + Chr$(0)
info.list(1).unit = "unit2" + Chr$(0)
' Set the scale information of all channels (2 channels).
ret = WeSetScaleInfo (hMo, -1, info)
```

WeGetScaleInfo

Description

Gets scale conversion information of the measurement module.

Syntax

WeGetScaleInfo (ByVal *hMo* As Long, ByVal *ch* As Integer, ByRef *info* As LinearScaleInfoArray) As Integer

Parameters

hMo (IN) Module handle

ch (IN) Channel number
One origin. If the modules are linked, it is the link number. -1 represents all channels.

info (IN) Scale conversion table information
If -1 is specified for ch, the number of info parameters that needs to be specified is equal to the number of channels.

```
Type LinearScaleInfo            ' Whether or not to enable scaling.
        scaling As Long         ' (1: enable scaling
                                ' 0: disable scaling)
        rsrv  As Long           ' Reserved.
        a As Double             ' The value a of scaling parameter ax+b.
        b As Double             ' The value b of scaling parameter ax+b.
        label As String * 32    ' Label name.
        unit As String * 16     ' Name of the unit.
End Type
Type LinearScaleInfoArray
        list(80) As LinearScaleInfo     ' Scaling information.
End Type
```

Example (Visual Basic)

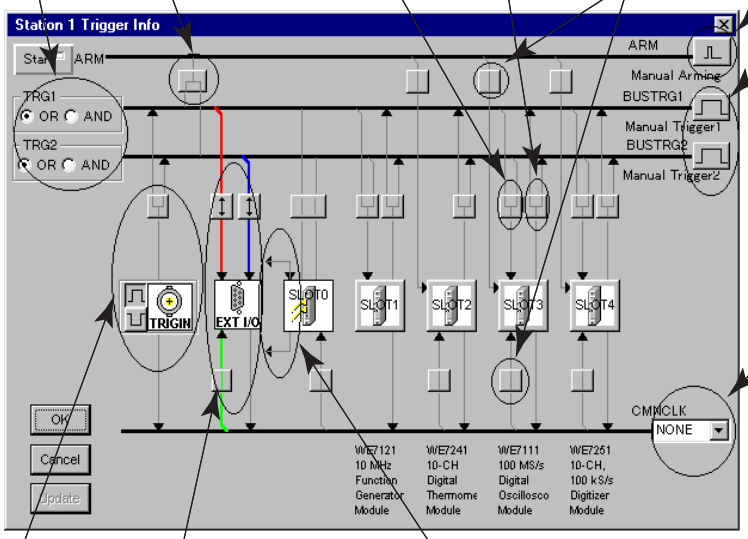
```
' Get the station handle of the station named "Station 1."
Dim hSt As Long, hMo As Long
Dim ret As Integer
ret = WeOpenStation ("Station1", hSt)
' Open the first WE7275 module with a link number of 1.
ret = WeOpenModule (hSt, "WE7275:1", 1, hMo)
Dim info As LinearScaleInfoArray
' Queries the scale information of all channels (2 channels).
ret = WeGetScaleInfo (hMo, -1, info)
```

6.7 Synchronization between Modules

There are three signal buses inside the measuring station that support synchronized operation between the modules. They are the two trigger buses (BUSTRG1, 2) for trigger synchronization and the clock bus (CMNCLK) for sharing the sampling clock. On the front panel of the station, there are external I/O terminals (TRIG IN, EXT I/O) that can be used for feeding external trigger and sampling clock signals. In addition, there is a packet trigger function on the optical interface board to trigger off of communication packets. The arming function is used to start measurement of all modules simultaneously. All of these functions can also be used to synchronize multiple stations. This section describes the API functions that support the synchronized operation of the measuring stations and modules.

The WE400/WE800 screen is shown below.

WeSetModuleBus (HANDLE hMo, BYTE InItem, BYTE OutItem, BYTE ClockItem, BYTE ArmItem)



WeSetArmingSource
WeGetArmingSource

WeSetTrigBusLogic
WeGetTrigBusLogic

WeExecManualArming

WeExecManualTrig

WeSetClockBusSource
WeGetClockBusSource

Specify the module link using WeOpenModule.

WeSetTRIG*1
WeGetTRIG*1
WeSetTRIGIN*2
WeGetTRIGIN*2

WeSetEXTIO
WeGetEXTIO

WeSetRevTrigPacket*3
WeSetRecvClockPacket*3
WeSetSndTrigPacket*3
WeSetSndClockPackt*3
WeFireTrigPacket*3
WeFireClockPacket*3

*1 For the WE500/WE900
*2 For the WE400/WE800

*3 Can only be used during optical communication or Ethernet connection on the WE400/WE800. Can only be used during optical communication on the WE500/WE900.

WeExecManualTrig

Description

Manually generates a trigger signal on the trigger bus that is common to all modules in the station. The trigger bus is used as a trigger source and the modules are triggered off of it.

Syntax

WeExecManualTrig (ByVal *hSt* As Long, ByVal *busNo* As Byte, ByVal *pulse* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle, station link handle, or broadcast handle	
busNo (IN)	Trigger bus selection	
	WE_TRG1	' Trigger bus "BUSTRG 1"
	WE_TRG2	' Trigger bus "BUSTRG 2"
pulse (IN)	Trigger pulse selection	
	WE_TRGUP	' UP
	WE_TRGDOWN	' DOWM
	WE_TRGONESHOT	' One-shot pulse

Example (Visual Basic)

```
' Generates a trigger signal on trigger bus "BUSTRG 1" using one-
' shot pulse.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeExecManualTrig (stHandle, WE_TRG1, WE_TRGONESHOT)
```

WeSetModuleBus

Description

Sets the trigger source/time base source (sampling clock) and the input/output setting of the arming signal of the module. If a module is specified that does not have a trigger source, time base source, or an arming signal input/output function, the settings are discarded.

Syntax

WeSetModuleBus (ByVal *hMo* As Long, ByVal *InItem* As Byte, ByVal *OutItem* As Byte, ByVal *InClock* As Byte, ByVal *ArmItem* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle or module link handle	
InItem (IN)	Trigger input selection	
	WE_TRGNONE	' Do not use the bus trigger signal.
	WE_TRG1	' Use bus trigger signal "BUSTRG 1."
	WE_TRG2	' Use bus trigger signal "BUSTRG 2."
OutItem (IN)	Trigger output selection	
	WE_TRGNONE	' Do not output to the trigger bus signal.
	WE_TRG1	' Output to bus trigger signal "BUSTRG 1."
	WE_TRG2	' Output to bus trigger signal "BUSTRG 2."
	WE_BOTH	' Output to both bus trigger signals "BUSTRG 1, 2."
InClock (IN)	Sampling clock input selection	
	WE_CMNCLKNONE	' Do not use the bus clock signal.
	WE_CMNCLK	' Use the sampling clock signal.
ArmItem (IN)	Arming signal input selection	
	WE_ARMNONE	' Do not use arming signal.
	WE_ARM	' Use arming signal.

Example (Visual Basic)

```
' Use bus trigger signal "BUSTRG 2", output bus trigger signal
'  "BUSTRG 1" and "BUSTRG 2", do not use the sampling clock signal,
use arming signal.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7111 module with a link number of 3.
ret = WeOpenModule (stHandle, "WE7111: 1", 3, dsoHandle)
ret = WeSetModuleBus (dsoHandle, WE_TRG2, WE_BOTH, WE_CMNCLKNONE,
WE_ARM)
```

WeGetModuleBus

Description

Gets the trigger source/time base source (sampling clock) and the input/output setting of the arming signal of the module. If a module is specified that does not have a trigger source, time base source, or an arming signal input/output function, a “0” is returned for each setting.

Syntax

WeGetModuleBus (ByVal *hMo* As Long, ByRef *InItem* As Byte, ByRef *OutItem* As Byte, ByRef *InClock* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle or module link handle	
InItem (OUT)	Trigger input setting	
	WE_TRGNONE	' Do not use the bus trigger signal.
	WE_TRG1	' Use bus trigger signal "BUSTRG 1."
	WE_TRG2	' Use bus trigger signal "BUSTRG 2."
OutItem (OUT)	Trigger output setting	
	WE_TRGNONE	' Do not output to the trigger bus signal.
	WE_TRG1	' Output to bus trigger signal "BUSTRG 1."
	WE_TRG2	' Output to bus trigger signal "BUSTRG 1."
	WE_BOTH	' Output to both bus trigger signals "BUSTRG 1, 2."
InClock (OUT)	Sampling clock input setting	
	WE_CMNCLKNONE	' Do not use the sampling clock signal.
	WE_CMNCLK	' Use the sampling clock signal.
ArmItem (OUT)	Arming signal input setting	
	WE_ARMNONE	' Do not use arming signal.
	WE_ARM	' Use arming signal.

Example (Visual Basic)

```
' Gets the trigger/time base source and arming settings of the module.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
Dim InItem As Byte
Dim OutItem As Byte
Dim InClock As Byte
Dim ArmItem As Byte
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the first WE7111 module with a link number of 3.
ret = WeOpenModule (stHandle, "WE7111: 1", 3, dsoHandle)
ret = WeGetModuleBus (dsoHandle, InItem, OutItem, InClock, ArmItem)
```

WeSetTrigBusLogic

Description

Sets the trigger condition of the trigger bus (AND/OR operation). This function is effective when multiple modules are driving the trigger bus. You can specify the trigger to occur when all trigger conditions are satisfied (AND operation) or when one of the trigger conditions is satisfied (OR operation).

Syntax

WeSetTrigBusLogic (ByVal *hst* As Long, ByVal *item* As Byte, ByVal *logic* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle or station link handle	
item (IN)	Trigger bus selection	
	WE_TRG1	' Set bus trigger "BUSTRG 1."
	WE_TRG2	' Set bus trigger "BUSTRG 2."
logic (IN)	Logic selection	
	WE_TRGAND	' Set the bus logic to AND.
	WE_TRGOR	' Se the bus logic to OR.

Example (Visual Basic)

```
' Set the logic of bus trigger "BUSTRG 1" to AND.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeSetTrigBusLogic (stHandle, WE_TRG1, WE_TRGAND)
```

WeGetTrigBusLogic

Description

Gets the trigger condition of the trigger bus.

Syntax

WeGetTrigBusLogic (ByVal *hst* As Long, ByVal *item* As Byte, ByRef *logic* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle	
item (IN)	Trigger bus selection	
	WE_TRG1	' Get trigger bus "BUSTRG 1. "
	WE_TRG2	' Get trigger bus "BUSTRG 2. "
logic (OUT)	Trigger bus logic	
	WE_TRGAND	' Bus logic is AND.
	WE_TRGOR	' Bus logic is OR.

Example (Visual Basic)

```
' Get the bus logic of bus trigger "BUSTRG 1."
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim logic As Byte
ret = WeGetTrigBusLogic (stHandle, WE_TRG1, logic)
```

WeSetEXTIO

Description

Sets the input/output setting of the trigger signal I/O pin and the time base signal I/O pin of the EXT I/O connector located on the front panel of the station. These signal pins are bi-directional and can (1) pass the external signals to the trigger and clock buses in the station, (2) output the trigger signal or the clock, and (3) provide input signals for other stations. In effect, multiple stations can be synchronized.

Syntax

WeSetEXTIO (ByVal *hst* As Long, ByVal *trig1* As Byte, ByVal *trig2* As Byte, ByVal *clock* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle or station link handle	
trig1 (IN)	Input/output setting of the trigger signal I/O 1 pin of the EXT I/O terminal	
	WE_TRGIN	' Use as an input pin. Pass the external signal ' to the trigger bus "BUSTRG 1."
	WE_TRGOUT	' Use as an output pin. ' Output the trigger bus signal "BUSTRG 1."
trig2 (IN)	Input/output setting of the trigger signal I/O 2 pin of the EXT I/O terminal	
	WE_TRGIN	' Use as an input pin. Pass the external signal ' to the trigger bus "BUSTRG 2."
	WE_TRGOUT	' Use as an output pin. ' Output the trigger bus signal "BUSTRG 2."
clock (IN)	Input/output setting of the time base signal I/O pin of the EXT I/O terminal	
	WE_CMNCLKOUT	' Use as an output pin.
	WE_CMNCLKIN	' Use as an input pin.

Example (Visual Basic)

```
' Set the trigger signal I/O 1 pin to output, trigger signal I/O 2
' pin to input, and the time base signal I/O pin to input a common
' clock.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeSetEXTIO (stHandle, WE_TRGOUT, WE_TRGIN, WE_CMNCLKIN)
```

WeGetEXTIO

Description

Gets the input/output setting of the trigger I/O pin and the timebase I/O pin of the EXT I/O connector located on the front panel of the station.

Syntax

WeGetEXTIO (ByVal *hst* As Long, ByRef *trig1* As Byte, ByRef *trig2* As Byte, ByRef *clock* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle	
trig1 (OUT)	Input/output setting of the trigger1 signal I/O 1 pin of the EXT I/O terminal	
	WE_TRGIN	Use as an input pin. Pass the external signal ' to the trigger bus "BUSTRG 1."
	WE_TRGOUT	' Use as an output pin. ' Output the trigger bus signal "BUSTRG 1."
trig2 (OUT)	Input/output setting of the frigger2 signal I/O 2 pin of the EXT I/O terminal	
	WE_TRGIN	' Use as an input pin. Pass the external signal ' to the trigger bus "BUSTRG 2."
	WE_TRGOUT	' Use as an output pin. ' Output the trigger bus signal "BUSTRG 2."
clock (OUT)	Input/output setting of the time base signal I/O pin of the EXT I/O terminal	
	WE_CMNCLKOUT	' Use as an output pin.
	WE_CMNCLKIN	' Use as an input pin.

Example (Visual Basic)

```
' Gets the input/output setting of the trigger signal I/O 1,
' trigger signal I/O 2, and
' time base signal I/O pins of the EXT I/O terminal.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim trig1 As Byte, trig2 As Byte
Dim clock As Byte
ret = WeGetEXTIO (stHandle, trig1, trig2, clock)
```

WeSetTRIG

Description

Sets the destination and polarity of the trigger bus signal entering the TRIG terminal located on the front panel of the measuring station.

Syntax

WeSetTRIG (ByVal *hst* As Long, ByVal *item* As Byte, ByVal *polarity* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle or station link handle	
item (IN)	Select the trigger bus destination of the signal entering the TRIG terminal.	
	WE_TRGNONE	' Do not pass signal to the trigger bus.
	WE_TRG1	' Pass the signal to trigger bus "BUSTRG 1. "
	WE_TRG2	' Pass the signal to trigger bus "BUSTRG 2. "
	WE_BOTH	' Pass the signal to trigger buses "BUSTRG 1, 2. "
	WE_TRG1OUT	' Output the signal to trigger bus "BUSTRG 1. "
	WE_TRG2OUT	' Output the signal to trigger bus "BUSTRG 2. "
polarity (IN)	Select the polarity of the input signal at the TRIG terminal	
	WE_TRGPOS	' Input/output the signal as is.
	WE_TRGNEG	' Input/output the signal after inverting it.

Example (Visual Basic)

```
' Pass the trigger signal entering the TRIG terminal to trigger
buses
' "BUSTRG 1" and "BUSTRG 2" and set the polarity to positive.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeSetTRIG (stHandle, WE_BOTH, WE_TRGPOS)
```

Note:

WE_TRG1OUT, WE_TRG2OUT, and WE_CMNCLKOUT are ignored on the WE400 or WE800. (The function ends normally, but does not work.)

WeGetTRIG

Description

Gets the destination and polarity of the trigger input signal entering the TRIG terminal located on the front panel of the station.

Syntax

WeGetTRIG (ByVal *hst* As Long, ByRef *item* As Byte, ByRef *polarity* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle
item (OUT)	Select the trigger bus destination of the signal entering the TRIG terminal.
	WE_TRGNONE ' Do not pass signal to the trigger bus.
	WE_TRG1 ' Pass the signal to trigger bus "BUSTRG 1."
	WE_TRG2 ' Pass the signal to trigger bus "BUSTRG 2."
	WE_BOTH ' Pass the signal to trigger buses "BUSTRG 1, 2."
	WE_TRG1OUT ' Output the signal to trigger bus "BUSTRG 1. "
	WE_TRG2OUT ' Output the signal to trigger bus "BUSTRG 2. "
polarity (OUT)	Select the polarity of the input signal at the TRIG terminal
	WE_TRGPOS ' Input/output the signal as is.
	WE_TRGNEG ' Input/output the signal after inverting it.

Example (Visual Basic)

```
' Get the destination and polarity of the trigger bus signal
' entering the TRIG terminal.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim item As Byte
Dim pol As Byte
ret = WeGetTRIG (stHandle, item, pol)
```

Note:

WE_TRG1OUT, WE_TRG2OUT, and WE_CMNCLKOUT are ignored on the WE400 or WE800. (The function ends normally, but does not work.)

WeSetTRIGIN

Description

Sets the destination and polarity of the trigger bus signal entering the TRIG IN terminal located on the front panel of the measuring station.

Syntax

WeSetTRIGIN (ByVal *hst* As Long, ByVal *item* As Byte, ByVal *polarity* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle or station link handle
item (IN)	Select the trigger bus destination of the signal entering the TRIG IN terminal.
	WE_TRGNONE ' Do not pass signal to the trigger bus.
	WE_TRG1 ' Pass the signal to trigger bus "BUSTRG 1. "
	WE_TRG2 ' Pass the signal to trigger bus "BUSTRG 2. "
	WE_BOTH ' Pass the signal to trigger buses "BUSTRG 1, 2. "
polarity (IN)	Select the polarity of the input signal at the TRIG IN terminal
	WE_TRGPOS ' Input the signal as is.
	WE_TRGNEG ' Input the signal after inverting it.

Example (Visual Basic)

```
' Pass the trigger signal entering the TRIG IN terminal to trigger
buses
' "BUSTRG 1" and "BUSTRG 2" and set the polarity to positive.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeSetTRIGIN (stHandle, WE_BOTH, WE_TRGPOS)
```

Note:

Use this function when you need to maintain compatibility with the APIs for the WE400 and WE800. Use the WeSetTRIG function if you are creating a new program.

WeGetTRIGIN

Description

Gets the destination and polarity of the trigger input signal entering the TRIG IN terminal located on the front panel of the station.

Syntax

WeGetTRIGIN (ByVal *hst* As Long, ByRef *item* As Byte, ByRef *polarity* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle	
item (OUT)	TRIG IN terminal's bus	
	WE_TRGNONE	' Do not pass signal to the trigger bus.
	WE_TRG1	' Pass the signal to trigger bus "BUSTRG 1."
	WE_TRG2	' Pass the signal to trigger bus "BUSTRG 2."
	WE_BOTH	' Pass the signal to trigger buses "BUSTRG 1, 2."
polarity (OUT)	Polarity of the polarity of the input signal at the TRIG IN terminal	
	WE_TRGPOS	' Input the signal as is.
	WE_TRGNEG	' Input the signal after inverting it.

Example (Visual Basic)

```
' Get the destination and polarity of the trigger bus signal
' entering the TRIG IN terminal.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim item As Byte
Dim pol As Byte
ret = WeGetTRIGIN (stHandle, item, pol)
```

Note:

Use this function when you need to maintain compatibility with the APIs for the WE400 and WE800. Use the WeGetTRIG function if you are creating a new program.

WeSetClockBusSource

Description

Sets the time base source that is output to the common clock bus (CMNCLK). The clock can only have one source.

Syntax

WeSetClockBusSource (ByVal *hst* As Long, ByVal *source* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle or station link handle
source (IN)	Time base source selection
	WE_CMNCLKSRC_NONE ' No source
	WE_CMNCLKSRC_TRIGIN ' TRG IN terminal of the front panel
	WE_CMNCLKSRC_EXT.IO ' EXT I/O terminal of the front panel
	WE_CMNCLKSRC_SLOT0/WE_CMNCLKSRC_SLOT1/
	WE_CMNCLKSRC_SLOT2/WE_CMNCLKSRC_SLOT3/
	WE_CMNCLKSRC_SLOT4/WE_CMNCLKSRC_SLOT5/
	WE_CMNCLKSRC_SLOT6/WE_CMNCLKSRC_SLOT7/
	WE_CMNCLKSRC_SLOT8 ' Slot 0 to slot 8

Example (Visual Basic)

```
' Set the time base output of the module in slot 1 to be the time
' base source.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeSetClockBusSource (stHandle, WE_CMNCLKSRC_SLOT1)
```

WeGetClockBusSource

Description

Gets the time base source that is currently being output to the common clock bus.

Syntax

WeGetClockBusSource (ByVal *hst* As Long, ByRef *source* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN) Station handle

source (OUT) Timebase source

```
WE_CMNCLKSRC_NONE       ' No source
WE_CMNCLKSRC_TRIGIN     ' TRG IN terminal of the front panel
WE_CMNCLKSRC_EXT.IO     ' EXT I/O terminal of the front panel
WE_CMNCLKSRC_SLOT0/WE_CMNCLKSRC_SLOT1/
WE_CMNCLKSRC_SLOT2/WE_CMNCLKSRC_SLOT3/
WE_CMNCLKSRC_SLOT4/WE_CMNCLKSRC_SLOT5/
WE_CMNCLKSRC_SLOT6/WE_CMNCLKSRC_SLOT7/
WE_CMNCLKSRC_SLOT8      ' Slot 0 to slot 8
```

Example (Visual Basic)

```
' Get the time base source setting.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim item As Byte
ret = WeGetClockBusSource (stHandle, item)
```

WeSetRcvTrigPacket

Description

Sets the stations that will receive the trigger packets that are used for packet triggering. Up to eight stations can be specified. WeFireTrigPacket () can be used to generate a trigger packet and send it to the stations specified here. The trigger packet can also be generated from the specified trigger source station using WeSetTrigPacket ().

Syntax

WeSetRcvTrigPacket (ByVal *hst* As Long, ByVal *num* As Integer, *name* As PacketListArray) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN) Station handle, station link handle, or broadcast handle

num (IN) Number of stations to specify (up to 8).

name (IN) Pointer to the station name structure

```
Type PacketList                         ' Station name structure
  name As String * MaxStationName       ' Station name
End Type
```

Example (Visual Basic)

```
' Set station 1 and station 2 to receive the packet triggers.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim packet As PacketListArray
packet.list (0).name = "Station 1"
packet.list (1).name = "Station 2"
ret = WeSetRcvTrigPacket (stHandle, 2, packet)
```

WeSetSndTrigPacket

Description

Sets the trigger packet source stations to be used for packet triggering. Up to eight stations can be specified. When trigger bus “BUSTRG 1” or “BUSTRG 2” becomes active in the source station, the station generates the trigger packet. The receiving station outputs a trigger signal according to the trigger source to its own trigger bus “BUSTRG 1” or “BUSTRG 2.”

Syntax

WeSetSndTrigPacket (ByVal *hst* As Long, ByVal *num* As Integer, *name* As PacketListArray) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle, station link handle, or broadcast handle
num (IN)	Number of stations (up to 8).
name (IN)	Pointer to the station name structure

```
Type PacketList                          ' Station name structure
  name As String * MaxStationName        ' Station name
End Type
```

Example (Visual Basic)

```
' Set Station 1 and Station 2 to send packet triggers.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim packet As PacketListArray
packet.list (0).name = "Station 1"
packet.list (1).name = "Station 2"
ret = WeSetSndTrigPacket (stHandle, 2, packet)
```

WeFireTrigPacket

Description

Generates a trigger packet using software for packet triggering. The trigger packets are sent to the measuring stations specified by WeSetRcvTrigPacket ().

Syntax

WeFireTrigPacket (ByVal *hst* As Long, ByVal *item* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle, station link handle, or broadcast handle
item (IN)	Trigger bus selection. The trigger signal is output to the receiving station trigger bus specified by this parameter.
	WE_TRG1 ' Output to trigger bus "BUSTRG 1."
	WE_TRG2 ' Output to trigger bus "BUSTRG 2."

Example (Visual Basic)

```
' Generate a trigger packet for Station 1.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeFireTrigPacket (stHandle, WE_TRG1)
```

WeSetSndClockPacket

Description

Set the time base packet source for packet time base. The specified measuring station generates time base packets using its own time base signal. Up to eight time base packet measuring stations can be specified.

Syntax

WeSetSndClockgPacket (ByVal *hst* As Long, ByVal *num* As Integer, *name* As PacketListArray) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle, station link handle, or broadcast handle
num (IN)	Number of stations (up to 8)
name (IN)	Pointer to the station name structure
	Type PacketList ' Station name structure
	name As String * MaxStationName ' Station name
	End Type

Example (Visual Basic)

```
' Set Station 1 and Station 2 to send time base packets.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim packet As PacketListArray
packet.list (0).name = "Station 1"
packet.list (1).name = "Station 2"
ret = WeSetSndClockPacket (stHandle, 2, packet)
```

WeSetRcvClockPacket

Description

Sets the time base packet receiving measuring for packet time base. Up to eight receiving measuring stations can be specified. The measuring station receiving the time base packet generates one pulse of time base signal on its own clock bus for each packet.

Syntax

WeSetRcvClockgPacket (ByVal *hst* As Long, ByVal *num* As Integer, *name* As PacketListArray) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle, station link handle, or broadcast handle
num (IN)	Number of stations (up to 8)
name (IN)	Pointer to the station name structure

```
Type PacketList                          ' Station name structure
  name As String * MaxStationName        ' Station name
End Type
```

Example (Visual Basic)

```
' Set Station 1 and Station 2 to receive time base packets.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim packet As PacketListArray
packet.list (0).name = "Station 1"
packet.list (1).name = "Station 2"
ret = WeSetRcvClockPacket (stHandle, 2, packet)
```

WeFireClockPacket

Description

Generates time base packets using software for packet time base.

Syntax

WeFireClockgPacket (ByVal *hst* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle, station link handle, or broadcast handle

Example (Visual Basic)

```
' Generates a time base packet.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeFireClockgPacket (stHandle)
```

WeOutputEXTIOEvent

Description

Outputs an event signal using software to the event output signal pin of the EXT I/O terminal located on the front panel of the measuring station.

Syntax

WeOutputEXTIOEvent (ByVal *hst* As Long, ByVal *pulse* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hst (IN)	Station handle, station link handle, or broadcast handle	
pulse (IN)	Output selection	
	WE_EXTIO_OFF	' OFF
	WE_EXTIO_ON	' ON
	WE_EXTIO_PULSE	' Pulse

Example (Visual Basic)

```
' Turn ON the event signal output.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeOutputEXTIOEvent (stHandle, WE_EXTIO_ON)
```

Note:

If this function is used on the WE500 or WE900, the output of DIO pin 0 of the EXT. I/O is turned ON, and STATUS LED A illuminates in orange.

WeExecManualArming

Description

Manually generates an arming signal.

Syntax

WeExecManualArming (ByVal *hSt* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle, station link handle, or broadcast handle

Example (Visual Basic)

```
' Manually generate an arming signal.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeExecManualArming (stHandle)
```

WeSetArmingSource

Description

Sets the arming signal source.

Syntax

WeSetArmingSource (ByVal *hSt* As Long, ByVal *item* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle, station link handle, or broadcast handle	
item (IN)	Arming signal source selection	
	WE_TRGNONE	' Do not select arming source.
	WE_TRG1	' Set bus trigger signal "BUSTRG 1" as the ' arming signal source.
	WE_TRG2	' Set bus trigger signal "BUSTRG 2" as the ' arming signal source.

Example (Visual Basic)

```
' Set the bus trigger signal "BUSTRG 1" to the arming signal
' source.
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeSetArmingSource (stHandle, WE_TRG1)
```

WeGetArmingSource

Description

Get the arming signal source setting.

Syntax

WeGetArmingSource (ByVal *hSt* As Long, ByRef *item* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle, station link handle, or broadcast handle	
item (OUT)	Arming source selection	
	WE_TRGNONE	' Do not select arming signal source.
	WE_TRG1	' Set bus trigger signal "BUSTRG 1" as the ' arming signal source.
	WE_TRG2	' Set bus trigger signal "BUSTRG 2" as the ' arming signal source.

Example (Visual Basic)

```
' Gets the arming signal source setting.
Dim stHandle As Long
Dim ret As Integer
Dim item As Byte
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeGetArmingSource (stHandle, item)
```

6.8 GUI Control

WeShowModuleWindow

Description

Displays the module GUI Window for controlling the module.

Syntax

WeShowModuleWindow (ByVal *hMo* As Long, ByRef *hWnd* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
hWnd (OUT)	GUI window handle

Note:

The operation panel that is displayed using this API can be used only to make setting changes. It cannot be used to start/stop the module. In addition, for operation panels that display measured values (instantaneous values), the displayed values are not updated.

Example (Visual Basic)

```
' Open the WE7111 GUI panel.
Dim hWnd As Long
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with  a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeShowModuleWIndow (dsoHandle, hWnd)
' Move the window position to x = 520 and y = 220.
ret = SetWindowPos (hWnd, 0, 520, 220, 0, 0, 1)
```

WeCloseModuleWindow

Description

Closes the module GUI Window used to control the module.

Syntax

WeCloseModuleWindow (ByVal *hMo* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle

Example (Visual Basic)

```
' Close the WE7111 GUI panel.
Dim hWnd As Long
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeCloseModuleWindow (dsoHandle)
```

WeIsModuleWindow

Description

Queries whether or not the module GUI window for controlling the module is open.

Syntax

WeIsModuleWindow (ByVal *hMo* As Long, ByRef *sw* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
sw (OUT)	Returns 1 if the module window is open, 0 if it is not.

Example (Visual Basic)

```
' Query whether or not the WE7111 GUI panel is open.
Dim stHandle As Long
Dim ret As Integer
Dim dsoHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
Dim sw As Byte
ret = WeIsModuleWIndow (dsoHandle, sw)
```

WeShowTrigWindow

Description

Displays the trigger source/timebase source/arming setting dialog box that is used to set the module synchronization function within the measuring station.

Syntax

WeShowTrigWindow (ByVal *hSt* As Long, ByRef *hWnd* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN) Station handle

hWnd (OUT) Trigger source/time base source/arming setting dialog box handle. This can use to change the display position of the setting dialog box.

Example (Visual Basic)

```
' Open the trigger source/time base source/arming setting dialog box.
Dim hWnd As Long
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim hWnd As Long
ret = WeShowTrigWIndow (stHandle, hWnd)
ret = SetWIndowPos (hWnd, 0, 520, 220, 0, 0, 1)
```

WeCloseTrigWindow

Description

Closes the trigger source/time base source/arming setting dialog box that is used to set the module synchronization function within the measuring station.

Syntax

WeCloseTrigWindow (ByVal *hSt* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN) Station handle

Example (Visual Basic)

```
' Close the trigger source/time base source/arming setting dialog box.
Dim hWnd As Long
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
ret = WeCloseTrigWindow (stHandle)
```

WeIsTrigWindow

Description

Queries whether or not the trigger source/time base source/arming setting dialog box, that is used to set the module synchronization function within the station, is open.

Syntax

WeIsTrigWindow (ByVal *hSt* As Long, ByRef *sw* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hSt (IN)	Station handle
sw (OUT)	Returns 1 if the trigger source/time base source/arming setting dialog box is open, 0 if not.

Example (Visual Basic)

```
' Query whether or not the trigger source/time base source/arming
' setting dialog box is open.
Dim hWnd As Long
Dim stHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
Dim sw As Byte
ret = WeIsTrigWIndow (stHandle, sw)
```

WeShowLinearScaleWindow

Description

Displays the convert scale dialog box.

Syntax

WeShowLinearScaleWindow (ByVal *hMo* As Long, ByRef *hWnd* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
hWnd (OUT)	Convert scale dialog box handle Used when changing the position of the dialog box, for example.

Example (Visual Basic)

```
' Display the convert scale dialog box for the WE7251 module.
Dim hSt As Long, hMo As Long
Dim ret As Integer
Dim hWnd As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
' Open the WE7251 module with a link number of 2.
ret = WeOpenModule (hSt, "WE7251:1", 2, hMo)
ret = WeShowLinearScaleWindow (hMo, hWnd)
' Calls the WIN32API and moves the window position to x = 520, y =
220.
ret = SetWindowPos(hWnd, 0, 520, 220, 0, 0, 1)
```

WeCloseLinearScaleWindow

Description

Closes the convert scale dialog box.

Syntax

WeCloseLinearScaleWindow (ByVal *hMo* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN) Module handle

Example (Visual Basic)

```
' Close the convert scale dialog box for the WE7251 module.
Dim hSt As Long, hMo As Long
Dim ret As Integer
Dim hWnd As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
' Open the WE7251 module with a link number of 2.
ret = WeOpenModule (hSt, "WE7251:1", 2, hMo)
ret = WeShowLinearScaleWindow (hMo, hWnd)
ret = WeCloseLinearScaleWindow (hMo)
```

WeIsLinearScaleWindow

Description

Queries whether or not the convert scale dialog box is open.

Syntax

WeIsLinearScaleWindow (ByVal *hMo* As Long, ByRef *sw* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN) Module handle
sw (OUT) Returns 1 if the module operation panel is displayed, 0 if it is not.

Example (Visual Basic)

```
' Query whether or not the convert scale dialog box for the WE7251
module is open.
Dim hSt As Long, hMo As Long
Dim ret As Integer
Dim hWnd As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", hSt)
' Open the WE7251 module with a link number of 2.
ret = WeOpenModule (hSt, "WE7251:1", 2, hMo)
Dim sw As Byte
ret =WeIsLinearScaleWindow(hMo, sw)
```

6.9 Measurement Control

WeStart

Description

Starts the operation of the measurement module.

Syntax

WeStart (ByVal *hHandle* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Note:

This function only issues the start command to a module. For example,it does not carry out operations such as synchronize to the end of an acquisition on the acquisition module. If such operations are necessary, poll (monitor) for the end of the execution (WeIsRun function) or use a function that synchronizes to the end of an acquisition (WeStartSingle ()).

Parameters

hHandle (IN) Module handle, station handle, module link handle, or station link handle.

Example (Visual Basic)

```
' Start WE7111.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeStart (dsoHandle)
```

WeStop

Description

Stops the operation of the measurement module.

Syntax

WeStop (ByVal *hHandle* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hHandle (IN) Module handle, station handle, module link handle, or station link handle.

Example (Visual Basic)

```
' Stop WE7111.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
  ......
ret = WeStop (dsoHandle)
```

WeStartEx

Description

Starts the operation of the measurement module. This function can be used to simply issue the start command (same as WeStart ()) or or have the module notify the end of the acquisition with an event (same as WeStartWithEvent ()) by specifying the operation mode. In the event notification mode, event notification for each block and event notification according to the logic block during the free run mode are possible (unlike WeStartWithEvent ()).

Syntax

WeStartEx (ByVal *hMo* As Long, ByVal *blockLen* As Long, ByVal *blockCount* As Byte, ByVal *acqCount* As Long, ByVal *mode* As Integer, ByRef *evHandle* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
blockLen (IN)	Number of data points per block (record length)
blockCount (IN)	Number of memory partitions (blocks) Number of blocks can only be specified in powers of two's. Make sure the following equation is satisfied: "blockLen* (2 to the blockCount power) ≤ Acquisition memory length in the module."
acqCount (IN)	Number of acquisitions. The data acquisition operation terminates after acquiring the amount of data specified by this number. If 0 is specified, the operation continues until the user issues the stop command.
mode (IN)	Acuquisition operation mode WE_ACQ_NO_EVENT ' Do not generate events WE_ACQ_BLOCK_EVENT ' Generate events for each block WE_ACQ_STOP_EVENT ' Generate events after acquiring all ' blocks and the operation stops
evHandle (OUT)	Event handle Used with WeStopEx function.

Note:

acqCount is ignored when WE_ACQ_BLOCK_EVENT is specified. In other words, data acquisition operation continues until the stop command is issued. When the acquisition mode is set to free run, blockCount is ignored.

Example (Visual Basic)

- Example in which data are read using the event notification in the trigger mode on the WE7251

```
' Initialization procedure
  ret = WeInit(WeEvent1.hwnd, "Optical devicename=we7036", WE_CONTROLLER)
   ' Get the station handle of the station named "Station 1."
   ret = WeOpenStation("Station1", hSt)
   ' Turn ON Station 1.
   ret = WePower(hSt, WE_ON)
   ' Open module WE7251 with a link number of 1.
   ret = WeOpenModule(hSt, "WE7251:1", 1, hMo7251)
  ' Enable aquisition on CH1.
  ret = WeSetControl(hMo7251, "CH1:On", "On")
  ' Set the aquisition mode to "Triggered."
  ret = WeSetControl(hMo7251, "Acquisition Mode", "Triggered")
  ......

' Start procedure
  ' Set the sampling interval to 0.001 s.
  ret = WeSetControl(hMo7251, "Sampling Interval", "0.001")
  ' Set the number of memory partitions to 2 (2 to the 1st power)
  ' and the number of measurements to 2.
  ' Request that the WeEvent be issued when two blocks of data
  ' (1000 points) are acquired.
  ' Therefore, an event is notified after 2 s (sampling interval of
  ' 0.001 x block length of 1000 x number of measurments of 2).
  ' Start measurement.
  ret = WeStartEx(hMo7251, 1000, 1, 2, WE_ACQ_STOP_EVENT, evHandle)
   ......

' WeEvent handler
  Dim recSize As Long
  Dim buf() As Double
  Dim sparam As ScalingParam
    ' If it is an acquisition stop event,
    If ev = WE_EV_MEASEND Then
      sparam.a = 1
      sparam.b = 0
      ' Since only the CH1 data are read, the buffer size is
      ' 1000 points x 8 bytes (Double type).
      recSize = 1000*8
      ' Allocate a buffer for 1000 points.
      ReDim buf(1000) As Double
      ' Read the data from CH1 block 0.
      ' Read the voltage.
      ret = WeGetScaleData(hMo7251, 1, 0, sparam, recSize, WE_DOUBLE,
buf(0))
      ' Display the data of block 0 or make an analysis.
      ......
      ' Read the data from CH1 block 1.
      ' Read the voltage.
      ret = WeGetScaleData(hMo7251, 1, 1, sparam, recSize, WE_DOUBLE,
buf(0))
      ......
    End If

' Stop procedure
  ' Stops the measurement and releases the event.
  ret = WeStopEx(hMo7251, evHandle)
  ......
```

- Example in which data are read using the event notification in the free run mode on the WE7251

```
' Initialization procedure
  ret = WeInit (WeEvent1.hwnd, "Optical devicename=we7036", WE_CONTROLLER)
   ' Get the station handle of the station named "Station 1."
   ret = WeOpenStation ("Station1", hSt)
   ' Turn ON Station 1.
   ret = WePower (hSt, WE_ON)
   ' Open module WE7251 with a link number of 1.
   ret = WeOpenModule (hSt, "WE7251:1", 1, hMo7251)
  ' Enable aquisition on CH1.
  ret = WeSetControl (hMo7251, "CH1:On", "On")
  ' Set the aquisition mode to "Free Run."
  ret = WeSetControl (hMo7251, "Acquisition Mode", "Free Run")
  ......

' Start procedure
  ' Set the sampling interval to 0.01 s.
  ret = WeSetControl (hMo7251, "Sampling Interval", "0.01")
  ' Request that the WeBlock event be issued when 100 points of
  ' data are acquired.
  ' Therefore, an event is notified every 1 s (sampling interval of
  ' 0.01 x block length of 100).
  ' In the free run mode, the number of blocks is ignored and the
  ' number of acquisitions is set to infinity.
  ret = WeStartEx (hMo7251, 100, 0, 0, WE_ACQ_BLOCK_EVENT, evHandle)
  ' Start measurement.
  ......

' WeBlock handler
  Dim recSize As Long
  Dim buf () As Double
  Dim sparam As ScalingParam
  sparam.a = 1
  sparam.b = 0
  ' Since only the CH1 data are read, the buffer size is 100 points
  ' x 8 bytes (Double type).
  recSize = 100*8
  ' Allocate a buffer for 100 points.
  ReDim buf (100) As Double
  ' Read the voltage.
  ret = WeGetScaleData (hMo7251, 1, blockNo, sparam, recSize,
WE_DOUBLE, buf (0))
  ' Display the data or make an analysis.
  ......

' Stop procedure
  ' Stops the measurement and releases the event.
  ret = WeStopEx (hMo7251, evHandle)
  ......
```

WeStopEx

Description

Stops the operation of the measurement module. Use this function to stop the operation only if the acquisition was started with the WeStartEx function.

Syntax

WeStopEx (ByVal *hMo* As Long, ByVal *evHandle* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
evHandle (IN)	Event handle to release Specify the event handle that was obtained when WeStartEx () was called.

Example (Visual Basic)

```
ret = WeStartEx(dsoHandle, 1000, 0, 0, WE_ACQ_BLOCK_EVENT, evHandle)
  ......
ret = WeStopEx(hMoDl, evHandle)
```

WeStartSingle

Description

Data are acquired once on the measurement module. The function terminates when the data acquisition is complete.

Syntax

WeStartSingle (ByVal *handle* As Long, ByVal *timeout* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Note:

This function does not return until the data acquisition completes. The program calling this function will have to wait both if the trigger wait time is long or if the acquisition takes a long time.

Parameters

hHandle (IN)	Module handle, station handle, module link handle, or station link handle.
timeOut (IN)	Timeout value (seconds). Specify the time to wait for the detection the end of an acquisition. The program will be in a deadlock if this parameter is not specified and the trigger does not occur.

Example (Visual Basic)

```
' Start the WE7111 and wait 10 seconds for the acquisition to finish.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeStartSingle (dsoHandle, 10)
```

WeStartWithEvent

Description

Start the measurement module that is to acquire the data, and notify the end of the acquisition with an event. This function returns immediately after starting the measurement module and asynchronously generates the system defined event, WE_EV_MEASEND, when the acquisition terminates. Because the end of the acquisition notified by an event, the user program does not need to wait even when the trigger wait or the acquisition takes a long time. To perform the next acquisition, call this function again after processing the event. For details relating to the processing of events, see chapter 5.

Syntax

WeStartWithEvent (ByVal *hHandle* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Note:

On acquisition modules that have memory partition functions, the event is notified after acquiring all blocks. If you need an event notification for each block, use the WeStartEx function. In addition, this function cannot generate events when the module is in the free run mode in which the acquisition continues until the user stops the operation. Use WeStartEx in this case.

Parameters

hHandle (IN) Module handle, station handle, module link handle, or station link handle.

Example (Visual Basic)

```
' Start WE7111 and request event notification.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeStartWithEvent (dsoHandle)
```

WeIsRun

Description

Queries the execution state of the measurement module (run/stop).

Syntax

WeIsRun (ByVal *hMo* As Long, ByRef *status* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
status (OUT)	Run state (1) or stop state (0)

Example (Visual Basic)

```
' Queries the execution state of WE7111.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeStart (dsoHandle)
Dim status As Byte
Do
  ret = WeIsRun (dsoHandle, status)
  If status=0 Then
   Exit Do
  End If
Loop
```

WeLatchData

Description

Issues the latch command that specifies the range of acquisition data to retrieve during the free run mode. For details, see chapter 4.

Syntax

WeLatchData (ByVal *hMo* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle

Example (Visual Basic)

```
' Issue the latch command to WE7241.
Dim stHandle As Long
Dim tcHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", tcHandle)
' Open the WE7241 module with link number of 2.
ret = WeOpenModule (stHandle, "WE7241: 1", 2, tcHandle)
ret = WeStart (tcHandle)
  ......
ret = WeLatchData (tcHandle)
```

WeGetAcqDataInfo

Description

Gets the additional information pertaining to the acquisition data of the measurement module.

Syntax

WeGetAcqDataInfo (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Integer, ByRef *info* As AcqDataInfo, ByRef *infoNum* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the number exceeds the valid number of blocks, an error occurs.

Parameters

hMo (IN) Module handle

ch (IN) Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.

blockNo (IN) Block number. For details, see section 4.2.

info (OUT) Data information pointer

```
Type AcqDataInfo
  channel As Integer      ' CH number (1 or greater)
  dataType As Integer     ' Acquisition data type
                                ' WE_NULL      No parameter
                                ' WE_UBYTE     8-bit unsigned integer
                                ' WE_SBYTE     8-bit signed integer
                                ' WE_UWORD     16-bit unsigned integer
                                ' WE_SWORD     16-bit signed integer
                                ' WE_ULONG     32-bit unsigned integer
                                ' WE_SLONG     32-bit signed integer
                                ' WE_FLOAT     32-bit real number
                                ' WE_DOUBLE    64-bit real number
  blockNum As Integer   ' Valid number of blocks
  startBit As Integer       ' Start position of the valid bit when the data
                            ' type is integer (0 bit or higher).
  effectiveBit As Integer ' Valid bit length when the data type is
                            ' integer (0 represents the length up to the
                            ' highest bit)
  trigActive As Integer     ' Trigger active (0/1)
  record As Long            ' Record length (Number of samples)
  recordLen As Long         ' Display record length
  trigPosition As Long      ' Trigger position (Record position, with zero
                            ' as the origin)
  interval As Double        ' Sampling interval (s)
  vResolution As Double ' Scaling factor for converting physical values
  vOffset As Double         ' Off set for converting physical values
  trigLevel As Double       ' Trigger level (in the same dimension as the
                            ' physical values)
  trigWidth As Double       ' Trigger width (in the same dimension as the
                            ' physical values)
  plusOverData As Double' Upper limit
                            ' (in the same dimension as the acquisition data)
  minusOverData As Double' Lower limit
                            ' (in the same dimension as the acquisition data)
  nonData As Double         ' Illegal value (in the same dimension as the
                            ' acquisition data)
  dispMaxData As Double ' Maximum display value
                            ' (in the same dimension as the acquisition data)
  dispMinData As Double ' Minimum display value
                            ' (in the same dimension as the acquisition data)
End Type
```

infoNum (OUT) Number of AcqDataInfo that is actually retrieved.

Note:

When the acquisition mode is set to free run, the block number, blockNum, is always “1.”
The information attached to the acquired data varies depending on the module as follows.

• WE7081(Acquisition Mode)
Description:Information regarding raw data
dataType:Depends on the channel definition
startBit: Depends on the channel definition
effectiveBit: Depends on the channel definition
trigActive: When the trigger type is not Off: 1
When set to other or BUSTRG: 0
record:Record length setting
recordLen:Record-1
trigPosition:Pre-trigger setting
interval:Sampling interval setting
vResolution:1.0
vOffset:0.0
trigLevel:Trigger level setting
trigWidth:Meaningless
plusOverData:Meaningless
minusOverData:Meaningless
nonData:Meaningless
dispMaxData:Meaningless
dispMinData:Meaningless

• WE7111
Description: Information regarding the raw data
dataType: WE_SWORD in the Average mode, WE_SBYTE for all other modes.
blockNum: 1
startBit: Meaningless
effectiveBit: Meaningless
trigActive: 1 for data from the channel which is the trigger source, 0 otherwise. Meaningless when the trigger source is set to BUSTRG.
record: Record length setting value
recordLen: Record length, depends on the time/div setting.
trigPosition: Trigger position, depends on the trigger position setting.
interval: Depends on the time/div setting.
vResolution: Depends on the V/div and probe attenuation settings.
vOffset: Depends on the offset voltage and probe attenuation settings
trigLevel: Depends on the trigger level and probe attenuation settings.
trigWidth: Meaningless
plusOverData: 0x3F00 in the Average mode, 0xFE for all other modes.
minusOverData: 0xC000 in the Average mode, 0x01 for all other modes
nonData: 0x8000 in the Average mode, 0xFE for all other modes.
dispMaxData: 0x3E00 in the Average mode, 0xFD for all other modes.
dispMinData: 0xC100 in the Average mode, 0x02 for all other modes

• WE7116
Description: Information regarding A/D raw data
DataType: WE_SWORD
blockNum: Number of memory partitions
StartBit: Meaningless.
EffectiveBit: Meaningless.
TrigActive: When the trigger type is not Off: 1
When set to other or BUSTRG: 0
Record: Record length setting
RecordLen: Record -1
TrigPosition: Pre-trigger setting
Interval: Sampling interval setting
VResolution: Scaling factor for converting physical values
VOffset: Offset for converting physical values
TrigLevel: Trigger level setting
TrigWidth: Meaningless.
PlusOverData: DispMaxData+1
MinusOverData: DispMinData-1
NonData: PlusOverData
DispMaxData: Maximum display value
DsipMinData: Minimum display value

• WE7141
Description: Information regarding the physical value (depends on the measurement function setting)
dataType: WE_DOUBLE
blockNum: 1
startBit: Meaningless
effectiveBit: Meaningless
trigActive: Meaningless
record: Record length setting value
recordLen: Meaningless
trigPosition: Meaningless
interval: Sampling interval setting
vResolution: 1.0
vOffset: 0.0
trigLevel: Meaningless
trigWidth: Meaningless
plusOverData: +∞
minusOverData: -∞
nonData: Non number (NAN)
dispMaxData: Depends on the measurement function setting
dispMinData: Depends on the measurement function setting

• WE7231
Description: Information regarding the physical value (temperature, voltage, or resistance)
DataType: WE_FLOAT
BlockNum: Number of blocks measured after starting data acquisition
StartBit: Meaningless
EffectiveBit: Meaningless
TrigActive: Meaningless
Record: Record length setting
RecordLen: Meaningless

TrigPosition: Meaningless
Interval: Depends on the Sampling Interval setting
Vresolution: 1.0
Voffset: 0.0
TrigLevel: Meaningless
TrigWidth: Meaningless
PlusOverData: DispMaxData +100
MinusOverData: DispMinData -100
NonData: Non number (NAN)
DispMaxData: Depends on the measurement range
DsipMinData: Depends on the measurement range

• WE7235
Description: Information regarding A/D raw data
DataType: WE_SWORD
blockNum: Number of memory partitions
StartBit: Meaningless
EffectiveBit: Meaningless
TrigActive: When the trigger type is not Off: 1
When set to other or BUSTRG: 0
Record: Record length setting
RecordLen: Record -1
TrigPosition: Pre-trigger setting
Interval: Sampling interval setting
VResolution: Scaling factor for converting physical values
VOffset: Offset for converting physical values
TrigLevel: Trigger level setting
TrigWidth: Meaningless
PlusOverData: DispMaxData +1
MinusOverData: DispMinData -1
NonData: PlusOverData
DispMaxData: Maximum display value
DsipMinData: Minimum display value

• WE7241
Description: Information regarding the physical value (temperature or voltage)
dataType: WE_FLOAT
blockNum: 1
startBit: Meaningless
effectiveBit: Meaningless
trigActive: Meaningless
record: Record length setting
recordLen: Meaningless
trigPosition: Meaningless
interval: Sampling interval setting
vResolution: 1.0
vOffset: 0.0
trigLevel: Meaningless
trigWidth: Meaningless
plusOverData: “dispMaxData” +100
minusOverData: “dispMinData” -100
nonData: Non number (NAN)
dispMaxData: Depends on the measurement range setting

dispMinData: Depends on the measurement range setting

• WE7245

Description: Information regarding the A/D raw data
DataType: WE_SWORD
blockNum: Number of memory partitions
StartBit: Meaningless
EffectiveBit: Meaningless
TrigActive: 1 if the trigger type is not Off, 0 otherwise. 0 also for BUSTRG.
Record: Record length setting
RecordLen: Record-1
TrigPosition: Pre-trigger setting
Interval: Sampling interval setting
VResolution: Depends on the measurement range, the calibration at the time of shipment, and the execution of balancing.
VOffset: Depends on the measurement range, the calibration at the time of shipment, and the execution of balancing.
TrigLevel: Trigger level setting
TrigWidth: Meaningless
PlusOverData: DispMaxData+1
MinusOverData: DispMinData-1
NonData: PlusOverData
DispMaxData: Depends on the measurement range and the calibration at the time of shipment.
DsipMinData: Depends on the measurement range and the calibration at the time of shipment.

• WE7251

Description: Information regarding raw data
dataType: WE_SWORD
blockNum: Number of memory partitions
startBit: Meaningless
effectiveBit: Meaningless
trigActive: 1 if the trigger type is not Off 0 otherwise. 0 also for BUSTRG.
record: Record length setting
recordLen: “record” - 1
trigPosition: Pre-trigger setting
interval: Sampling interval setting
vResolution: Measurement range, depends on the calibration value at the time of shipment
vOffset: Measurement range, depends on the calibration value at the time of shipment
trigLevel: Trigger level setting
trigWidth: Difference between the High and Low levels when the trigger type is In or OUT, meaningless for all other types or when the trigger source is set to BUSTRG.
plusOverData: “dispMaxData” + 1
minusOverData: “dispMinData” - 1
nonData: “plusOverData”
dispMaxData: Measurement range, depends on the calibration value at the time of shipment
dispMinData: Measurement range, depends on the calibration value at the time of shipment

• WE7261/WE7262
Description: Information regarding the In/Out bit data
dataType: BIT16_TYP
blockNum: 1
startBit: 0
effectiveBit: 15
trigActive: Meaningless
record: 8192
recordLen: Meaningless
trigPosition: Meaningless
interval: Sampling interval setting
vResolution: Meaningless
vOffset: Meaningless
trigLevel: Meaningless
trigWidth: Meaningless
plusOverData: Meaningless
minusOverData: Meaningless
nonData: Meaningless
dispMaxData: 2
dispMinData: -1

• WE7271/7272/7275
Description: Information regarding raw data
dataType: WE_SWORD
blockNum: Number of memory partitions
startBit: Meaningless
effectiveBit: Meaningless
trigActive: 0 when the trigger type is set to OFF, 1 otherwise. Meaningless when the trigger source is set to BUSTRG.
record: Record length setting
recordLen: “record” - 1
trigPosition: Pre-trigger setting
interval: Sampling interval setting
vResolution: Measurement range, depends on the calibration value at the time of shipment
vOffset: Measurement range, depends on the calibration value at the time of shipment
trigLevel: Trigger level setting
trigWidth: Meaningless
plusOverData: “dispMaxData” + 1
minusOverData: “dispMinData” - 1
nonData: “plusOverData”
dispMaxData: Measurement range, depends on the calibration value at the time of shipment
dispMinData: Measurement range, depends on the calibration value at the time of shipment

• WE7273
Description: Information regarding A/D raw data
dataType: WE_SWORD
blockNum: Number of memory partitions
startBit: Meaningless
effectiveBit: Meaningless
trigActive: 0 when the trigger type is set to OFF, 1 otherwise. Meaningless when the trigger source is set to BUSTRG.
record: Record length setting
recordLen: “record” - 1

trigPosition: Pre-trigger setting
interval: Sampling interval setting
vResolution: Scaling factor for converting physical values
vOffset: Offset for converting physical values
TrigLevel: Trigger level setting
TrigWidth: Meaningless
PlusOverData: “DispMaxData” + 1
MinusOverData: “DispMinData” - 1
NonData: “PlusOverData”
DispMaxData: Maximum display value
DispMinData: Minimum display value

• WE7311
Description: Information regarding raw data
DataType: WE_SBYTE
BlockNum: Specified number of blocks
StartBit: Meaningless
EffectiveBit: Meaningless
TrigActive: 1 for data from the channel which is the trigger source. Meaningless for all other channels or when the trigger source is set to BUSTRG/External 1M/50.
Record: Record length setting
RecordLen: Depends on the record length and Time/div settings
TrigPosition: Depends on the pre-trigger, record length, and trigger position settings
Interval: Depends on the sampling interval and Time/div settings
Vresolution: Depends on the Range, V/div, and probe attenuation settings
Voffset: Depends on the offset voltage and probe attenuation settings
TrigLevel: Depends on the trigger level and probe attenuation settings
TrigWidth: Meaningless
PlusOverData: 0x7F
MinusOverData: 0x81
NonData: 0x80
DispMaxData: 0x7D
DsipMinData: 0x83

• WE7521
For Counter Mode
Description: Information regarding raw data
DataType: When the measurement function is UpDown1/UpDown2/UpDown4:
WE_SLONG
When the measurement function is Frequency:
Prior to transformation by the WeTransAcqData API: WE_ULONG
After transformation by the WeTransAcqData API: WE_FLOAT
When the measurement function is other than above: WE_ULONG
blockNum: Number of memory partitions
StartBit: Meaningless
EffectiveBit:
When the measurement function is Period/TI: value depending on the period stop determination time
When the measurement function is Frequency: value depending on the 128+period stop determination time
When the measurement function is other than above: 0
WE Control API Draft Version 1/3
TrigActive: 1 if the trigger type is not OFF, 0 otherwise. 0 also for BUSTRG.

Record: Record length setting
RecordLen: Record –1
TrigPosition: Pre-trigger setting
Interval: Sampling interval setting
VResolution: Scaling factor for converting physical values
VOffset: Offset for converting physical values
TrigLevel: Trigger level setting
TrigWidth: Meaningless
PlusOverData: DispMaxData + 1
MinusOverData: DispMinData – 1
NonData: When the measurement function is UpDown1/UpDown2/UpDown4: 0x80000000
When the measurement function is other than above: 0xffffffff
DispMaxData: Maximum display value
DsipMinData: Minimum display value

For time stamp mode
Description: Information regarding raw data
DataType: WE_ULONG
Upper 24 bits: Time stamp data Lower 8 bits: Input change information
blockNum: Meaningless
StartBit: 8
EffectiveBit: 24
TrigActive: Meaningless
Record: Record length retrieved between latches
RecordLen: Meaningless
TrigPosition: Meaningless
Interval: Meaningless
VResolution: Scaling factor for converting physical values
VOffset: Offset for converting physical values
TrigLevel: Meaningless
TrigWidth: Meaningless
PlusOverData: Meaningless
MinusOverData: Meaningless
NonData: Meaningless
DispMaxData: Meaningless
DsipMinData: Meaningless

• WE7562
Description: Information regarding A/D raw data
DataType: For PHA mode:
16bit/CH: WE_UWORD 32bit/CH: WE_ULONG
For MCS mode:
WE_ULONG
For LIST mode:
WE_ULLONG (Upper 16 bits: Pulse height value data, Lower 48 bits: Time stamp data)
blockNum: For PHA mode:
Number of readable pages including the page currently being measured
Always 1 for MCS or LIST mode.
StartBit: Meaningless.
EffectiveBit: Meaningless.
TrigActive: Set to 1 for self trigger.
Set to 0 for all other cases (such as BUSTRG).
Record: For PHA mode:

Gain (total number of channels)
For MCS mode:
Number of measurement channels selected by the previous latch operation or the total number of measured channels
For LIST mode:
Number of data values selected by the previous latch operation or the total number of measured data values

RecordLen: Record – 1
TrigPosition: 0 (meaningless)
Interval: For PHA or LIST mode: 1
For MCS mode: Dwell time
VResolution: 1.0
VOffset: 0.0
TrigLevel: 0 (meaningless)
TrigWidth: 0 (meaningless)
PlusOverData: 4294967295 (meaningless)
MinusOverData: 0 (meaningless)
NonData: 4294967295 (meaningless)
DispMaxData: For PHA mode:
16bit/CH: 65535
32bit/CH: 4294967295
For MCS mode: 4294967295
For LIST mode: 0 (meaningless)
DispMinData: For PHA or MCS mode: 0
For LIST mode: 0 (meaningless)

Example (Visual Basic)

```
' Query the additional information pertaining to the acquisition
' data of all channels of the WE7111.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
Dim info (2) As AcqDataInfo
Dim num As Integer
ret = WeGetAcqDataInfo (dsoHandle, -1, 0, info(0), num)
```

WeGetAcqDataSize

Description

Determines the acquisition data size (number of bytes) of the measurement module. The number of data bytes are adjusted according to the data type.

Syntax

WeGetAcqDataSize (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByRef *pointNum* As Long, ByRef *dataSize* As Long, ByRef *ptype* As Integer, ByRef *chNum* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the number exceeds the valid number of blocks, an error occurs.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
blockNo (IN)	Block number. For details, see section 4.2.
pointNum (OUT)	Number of data points. If -1 is specified for ch, the number of data points, of all channels that can currently acquire data, is returned.
dataSize (OUT)	Total number of bytes. If -1 is specified for ch, the data size, of all channels that can currently acquire data, is returned. This number is undefined for free run acquisition mode.
ptype (OUT)	Data type
chNum (OUT)	Total number of channels that can currently acuire data

Note:

For the free run acquisition mode, call the latch function WeLatchData as necessary before calling this function. The data size between latches can be obtained. For details, see chapter 4.

Example (Visual Basic)

```
' Get the data size of one channel of the WE7111.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
Dim pointNum As Long
Dim dataSize As Long
DIm ptype As Integer
Dim chNum As Integer
ret = WeGetAcqDataSize (dsoHandle, 1, 0, pointNum, dataSize, ptype,
chNum)
```

WeGetAcqData

Description

Retrieves acquisition data from the measurement module. The retrieved data are raw data that have been A/D converted. The data format depends on the measurement module. The relevant information can be obtained with the WeGetAcqDataInfo function.

Syntax

WeGetAcqData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByRef *recSize* As Long, ByRef *buf* As Any, ByRef *ptype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the number exceeds the valid number of blocks, an error occurs.

Parameters

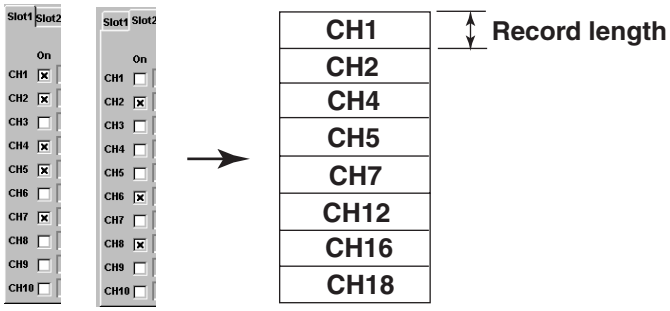
hMo (IN)	Module handle
ch (IN)	Channel number. One origin is. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
blockNo (IN)	Block number. For details, see section 4.2.
recSize (IN/OUT)	Data buffer size for retrieving data or the data size that was retrieved (bytes)
buf (OUT)	Pointer to the data buffer. Allocate enough buffer space to store the acquisition data being retrieved. You can use the WeGetAcqDataSize () to query the data size.
ptype (OUT)	Data type

```
WE_NULL      ' No parameter
WE_UBYTE     ' 8-bit unsigned integer
WE_SBYTE     ' 8-bit signed integer
WE_BIT8      ' 8-bit logical value
WE_UWORD     ' 16-bit unsigned integer
WE_SWORD     ' 16-bit signed integer
WE_BIT16     ' 16-bit logical value
WE_ULONG     ' 32-bit unsigned integer
WE_SLONG     ' 32-bit signed integer
WE_BIT32     ' 32-bit logical value
WE_FLOAT     ' 32-bit real number
WE_DOUBLE    ' 64-bit real number
WE_BIT64     ' 64-bit logical value
```

Note:

For the free run acquisition mode, call the latch function WeLatchData () before calling this function as necessary. The data size between latches can be obtained. For details, see chapter 4.

The following figure depicts the data format when "-1", that represents all channels, is specified for the channel number (ch (IN)). If two modules are linked and the channels are enabled as shown in the lower left figure, the disabled channels are skipped and data are continuously read as shown in the lower right figure.



Example (Visual Basic)

• Retrieving the raw data of channel 1 of WE7111

```
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
Dim recSize As Long
Dim ptype As Integer
' Allocate a buffer for a memory length of 10000 points.
Dim buf (10000) As Byte
' The number of bytes of the buffer is 10000 bytes
recSize = 10000
ret = WeGetAcqData (dsoHandle, 1, 0, recSize, buf(0), ptype)
```

• Retrieving the raw data for all channels of the WE7251

```
Dim stHandle As Long
Dim adcHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7251 module with a link number of 2.
ret = WeOpenModule (adcHandle, "WE7251: 1", 2, adcHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
Dim recSize As Long
Dim ptype As Integer
' Allocate a buffer for a memory length of 1000*20Ch
Dim buf (1000*20) As Integer
' The number of bytes of the buffer is 1000 x 20 x 2 bytes (WE_SWORD)
recSize = 1000*20*2
ret = WeGetAcqData (adcHandle, -1, 0, recSize, buf(0), ptype)
```

WeGetAcqDataEx

Description

Retrieves the acquisition data from the measurement module. Unlike WeGetAcqData, this API can read the interpolated data (Peak-to-peak (MIN-MAX) data).

Syntax

WeGetAcqDataEx (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Integer, ByVal *startPoint* As Long, ByVal *endPoint* As Long, ByVal *ppNum* As Integer, ByVal *Interpolation* As Integer, ByRef *recSize* As Long, ByRef *buf* As Any, ByRef *ptype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the number exceeds the valid number of blocks, an error occurs.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
blockNo (IN)	Block number. For details, see section 4.2.
startPoint (IN)	Start point for the retrieval of the data. The counting origin is zero. Specifying -1 sets the start point to the top of the data.
endPoint (IN)	Last point at which to retrieve the data. The counting origin is zero. Specifying -1 sets the end point to the end of the data.
ppNum (IN)	Number of display data sets (Pair of MIN and MAX values) If -1: Data compression/interpolation is not performed. (endPoint-startPoint+1) points of data are returned. If other than -1: If ppNum > (endPoint – startPoint + 1), data interpolation takes place and ppNum points of data are created. If ppNum < (endPoint – startPoint + 1), data compression takes place and ppNum points of data are created.
Interpolation (IN)	Data interpolation type selection WE_INTER_PULSE ' Pulse interpolation WE_INTER_SIN ' Sin(x)/x interpolation WE_INTER_LINE ' Line interpolation
recSize (IN/OUT)	Data buffer size for retrieving data or the data size that was retrieved
buf (OUT)	Pointer to the data buffer
type (OUT)	Data type WE_NULL ' No parameter WE_UBYTE ' 8-bit unsigned integer WE_SBYTE ' 8-bit signed integer WE_BIT8 ' 8-bit logical value WE_UWORD ' 16-bit unsigned integer WE_SWORD ' 16-bit signed integer WE_BIT16 ' 16-bit logical value WE_ULONG ' 32-bit unsigned integer WE_SLONG ' 32-bit signed integer WE_BIT32 ' 32-bit logical value WE_FLOAT ' 32-bit real number WE_DOUBLE ' 64-bit real number WE_BIT64 ' 64-bit logical value

Note:

For the free run acquisition mode, call the latch function WeLatchData before calling this function as necessary. The data size between latches can be obtained. For details, see chapter 4.

Example (Visual Basic)

```
' Retrieve the acquisition data of channel 1 of the WE7111.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
Dim recSize As Long
Dim buf(1000) As Byte
Dim ptype As Integer
recSize = 1000
ret = WeGetAcqDataEx (dsoHandle, 1, 0, 1, 1000, -1, WE_INTER_SIN,
recSize, buf(0), ptype)
```

WeGetCurrentData

Description

Retrieves instantaneous data from the measurement module. This function retrieves the newest data.

Syntax

WeGetCurrentData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByRef *recSize* As Long, ByRef *buf* As Any, ByRef *ptype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
recSize (IN/OUT)	Data buffer size for retrieving data or the data size that was retrieved
buf (OUT)	Pointer to the data buffer
ptype (OUT)	Data type

```
WE_NULL      ' No parameter
WE_UBYTE     ' 8-bit unsigned integer
WE_SBYTE     ' 8-bit signed integer
WE_UWORD     ' 16-bit unsigned integer
WE_SWORD     ' 16-bit signed integer
WE_ULONG     ' 32-bit unsigned integer
WE_SLONG     ' 32-bit signed integer
WE_FLOAT     ' 32-bit real number
WE_DOUBLE    ' 64-bit real number
```

Example (Visual Basic)

• Retrieve instantaneous data from all channels of the WE7251.

```
Dim stHandle As Long
Dim adcHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7251 module with a link number of 1
ret = WeOpenModule (stHandle, "WE7251: 1", 1, adcHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
' Allocate Double 10 CH of data buffer.
Dim buf (10) As Double
Dim recSize As Long
Dim ptype As Integer
' 10CH x 8 bytes
recSize = 10*8
ret = WeGetCurrentData (adcHandle, -1, recSize, buf(0), ptype)
```

• Retrieve instantaneous data from all channels of the WE7241.

```
Dim stHandle As Long
Dim tcHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7241 module with a link number of 1
ret = WeOpenModule (stHandle, "WE7241: 1", 1, tcHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
' Allocate Single 10 CH of data buffer.
Dim buf (10) As Single
Dim recSize As Long
Dim ptype As Integer
' 10CH x 4 bytes
recSize = 10*4
ret = WeGetCurrentData (adcHandle, -1, recSize, buf(0), ptype)
```

Note:

Retrieval is carried out even if there is no data for one or more of the specified channels.

WeGetScaleCurrentData

Description

Reads the data obtained after scaling the instantaneous values of the measurement module.

Syntax

WeGetScaleCurrentData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *ptype* As Integer, ByRef *param* As ScalingParam, ByRef *recSize* As Long, ByRef *buf* As Any) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, it is the link number. -1 represents all channels.
ptype (IN)	Type of data being read WE_FLOAT ' 32-bit real number WE_DOUBLE ' 64-bit real number
param (IN/OUT)	Scale parameter information If -1 is specified for ch, the number of parameters that needs to be specified is equal to the number of channels. Set the params in the order of channel data to be retrieved. The scale values are used to calculate ax + b where x is the physical value of the data. Type ScalingParam a As Double ' a of the scaling parameter ax + b b As Double ' b of the scaling parameter ax + b ch As Long ' Channel number retrieved End Type
recSize (IN/OUT)	Size of the received data buffer in bytes (IN)/Size of the received data in bytes (OUT)
buf (OUT)	Pointer to the data buffer to be read.

Example (Visual Basic)

```
Read the data obtained after scaling the instantaneous value of the
WE7271 module.
Dim hSt As Long, hMo As Long
' Get the station handle of the station named "Station 1."
ret =WeOpenStation ("Station1", hSt)
' Open the WE7271 module with a link number of 1.
ret =WeOpenModule (hSt,"WE7271:1",1,hMo)
' Waiting for completion of data acquisition after starting
measurement
  ......
' Allocate a buffer for a memory length of Double 4ch
Dim buf (4) As Double
Dim recSize As Long
recSize = 4*8        ' 4ch x 8 bytes
Dim param(4) As ScalingParam
param(0).a =1.0
param(0).b =0.0
param(1).a =2.0
param(1).b =0.0
param(2).a =3.0
param(2).b =0.0
param(3).a =4.0
param(3).b =0.0
ret =WeGetScaleCurrentData (hMo, -1, WE_DOUBLE, param(0), recSize,
buf(0))
```

Note:

Retrieval is carried out even if there is no data for one or more of the specified channels.

WeGetScaleCurrentDataEx

Description

Reads the data obtained after scaling the instantaneous values of the measurement module. Uses the scale conversion values that were specified using the WeSetScaleInfo API or the WeShowLinearScaleWindow API, that are stored in the module.

Syntax

WeGetScaleCurrentDataEx (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *ptype* As Integer, ByRef *recSize* As Long, ByRef *buf* AS Any) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number One origin. If the modules are linked, it is the link number. -1 represents all channels.
ptype (IN)	Type of data being read WE_FLOAT ' 32-bit real number WE_DOUBLE ' 64-bit real number
recSize (IN/OUT)	Size of the received data buffer in bytes (IN)/Size of the received data in bytes (OUT)
buf (OUT)	Pointer to the data buffer to be read.

Example (Visual Basic)

```
Read the data obtained after scaling the instantaneous value of the
WE7271 module.
Dim hSt As Long, hMo As Long
' Get the station handle of the station named "Station 1."
ret =WeOpenStation ("Station1", hSt)
' Open the WE7271 module with a link number of 1.
ret =WeOpenModule (hSt,"WE7271:1",1,hMo)
' Waiting for completion of data acquisition after starting
measurement
  ......
' Allocate a buffer for a memory length of Double 4ch
Dim buf (4) As Double
Dim recSize As Long
recSize = 4*8        ' 4ch x 8 bytes
ret =WeGetScaleCurrentDataEx (hMo, -1, WE_DOUBLE, recSize, buf(0))
```

Note:

Channels not specified for measurement are also included in the data, so you must specify enough buffers to accommodate those channles as well.

WeGetScaleData

Description

Retrieves the acquisition data of the measurement module that have been scaled. Converts the acquisition data that have been A/D converted to physical values and then scales them according to the user-specified scale parameter.

Syntax

WeGetScaleData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByRef *param* As ScalingParam, ByRef *recSize* As Long, ByVal *ptype* As Integer, ByRef *buf* As Any) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the number exceeds the valid number of blocks, an error occurs.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
blockNo (IN)	Block number. For details, see section 4.2.
param (IN/OUT)	Scaling parameter information. If -1 is specified for ch, specify the channel worth of params. Set the params in the order of channel data to be retrieved. The scale values are used to calculate ax + b where x is the physical value of the data.

```
Type ScalingParam
  a  As Double          ' a of the scaling parameter ax + b
  b  As Double          ' b of the scaling parameter ax + b
  ch As Long            ' Channel number retrieved
End Type
```

recSize (IN/OUT)	Data buffer size for retrieving data or the data size that was retrieved (bytes)
ptype (IN)	Data type

```
WE_FLOAT              ' 32-bit real number
WE_DOUBLE             ' 64-bit real number
```

buf (OUT)	Pointer to the data buffer. Allocate enough buffer space to store the acquisition data being retrieved. Make sure to take ptype into account.

Note:

For the free run acquisition mode, call the latch function WeLatchData before calling this function as necessary. The data size between latches can be obtained. For details, see chapter 4.

Example (Visual Basic)

```
' Retrieve the physical value data of all channels of the WE7111.
' Retrieve the physical value data without scale conversion
' in 32-bit real number (4 bytes) format.
' (Specify 1 and 0 for a and b, respectively, in the scaling
' equation.)
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
Dim recSize As Long
' Allocate a buffer for a memory length of 1000 x 2ch
Dim buf (1000*2) As Single
' The number of bytes of buffer is 1000 x 2ch x 4 bytes (WE_FLOAT)
recSize = 1000*2*4
Dim param (2) As ScalingParam
param (0).a = 1
param (0).b = 0
param (1).a = 1
param (1).b = 0
ret = WeGetScaleData (dsoHandle, -1, 0, param (0), recSize,
WE_FLOAT, buf (0))
```

WeGetScaleDataEx

Description

Reads the data obtained after scaling the acquisition data of the measurement module. Uses the scale conversion values that were specified using the WeSetScaleInfo API or the WeShowLinearScaleWindow API, that are stored in the module.

Syntax

WeGetScaleDataEx (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByRef *recSize* As Long, ByVal *ptype* As Integer, ByRef *buf* As Any) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number One origin. If the modules are linked, it is the link number. -1 represents all channels.
blockNo (IN)	Block number For details, see section 4.2, "Specifying the Data Block that You Wish to Retrieve."
recSize (IN/OUT)	Size of the received data buffer in bytes (IN)/Size of the received data in bytes (OUT)
ptype (IN)	Type of data being read WE_FLOAT ' 32-bit real number WE_DOUBLE ' 64-bit real number
buf (OUT)	Pointer to the data buffer to be read. Allocate enough buffer space to store the acquisition data being retrieved. Make sure to take ptype into account.

Example (Visual Basic)

```
Read the scale converted data of all the channels of the WE7111
module.
Dim hSt As Long, hMo As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret =WeOpenStation ("Station1", hSt)
' Open the WE111 module with a link number of 2.
ret =WeOpenModule (hSt,"WE7111:1",2, hMo)
' Waiting for completion of data acquisition after starting
measurement
  ......
Dim recSize As Long
' Allocate a buffer for a memory length of 1000 x 2ch
Dim buf (1000*2) As Single
' The number of bytes of buffer is 1000 x 2ch x 4 bytes (WE_FLOAT)
recSize = 1000*2*4
ret =WeGetScaleDataEx (hMo, -1, 0, recSize, WE_FLOAT, buf(0))
```

WeGetMeasureParam

Description

Gets the automated measurement values from the measurement module. The automated measurement values are the results analyzed by the modules. For modules that do not have the automated measurement function, the WeExecMeasureParam function can be used.

Syntax

WeGetMeasureParam (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByVal *startPoint* As Long, ByVal *endPoint* As Long, ByRef *Item* As MeasureItem, ByRef *itemNum* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the number exceeds the valid number of blocks, an error occurs.

Parameters

hMo (IN) Module handle

ch (IN) Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.

blockNo (IN) Block number. For details, see section 4.2.

startPoint (IN) Start point of the data for the automated measurement. One origin. Specifying -1 sets the start point to the top of the data.

endPoint (IN) End point of the data for the automated measurement. One origin. Specifying -1 sets the end point to the end of the data.

Item (OUT) Measurement information pointer

```
Type MeasureItem        ' Measurement result storage structure
  Channel As Double     ' CH number (1 or greater)
  Max As Double         ' Maximum value
  Min As Double         ' Minimum value
  High As Double        ' High level
  Low As Double         ' Low level
  PP As Double          ' P-P value
  Ampl As Double        ' Amplitude
  Avg As Double         ' Average value
  Rms As Double         ' RMS value
  Middle As Double      ' Center value of the amplitude
  StdDev As Double      ' Standard deviation
  Oshoot As Double      ' Overshoot
  Ushoot As Double      ' Undershoot
  Rise As Double        ' Rise time
  Fall As Double        ' Fall time
  Freq As Double        ' Frequency
  Period As Double      ' Period
  Duty1 As Double       ' Duty cycle on the High side
  Duty2 As Double       ' Duty cycle on the Low side
  Width1 As Double      ' Width above the mesial value
  Width2 As Double      ' Width below the mesial value
End Type
```

itemNum (OUT) Number of MeasureItem that are actually retrieved

Example (Visual Basic)

```
' Get the measurement data of channel 1 of the WE7111.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
Dim buf  As MeasureItem
Dim num As Integer
ret = WeGetMeasureParam (dsoHandle, 1, 0, 1, 10000, buf, num)
```

WeSaveAcqData

Description

Saves the acquisition data of the measurement module to a file.

Syntax

WeSaveAcqData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByVal *filename* As String, ByVal *htype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin is. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
blockNo (IN)	Block number. If the number exceeds the valid number of blocks, all block data are saved. For details, see section 4.2.
filename (IN)	File name. The file extension is not necessary. A file with the extension, WVF, is created.
htype (IN)	Whether or not to create the waveform information file (HDR file extension) for the acquisition data (0: do not create, 1: create).

Example (Visual Basic)

```
' Save the raw data of channel 1 of the WE7111 to a file.
' Create a header file.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
ret = WeSaveAcqData (dsoHandle, 1, 0, "c:\dsoparam", 1)
```

WeSaveScaleData

Description

Saves the scaled acquisition data of the measurement module to a file.

Syntax

WeSaveScaleData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByRef *param* As ScalingParam, ByVal *filename* As String, ByVal *htype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
blockNo (IN)	Block number. For details, see section 4.2.
param (IN/OUT)	Scaling parameter information Type ScalingParam a As Double ' a of the scaling parameter ax + b b As Double ' b of the scaling parameter ax + b ch As Long ' Channel number retrieved End Type
filename (IN)	File name. The file extension is not necessary. A file with the extension, WVF, is created.
htype (IN)	Whether or not to create the waveform information file (HDR file extension) for the acquisition data (0: do not create, 1:create).

Example (Visual Basic)

```
' Save the physical value data of channel 1 of the WE7111.
' Create a header file.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
Dim param (2) As ScalingParam
param (0).a = 1.0
param (0).b = 0.0
param (1).a = 1.0
param (1).b = 0.0
ret = WeSaveScaleData (dsoHandle, 1, 0, param (0), "c:\dsoparam", 1)
```

WeSaveScaleDataEx

Description

Saves the scaled acquisition data of the measurement module to a file in binary format. Uses the scale conversion values that were specified using the WeSetScaleInfo API or the WeShowLinearScaleWindow API, that are stored in the module.

Syntax

WeSaveScaleDataEx (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByVal *filename* As String, ByVal *htype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number One origin. If the modules are linked, it is the link number. -1 represents all channels.
blockNo (IN)	Block number For details, see section 4.2, "Specifying the Data Block that You Wish to Retrieve."
filename (IN)	File name. The file extension is not necessary. A file with the extension, WVF, is created.
htype (IN)	Whether or not to create the waveform information file (HDR file extension) for the acquisition data (0: do not create, 1: create).

Example (Visual Basic)

```
' Save the raw data of all the channels of the WE7111 to a file.
' Create a header file.
Dim hSt As Long, hMo As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret =WeOpenStation ("Station1", hSt)
' Open the WE111 module with a link number of 2.
ret =WeOpenModule (hSt,"WE7111:1", 2, hMo)
' Waiting for completion of data acquisition after starting
measurement
  ......
ret =WeSaveScaleDataEx (hMo, -1, 0, "c:\ WE7111Data", 1)
```

WeSaveAsciiData

Description

Saves the physical value data of the measurement module to a file in ASCII format (CSV format). the physical value data is the measurement unit data of the module. This is not the scaled acquisition data. The measurement unit of the digitizer module is “voltage,” for example.

Syntax

WeSaveAsciiData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByVal *filename* As String, ByVal *htype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
blockNo (IN)	Block number
filename (IN)	File name. The file extension is not necessary. A file with the extension, CSV, is created.
htype (IN)	Whether or not to create the waveform information file (HDR file extension) for the acquisition data (0: do not create, 1:create).

Example (Visual Basic)

```
' Save the physical value data of channel 1 of the WE7111 in
' ASCII format.
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
ret = WeSaveAsciiData (dsoHandle, 1, 0, "c:\dsodata", 1)
```

WeSaveScaleAsciiData

Description

Saves the scaled acquisition data of the measurement module to a file in ASCII format (CSV format).

Syntax

WeSaveScaleAsciiData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByRef *param* As ScalingParam, ByVal *filename* As String, ByVal *htype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number One origin. If the modules are linked, it is the link number. -1 represents all channels.
blockNo (IN)	Block number
param (IN/OUT)	Scale value information If -1 is specified for ch, the number of parameters that needs to be specified is equal to the number of channels. Set the params in the order of channel data to be retrieved. The scale values are used to calculate ax + b where x is the physical value of the data. Type ScalingParam a As Double ' a of the scaling parameter ax + b b As Double ' b of the scaling parameter ax + b ch As Long ' Channel number retrieved End Type
filename (IN)	File name. The file extension is not necessary. A file with the extension, CSV, is created.
htype (IN)	Whether or not to create the waveform information file (HDR file extension) for the acquisition data (0: do not create, 1: create).

Example (Visual Basic)

```
' Save the scaled physical value data of all the channels of the
WE7111 in ASCII format.
Dim hSt As Long, hMo As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret =WeOpenStation ("Station1", hSt)
' Open the WE111 module with a link number of 2.
ret =WeOpenModule (hSt, "WE7111:1", 2, hMo)
Dim param(2)As ScalingParam
param(0).a =1.0
param(0).b =0.0
param(1).a =2.0
param(1).b =0.0
' Waiting for completion of data acquisition after starting
measurement
  ......
ret =WeSaveScaleAsciiData (hMo, -1, 0, param(0), "c:\WE7111Data",
1)
```

WeSaveScaleAsciiDataEx

Description

Saves the scaled acquisition data of the measurement module to a file in ASCII format (CSV format). Uses the scale conversion values that were specified using the WeSetScaleInfo API or the WeShowLinearScaleWindow API, that are stored in the module.

Syntax

WeSaveScaleAsciiDataEx (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Long, ByVal *filename* As String, ByVal *htype* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number One origin. If the modules are linked, it is the link number. -1 represents all channels.
blockNo (IN)	Block number
filename (IN)	File name. The file extension is not necessary. A file with the extension, CSV, is created.
htype (IN)	Whether or not to create the waveform information file (HDR file extension) for the acquisition data (0: do not create, 1: create).

Example (Visual Basic)

```
' Save the scaled physical value data of all the channels of the
WE7111 in ASCII format.
Dim hSt As Long, hMo As Long
Dim ret As Integer
ret =WeOpenStation ("Station1", hSt)
' Get the station handle of the station named "Station 1."
ret =WeOpenModule (hSt,"WE7111:1", 2, hMo)
' Open the WE111 module with a link number of 2.
' Waiting for completion of data acquisition after starting
measurement
  ......
ret =WeSaveScaleAsciiDataEx (hMo, -1, 0, "c:\WE7111Data", 1)
```

WeSaveAcqHeader

Description

Creates the waveform information file for the acquisition data of the measurement module. The file that is created is the same file created with the WeSaveAcqData function.

Syntax

WeSaveAcqHeader (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *filename* As String) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful. If the number exceeds the valid number of blocks, an error occurs.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number to save. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
filename (IN)	File name. The file extension is not necessary. A file with the extension, HDR, is created.

Example (Visual Basic)

```
' Save the waveform information file for all channels of the WE7111
Dim stHandle As Long
Dim dsoHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open module WE7111 with a link number of 2.
ret = WeOpenModule (stHandle, "WE7111: 1", 2, dsoHandle)
' Waiting for completion of data acquisition after starting measurement
  ......
ret = WeSaveAcqHeader (dsoHandle, -1, "c:\station")
```

WeSavePatternData

Description

Saves the pattern data that are dependent on the measurement module to a file. The pattern data are different for each module. Some modules do not have pattern data defined. Pattern data are, for example, the arbitrary waveform data for the WE7121 and the digital pattern data for the WE7131.

Syntax

WeSavePatternData (IByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *filename* As String) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
filename (IN)	File name. The file extension is not necessary. The file extension defined for each measurement module is automatically added.

Example (Visual Basic)

```
' Save the pattern data of channel 1 of the WE7131.
Dim stHandle As Long
Dim pioHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7131 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7131: 1", 1, pioHandle)
ret = WeSavePatternData (pioHandle, 1, "c:\piopattern")
```

WeLoadPatternData

Description

Loads the pattern data that are dependent on the measurement module . The pattern data are different for each module. Some modules do not have pattern data defined. Pattern data are, for example, the arbitrary waveform data for the WE7121 and the digital pattern data for the WE7131.

Syntax

WeLoadPatternData (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *filename* As String) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
ch (IN)	Channel number. One origin. If the modules are linked, the channel number spans across the linked modules. -1 represents all channels.
filename (IN)	File name

Example (Visual Basic)

```
' Load the arbitrary waveform data to channel 1 of the WE7121.
Dim stHandle As Long
Dim fgHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7121 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7121: 1", 1, fgHandle)
ret = WeLoadPatternData (fgHandle, 1, "c:\fgarb")
```

WeLoadPatternDataEx

Description

Loads the waveform data that are retrieved using the specified parameter from the specified file (wvf or csv format) to the measurement module.

Syntax

WeLoadPatternDataEx (ByVal *hMo* As Long, ByVal *command* As String, ByVal *filename* As String, ByVal *ch* As Integer, ByVal *blockNo* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
command (IN)	ASCII command
filename (IN)	File name
ch (IN)	Channel number. One origin.
blockNo (IN)	Block number. Zero origin.

Note:

By setting the channel number to 0x7FFF or the block number to 0x7FFFFFFF, the waveform data of all channels or all blocks are transferred.
For the WE7281 module, auto load (AG:Misc:Load:Auto command) must be turned On.

Example (Visual Basic)

• Example in which the WeLoadPatternEx API is used

```
Dim ret As Integer
' Transfer the Block0 data of CH1 of the wvf file saved by the
' WE7271 module to CH1 of the WE7281 module.
ret = WeLoadPatternDataEx (hMo, "FG:CH1:Load ARB", "c:\we7271.wvf", 1, 0)
```

• Example in which the Filter API is used

```
Dim size As Long
Dim ret As Integer
' Determine the byte size when the Block0 data of CH1 of the wvf
' file saved by the WE7271 module are converted to the format used
' by the FG mode on the WE7281 module.
ret = WeWvf2S16GetSize("c:\we7271.wvf", 1, 0, size)
' Allocate the buffer using the retrieved data size.
size = size/2
ReDim s16buf(size) As Integer
' Convert the Block0 data of CH1 of the wvf file saved by the
' WE7271 module to the format used by the FG mode on the WE7281
' module.
ret = WeWvf2S16("c:\we7271.wvf", 1, 0, s16buf(0), size)
' Transfer the converted data to CH1 of the WE7281 module.
ret = WeSetControl (hMo, "FG:CH1:Load ARB", s16buf)
```

WeSetOverRun

Description

Select whether or not to detect overruns.

Syntax

WeSetOverRun (ByVal *hMo* As Long, ByVal *sw* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
sw (IN)	Overrun detection setting
	0 ' Do not detect overruns (do not stop when an overrun occurs)
	1 ' Detect overruns (stop when an overrun occurs)

Note:

Overruns are detected as default.

Example (Visual Basic)

```
' Disable the overrun detection on the WE7241 module.
Dim hSt As Long
Dim hMo As Long
Dim ret As Integer
' Get the station handle.
ret = WeOpenStation ("Station 1", hSt)
' Open module WE7241.
ret = WeOpenModule (hSt, "WE7241: 1", 1, hMo)
' Disable the overrun detection on the WE7241 module.
ret = WeSetOverRun (hMo, 0)
```

WeGetOverRun

Description

Queries whether or not overruns are to be detected.

Syntax

WeGetOverRun (ByVal *hMo* As Long, ByVal *sw* As Byte) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle
sw (IN)	Overrun detection setting
	0 ' Do not detect overruns (do not stop when an overrun occurs)
	1 ' Detect overruns (stop when an overrun occurs)

Example (Visual Basic)

```
' Queries the overrun detection setting on the WE7241 module.
Dim hSt As Long
Dim hMo As Long
Dim ret As Integer
Dim sw As Integer
' Get the station handle.
ret = WeOpenStation ("Station 1", hSt)
' Open module WE7241.
ret = WeOpenModule (hSt, "WE7241: 1", 1, hMo)
' Queries the overrun detection setting.
ret = WeGetOverRun (hMo, sw)
```

6.10 Events

WeCreateEvent

Description

Creates a handle for handling measurement module events and enables the operation. Up to 32 handles can be created for each measurement module. The cause of events are module common events and module dependent events. For the module dependent events, see the command table for each module.

Syntax

WeCreateEvent (ByVal *hMo* As Long, ByVal *pattern* As Long, ByVal *mode* As Long, ByRef *evHandle* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN) Station handle, module handle, module link handle

pattern (IN) Causes of events. The following common events are defined. Multiple events can be specified. There are other module dependent events that are defined beside the one listed below. For details, see the command table for each module.

```
WE_EV_MEASSTART   ' Generated when the start operation completes
WE_EV_MEASEND     ' Generated when the specified number of
                  ' blocks of data are acquired.
WE_EV_MEASABORT   ' Generated when the start state (RUN state)
                  ' is cleared.
WE_EV_BLOCKEND    ' Generated each time data are acquired to the
                  ' block when acquiring data using memory
                  ' partitions (blocks).
WE_EV_DIO0/WE_EV_DIO1/   ' Issued when a DIO pin of the EXT I/O
WE_EV_DIO2/WE_EV_DIO3    ' terminal changes.
                         ' When an event occurs, the higher 16
                         ' bits and the lower 16 bits of the second
                         ' parameter are set to the station number
                         ' and 0x00ff, respectively.
WE_EV_ERROR       ' Generated when an error defined by  the
                  ' measurement module occurs.
                  ' For details regarding the errors see the
                  ' command table for each module.
```

mode (IN) Event operation mode

```
WE_EV_STOP            ' Disable the event operation.
WE_EV_CONT            ' Detect and generate the event repetitively.
WE_EV_SINGLE          ' Detect and generate the event once.
WE_EV_SINGLE_RELEASE  ' Detect and generate the event once
                      ' and then release the event handle.
```

evHandle (OUT) Event handle retrieved

Note:

Handles do not need to be created for the power event (WE_EV_POWER) and the fan stop event (WE_EV_FANSTOP). These events are always detected.

Example (Visual Basic)

```
' Set (Enable) the block end event for the WE7251.
Dim stHandle As Long
Dim adcHandle As Long
Dim evHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7251 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7251", 1, adcHandle)
' Set the block end event.
ret = WeCreateEvent (adcHandle, WE_EV_BLOCKEND, WE_EV_CONT, evHandle)
```

WeSetEventPattern

Description

Set the cause of an event of the measurement module.

Syntax

WeSetEventPattern (ByVal *hMo* As Long, ByVal *evHandle* As Long, ByVal *pattern* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle or module link handle
evHandle (IN)	Event handle (0 to 31)
pattern (IN)	Cause of an event

Example (Visual Basic)

```
' Set (Enable) the block end event for the WE7251.
Dim stHandle As Long
Dim adcHandle As Long
Dim evHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7251 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7251", 1, adcHandle)
' Set the measurement end event.
ret = WeCreateEvent (adcHandle, WE_EV_MEASEND, WE_EV_CONT, evHandle)
ret = WeSetEventPattern (adcHandle, evHandle, WE_EV_BLOCKEND)
```

WeResetEventPattern

Description

Resets the cause of an event of the measurement module.

Syntax

WeResetEventPattern (ByVal *hMo* As Long, ByVal *evHandle* As Long, ByVal *pattern* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle or module link handle
evHandle (IN)	Event handle (0 to 31)
pattern (IN)	Cause of an event

Example (Visual Basic)

```
' Reset (Clear) the block end event for the WE7251.
Dim stHandle As Long
Dim adcHandle As Long
Dim evHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7251 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7251", 1, adcHandle)
ret = WeCreateEvent (adcHandle, WE_EV_BLOCKEND, WE_EV_CONT, evHandle)
ret = WeResetEventPattern (adcHandle, evHandle, WE_EV_BLOCKEND)
```

WeSetEventMode

Description

Set the operation mode of the measurement module events.

Syntax

WeSetEventMode (ByVal *hMo* As Long, ByVal *evHandle* As Long, ByVal *mode* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handle or module link handle	
evHandle (IN)	Event handle (0 to 31)	
mode (IN)	Event operation mode	
	WE_EV_STOP	' Disable the event operation.
	WE_EV_CONT	' Detect and generate the event ' repetitively.
	WE_EV_SINGLE	' Detect and generate the event once.
	WE_EV_SINGLE_RELEASE	' Detect and generate the event once ' and then release the event handle.

Example (Visual Basic)

```
' Disable the event operation mode for the WE7251.
Dim stHandle As Long
Dim adcHandle As Long
Dim evHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7251 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7251", 1, adcHandle)
ret = WeCreateEvent (adcHandle, WE_EV_BLOCKEND, WE_EV_CONT, evHandle)
ret = WeSetEventMode (adcHandle, evHandle, WE_EV_STOP)
```

WeReleaseEvent

Description

Releases the event handle of the measurement module. The handle cannot be used after it is released. However, the number can be used to create a new handle.

Syntax

WeReleaseEvent (ByVal *hMo* As Long, ByVal *evHandle* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

hMo (IN)	Module handleor module link handle
evHandle (IN)	Event handle (0 to 31)

Example (Visual Basic)

```
' Set the block end event for the WE7251 and then release the handle.
Dim stHandle As Long
Dim adcHandle As Long
Dim evHandle As Long
Dim ret As Integer
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station 1", stHandle)
' Open the WE7251 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7251", 1, adcHandle)
' Set the block end event.
ret = WeCreateEvent (adcHandle, WE_EV_BLOCKEND, WE_EV_CONT, evHandle)
ret = WeReleaseEvent (hMo, evHandle)
```

6.11 Waveform Parameter Computation

WeExecMeasureParam

Description

Performs automated computations on physical value data. WeGetMeasureParam function performs the automated computation of waveform parameters in the measurement module, but this function computes the parameters on the PC.

Syntax

WeExecMeasureParam (ByRef *data* As Any, ByVal *ptype* As Integer, ByVal *dt* As Double, ByVal *startPoint* As Long, ByVal *endPoint* As Long, ByRef *item* As MeasureItem) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

data (IN) Data on which to perform an automated computation. The data must be physical value and conform to the parameter type that is specified with ptype.

ptype (IN) Parameter type

```
WE_FLOAT        ' 32-bit real number
WE_DOUBLE       ' 64-bit real number
```

dt (IN) Sampling interval (seconds)

startPoint (IN) Starting point of the data to compute the parameter values. One origin.

endPoint (IN) Last point of the data to compute the parameter values. One origin. Calculates the parameter values using the data in the range from startPoint to endPoint.

item (OUT) Waveform parameter pointer

```
Type MeasureItem      ' Computation result storage structure
  Channel As Double   ' CH number (0)
  Max As Double       ' Maximum value
  Min As Double       ' Minimum value
  High As Double      ' High level
  Low As Double       ' Low level
  PP As Double        ' P-P value
  Ampl As Double      ' Amplitude
  Avg As Double       ' Average value
  Rms As Double       ' RMS value
  Middle As Double    ' Center value of the amplitude
  StdDev As Double    ' Standard deviation
  Oshoot As Double    ' Overshoot
  Ushoot As Double    ' Undershoot
  Rise As Double      ' Rise time
  Fall As Double      ' Fall time
  Freq As Double      ' Frequency
  Period As Double    ' Period
  Duty1 As Double     ' Duty cycle on the High side
  Duty2 As Double     ' Duty cycle on the Low side
  Width1 As Double    ' Width above the mesial value
  Width2 As Double    ' Width below the mesial value
End Type
```

Example (Visual Basic)

```
' Retrieves the measurement results of 1000 points of data
' at a sampling rate of 0.5 sec.
Dim ret As Integer
Dim item As Measureitem
Dim data (1000) As Single
ret = WeExecMeasureParam (data (0), WE_FLOAT, 0.5, 1, 1000, item)
```

WeExecMeasureParamAcqData

Description

Performs automated computations on the acquisition data (data after A/D conversion). WeGetMeasureParam function performs the automated computation of waveform parameters in the measurement module, but this function computes the parameters on the PC.

Syntax

WeExecMeasureParamAcqData (ByRef *data* As Any, ByVal *ptype* As Integer, ByVal *gain* As Double, ByVal *ofst* As Double, ByVal *dt* As Double, ByVal *st* As Long, ByVal *ed* As Long, ByRef *item* As MeasureItem) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

data (IN)	Pointer to the data on which to perform the automated computation
ptype (IN)	Data type. The following symbols are defined.
	WE_UBYTE ' 8-bit unsigned integer
	WE_SBYTE ' 8-bit signed integer
	WE_UWORD ' 16-bit unsigned integer
	WE_SWORD ' 16-bit signed integer
	WE_ULONG ' 32-bit unsigned integer
	WE_SLONG ' 32-bit signed integer
	WE_FLOAT ' 32-bit real number
	WE_DOUBLE ' 64-bit real number
gain (IN)	Gain (Range)
ofst (IN)	Offset
	Converts the data to physical values according to the following equation. X is the value of the acquisition data.
	gain x X + ofst
dt (IN)	Sampling interval (seconds)
startPoint (IN)	Starting point of the data to be used to compute the parameter values.
endPoint (IN)	Last point of the data to be used to compute the parameter values. Calculates the parameter values using the data in the range from startPoint to endPoint.

item (OUT) Waveform parameter pointer

```
Type MeasureItem        ' Computation result storage structure
  Channel As Double     ' CH number
  Max As Double         ' Maximum value
  Min As Double         ' Minimum value
  High As Double        ' High level
  Low As Double         ' Low level
  PP As Double          ' P-P value
  Ampl As Double        ' Amplitude
  Avg As Double         ' Average value
  Rms As Double         ' RMS value
  Middle As Double      ' Center value of the amplitude
  StdDev As Double      ' Standard deviation
  Oshoot As Double      ' Overshoot
  Ushoot As Double      ' Undershoot
  Rise As Double        ' Rise time
  Fall As Double        ' Fall time
  Freq As Double        ' Frequency
  Period As Double      ' Period
  Duty1 As Double       ' Duty cycle on the High side
  Duty2 As Double       ' Duty cycle on the Low side
  Width1 As Double      ' Width above the mesial value
  Width2 As Double      ' Width below the mesial value
End Type
```

Example (Visual Basic)

```
' Retrieve the Computation result of 1000 points of data at
' sampling rate of 0.5 sec, gain of 1.0, and offset of 0.0.
Dim ret As Integer
Dim item As Measureitem
Dim data (1000) As Single
ret = WeExecMeasureParam AcqData (data (0), WE_FLOAT, 1.0, 0.0,
0.5, 1, 1000, item)
```

6.12 Filter API for the 4-CH, 100 kS/s D/A Module WE7281

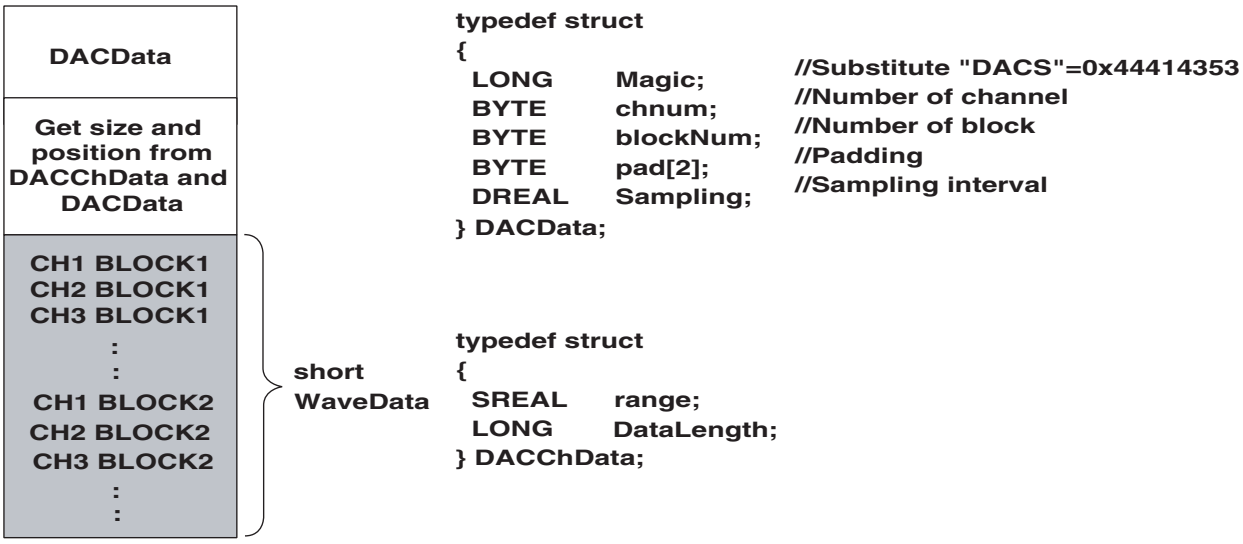
Description

This section describes the interface functions used to convert waveform data in wvf or csv format to the arbitrary waveform or sweep waveform data formats for the 4-CH, 100 kS/s D/A Module WE7281. These functions are used along with the WE Control API when transferring waveform data to the WE7281 module.

Data format

The WE7281 uses data in the following formats.

Data format	Description
s16	Arbitrary waveform data for the FG mode short x 65536
w32	Sweep waveform data for the FG mode UINT x 65536
w7281	Arbitrary waveform data for the AG mode (See figure below.)



WeWvf2S16GetSize

Description

Gets the byte size when waveform data retrieved using the specified parameter from the specified file (wvf or csv format), are converted to the arbitrary waveform data format for the FG mode.

Syntax

WeWvf2S16GetSize (ByVal *filename* As String, ByVal *ch* As Integer, ByVal *block* As Long, ByRef *size* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

filename (IN)	Name of the file containing the waveform data (wvf or csv format).
ch (IN)	Channel number of the input file. The counting origin is one.
block (IN)	Block number of the input file. The counting origin is zero.
size (OUT)	Data size (byte size).

Example (Visual Basic)

```
Dim size As Long
Dim ret As Integer
' Get the byte size when the Block0 data of CH1 of the wvf file
' saved by the WE7271 module are converted to the format used by
' the FG mode on the WE7281 module.
ret = WeWvf2S16GetSize("c:\we7271.wvf", 1, 0, size)
```

WeWvf2W32GetSize

Description

Gets the byte size when waveform data retrieved using the specified parameter from the specified file (wvf or csv format) are converted to the sweep waveform data format for the FG mode.

Syntax

WeWvf2W32GetSize(ByVal *filename* As String, ByVal *ch* As Integer, ByVal *block* As Long, ByRef *size* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

filename (IN)	Name of the file containing the waveform data (wvf or csv format).
ch (IN)	Channel number of the input file. The counting origin is one.
block (IN)	Block number of the input file. The counting origin is zero.
size (OUT)	Data size (byte size).

Example (Visual Basic)

```
Dim size As Long
Dim ret As Integer
' Get the byte size when the Block0 data of CH1 of the wvf file
' saved by the WE7271 module are converted to the sweep waveform
' data format used by the FG mode on the WE7281 module.
ret = WeWvf2W32GetSize("c:\we7271.wvf", 1, 0, size)
```

WeWvf2W7281GetSize

Description

Gets the byte size when waveform data retrieved using the specified parameter from the specified file (wvf or csv format), are converted to the arbitrary waveform data format for the AG mode.

Syntax

WeWvf2W7281GetSize(ByVal *filename* As String, ByVal *ch* As Integer, ByVal *block* As Long, ByRef *size* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

filename (IN)	Name of the file containing the waveform data (wvf or csv format).
ch (IN)	Channel number of the input file. The counting origin is one. Note that "&H7FFF" represents all channels.
block (IN)	Block number of the input file. The counting origin is zero. Note that "&H7FFFFFFF" represents all blocks.
size (OUT)	Data size (byte size).

Example (Visual Basic)

```
Dim size As Long
Dim ret As Integer
' Get the byte size when the Block0 data of CH1 of the wvf file
' saved by the WE7271 module are converted to the format used by
' the AG mode on the WE7281 module.
ret = WeWvf2W7281GetSize("c:\we7271.wvf", 1, 0, size)
```

WeWvf2S16

Description

Converts waveform data retrieved using the specified parameter from the specified file (wvf or csv format), to the arbitrary waveform data format for the FG mode.

Syntax

WeWvf2S16(ByVal *filename* As String, ByVal *ch* As Integer, ByVal *block* As Long, ByRef *buf* As Any, ByRef *size* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

filename (IN)	Name of the file containing the waveform data (wvf or csv format).
ch (IN)	Channel number of the input file. The counting origin is one.
block (IN)	Block number of the input file. The counting origin is zero.
buf (OUT)	Pointer to the data buffer.
size (IN/OUT)	Data buffer size/Actual size of the data retrieved (byte size).

Example (Visual Basic)

```
Dim size As Long
Dim ret As Integer
Dim bufsize As Long
' The byte size when the Block0 data of CH1 of the wvf file saved
' by the WE7271 module are converted to the s16 format used by the
' FG mode on the WE7281 module.
ret = WeWvf2S16GetSize("c:\we7271.wvf", 1, 0, size)
' Allocate the buffer using the retrieved data size.
bufsize = size/2
ReDim s16buf(bufsize) As Integer
' Convert the Block0 data of CH1 of the wvf file saved by the
' WE7271 module to the format used by the FG mode on the WE7281
' module.
ret = WeWvf2S16("c:\we7271.wvf", 1, 0, s16buf(0), size)
' Transfer the converted data to CH1 of the WE7281 module.
ret = WeSetControl (hMo, "FG:CH1:Load ARB", s16buf)
```

WeWvf2W32

Description

Converts waveform data retrieved using the specified parameter from the specified file (wvf or csv format), to the sweep waveform data format for the FG mode.

Syntax

WeWvf2W32(ByVal *filename* As String, ByVal *ch* As Integer, ByVal *block* As Long, ByRef *buf* As Any, ByRef *size* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

filename (IN)	Name of the file containing the waveform data (wvf or csv format).
ch (IN)	Channel number of the input file. The counting origin is one.
block (IN)	Block number of the input file. The counting origin is zero.
buf (OUT)	Pointer to the data buffer.
size (IN/OUT)	Data buffer size/Actual size of the data retrieved (byte size).

Example (Visual Basic)

```
Dim size As Long
Dim ret As Integer
Dim bufsize As Long
' The byte size when the Block0 data of CH1 of the wvf file saved
' by the WE7271 module are converted to the w32 format used by the
' FG mode on the WE7281 module.
ret = WeWvf2W32GetSize("c:\we7271.wvf", 1, 0, size)
' Allocate the buffer using the retrieved data size.
bufsize = size/4
ReDim w32buf(bufsize) As Long
' Convert the Block0 data of CH1 of the wvf file saved by the
' WE7271 module to the format used by the FG mode on the WE7281
' module.
ret = WeWvf2W32("c:\we7271.wvf", 1, 0, w32buf(0), size)
' Transfer the converted data to CH1 of the WE7281 module.
ret = WeSetControl (hMo, "FG:CH1:Load ARB", w32buf)
```

WeWvf2W7281

Description

Converts waveform data retrieved using the specified parameter from the specified file (wvf or csv format), to the arbitrary waveform data format for the AG mode.

Syntax

WeWvf2W7281(ByVal *filename* As String, ByVal *ch* As Integer, ByVal *block* As Long, ByRef *buf* As Any, ByRef *size* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

filename (IN)	Name of the file containing the waveform data (wvf or csv format).
ch (IN)	Channel number of the input file. The counting origin is one. Note that "&H7FFF" represents all channels.
block (IN)	Block number of the input file. The counting origin is zero. Note that "&H7FFFFFFF" represents all channels.
buf (OUT)	Pointer to the data buffer.
size (IN/OUT)	Data buffer size/Actual size of the data retrieved (byte size).

Example (Visual Basic)

```
Dim size As Long
Dim ret As Integer
' The byte size when the Block0 data of CH1 of the wvf file saved
' by the WE7271 module are converted to the format used by the
' AG mode on the WE7281 module.
ret = WeWvf2W7281GetSize("c:\we7271.wvf", 1, 0, size)
' Allocate the buffer using the retrieved data size.
ReDim w7281buf(size) As Byte
' Convert the Block0 data of CH1 of the wvf file saved by the
' WE7271 module to the format used by the AG mode on the WE7281
' module.
ret = WeWvf2W7281("c:\we7271.wvf", 1, 0, w7281buf(0), size)
' Transfer the converted data to CH1 of the WE7281 module.
ret = WeSetControl (hMo, "AG:Misc:Load Data", w7281buf)
```

6.13 Others

WeGetHandle

Description

Gets the module handle from the second parameter (the upper 16 bits is the station logical address, and the lower 16 bits is the slot number) of the event explained in chapter 5.

Syntax

WeGetHandle (ByVal *param* As Long, ByRef *hMo* As Long) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

param (IN)	The value of the second parameter of the event (lparam).
hMo (OUT)	Module handle to be retrieved.

Example (Visual Basic)

```
' Get the module handle.
Dim ret As Integer
Dim hMo As Long
ret = WeGetHandle (param, hMo)
```

WeIsNan

Description

Queries whether or not the data are non-numeric.

Syntax

USHORT WeIsNan (USHORT *ptype*, void* value) ;

Return value

Returns 0 when it is not non-numeric. Returns a nonzero number otherwise.

Parameters

ptype (IN)	Data type (WE_FLOAT or WE_DOUBLE)
value (IN)	Data specified

Example (Visual Basic)

```
' Check whether or not a value is non-numeric.
Dim ret As Integer
Dim data As Single
data = 9999999
ret = WeIsNan (WE_FLOAT, data)
```

WeLoadHostsFile

Description

Loads the host file (see "Note") containing the IP address of a measuring station. This allows the user to control a measuring station connected on a different segment.

Syntax

WeLoadHostsFile (ByVal name As String) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

filename (IN) Host file name.

Note:

Be sure to call WeLoadHostsFile after WeInit API.
An IP address is entered in a host file as follows. Any characters sequence proceded by a space or tab after the IP address are treated as a comment. To start a comment at the beginning of a line, precede the comment with a number sign (#).

```
---------------------------------------------------------------
# For example:
127.0.0.1
127.0.0.2
---------------------------------------------------------------
```

Example (Visual Basic)

```
Dim ret As Integer
ret = WeInit(WeEvent1.hWnd, "optical devicename=we7036",
WE_CONTROLLER)
ret = WeLoadHostsFile("c:\program files\we7000\hosts.txt")
```

WeTransAcqData

Description

This function transforms acquisition data (raw data) obtained using WeGetAcqData API or WeGetAcqDataEx API according to additional information. The additional information is acquired using the WeGetAcqDataInfo API. Only certain modules' raw data must be transformed using this API. For a list of modules whose data requires the transformation, see "Note" below.

Syntax

WeTransAcqData(ByVal chNum As Integer, ByVal recsize As Long, ByRef buf As Any, ByRef info As AcqDataInfo) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

chNum (IN)	Channel number (1 -)
recsize (IN)	The size in bytes of raw data obtained using WeGetAcqData or WeGetAcqDataEx.
buf (IN/OUT)	Pointer to the raw data buffer (IN)/transformed data is stored (OUT)
info (IN/OUT)	Pointer to the information obtained using WeGetAcqDataInfo (IN)/ transformed data information is stored (OUT)

Note:

This API performs data bit extraction and calculates the reciprocal on acquisition data per the values of the start bit and effective bit. The acquisition type is such that the data size will not change. For example, there is a change from WE_ULONG to WE_FLOAT, but not from WE_ULONG to WE_DOUBLE.

The only data that must be transformed by this API is the raw data obtained using the WeGetAcqData and WeGetAcqDataEx APIs. With the WeSaveAcqData API, data is already transformed when saved so no transformation by this API is necessary.
The following are the modules whose raw data must be transformed using this API.

- WE7081: performs bit extraction.
- WE7521: For measured frequency data, calculates the reciprocal.

Example (Visual Basic)

- Read out and transform data on all channels of the WE7081

```
Dim stHandle As Long
Dim moHandle As Long
Dim ret As Integer
'Get the station handle for the measuring station named "Station
1."
ret = WeOpenStation("Station1", stHandle)
'Open WE7081 module
ret = WeOpenModule(stHandle, "WE7081:1",1, moHandle)
' Waiting for completion of data acquisition after starting
measurement
......

Dim recSize As Long
Dim ptype As Integer
Dim info(4) As AcqDataInfo
Dim num As Integer
' Allocate a buffer for a memory length of 1000*20Ch
Dim buf(4000) As Double
' The number of bytes of the buffer is 1000 x 4 x 8 bytes
(WE_DOUBLE)
recSize = 1000*4*8
'Get the additional information and the raw data, and convert data
ret = WeGetAcqDataInfo(moHandle, -1, 0, info(0), num)
ret = WeGetAcqData(moHandle, -1, 0, recSize, buf(0), ptype)
ret = WeTransAcqData(num, recSize, buf(0), info(0))
```

- Read out and transform data on CH1 of the WE7521

```
Dim stHandle As Long
Dim moHandle As Long
Dim ret As Integer
' Get the station handle for the measuring station named "Station
1."
ret = WeOpenStation("Station1", stHandle)
'Open WE7521 module
ret = WeOpenModule(stHandle, "WE7521:1",1, moHandle)
' Waiting for completion of data acquisition after starting
measurement
```

```
Dim recSize As Long
Dim ptype As Integer
Dim info As AcqDataInfo
Dim num As Integer
' Allocate a buffer for a memory length of 1000
Dim buf(1000) As Float
' The number of bytes of the buffer is 1000 x 4 bytes (WE_FLOAT)
recSize = 1000*4
' Get the additional information and the raw data, and convert data
ret = WeGetAcqDataInfo(moHandle, 1, 0, info, num)
ret = WeGetAcqData(moHandle, 1, 0, recSize, buf(0), ptype)
ret = WeTransAcqData(1, recSize, buf(0), info)
```

6.14 C Language Syntax

Initialization

Function name	Syntax
WeInit	USHORT WeInit (HWND hWnd, char* comm, USHORT type) ;
WeExit	USHORT WeExit (void) ;

Handle/Link

Function name	Syntax
WeOpenStation	USHORT WeOpenStation (char* name, HANDLE* hSt) ;
WeOpenModule	USHORT WeOpenModule (HANDLE hSt, char* name, USHORT connection, HANDLE* hMo) ;
WeLinkStation	USHORT WeLinkStation (short num, HANDLE* hSt, HANDLE* hlSt) ;
WeLinkModule	USHORT WeLinkModule (short num, HANDLE* hMo, HANDLE* hlMo) ;
WeCloseHandle	USHORT WeCloseHandle (HANDLE handle) ;

Station control

Function name	Syntax
WeGetStationList	USHORT WeGetStationList (USHORT * num, StationList* list) ;
WeGetStationInfo	USHORT WeGetStationInfo (HANDLE handle, StationInfoEx* info) ;
WePower	USHORT WePower (HANDLE handle, BYTE sw) ;
WeRestart	USHORT WeRestart (HANDLE handle) ;
WeSetStationName	USHORT WeSetStationName (HANDLE hSt, char* name, char* comment) ;
WeGetStationName	USHORT WeGetStationName (HANDLE hSt, char* name, char* comment) ;
WeIdetifyStation	USHORT WeIdentifyStation (HANDLE hSt) ;
WeGetPower	USHORT WeGetPower (HANDLE hSt, BYTE* sw) ;

Module control

Function name	Syntax
WeGetModuleInfo	USHORT WeGetModuleInfo (HANDLE hMo, USHORT no, ModuleInfoEx* inf) ;

Settings

Function name	Syntax
WeInitSetup	USHORT WeInitSetup (HANDLE handle, USHORT no) ;
WeInitPreset	USHORT WeInitPreset (HANDLE handle, USHORT no) ;
WeSaveSetup	USHORT WeSaveSetup (HANDLE handle, USHORT no, char* filename) ;
WeLoadSetup	USHORT WeLoadSetup (HANDLE handle, USHORT no, char* filename) ;
WeCopySetup	USHORT WeCopySetup (HANDLE hSrcMo, HANDLE hDesMo) ;
WeCopyChSetup	USHORT WeCopyChSetup (HANDLE hMo, short srcCh, short desCh) ;
WeCopyChSetupEx	USHORT WeCopyChSetupEx (HANDLE hMo, int srcCh, int desStartCh, int desEndCh) ;
WeSetControl	USHORT WeSetControl (HANDLE hHandle, char* command, VARIANT param) ;
WeGetControl	USHORT WeGetControl (HANDLE hHandle, char* command, VARIANT* param) ;
WeSetControlEx	USHORT WeSetControlEx (HANDLE hHandle, char* command, BYTE type, UINT dataNum, void* data) ;
WeGetControlEx	USHORT WeGetControlEx (HANDLE hHandle, char* command, BYTE type, UINT dataNum, void* data) ;
WeSetQuryControl	USHORT WINAPI WeSetQueryControl (HANDLE hHanlde, char* command, USHORT type, UINT size, void* setvar, BYTE rtype, VARIANT* var);
WeSetScaleInfo	USHORT WeSetScaleInfo (HANDLE hMo, SHORT ch, LinearScaleInfo* info);
WeGetScaleInfo	USHORT WeGetScaleInfo (HANDLE hMo, SHORT ch, LinearScaleInfo* info);

Synchronized functions

Function name	Syntax
WeExecManualTrig	USHORT WeExecManualTrig (HANDLE hSt, BYTE busNo, BYTE pulse) ;
WeSetModuleBus	USHORT WeSetModuleBus (HANDLE handle, BYTE InItem, BYTE OutItem, BYTE InClock, BYTE ArmItem) ;
WeGetModuleBus	USHORT WeGetModuleBus (HANDLE handle, BYTE* InItem, BYTE* OutItem, BYTE* InClock, BYTE* ArmItem) ;
WeSetTrigBusLogic	USHORT WeSetTrigBusLogic (HANDLE handle, BYTE item, BYTE logic) ;
WeGetTrigBusLogic	USHORT WeGetTrigBusLogic (HANDLE handle, BYTE item, BYTE* logic) ;
WeSetEXTIO	USHORT WeSetEXTIO (HANDLE handle, BYTE direction1, BYTE direction2, BYTE clock) ;
WeGetEXTIO	USHORT WeGetEXTIO (HANDLE handle, BYTE* direction1, BYTE* direction2, BYTE* clock) ;
WeSetTRIGIN	USHORT WeSetTRIGIN (HANDLE handle, BYTE item, BYTE polarity) ;
WeGetTRIGIN	USHORT WeGetTRIGIN (HANDLE handle, BYTE* item, BYTE* polarity) ;
WeSetClockBusSource	USHORT WeSetClockBusSource (HANDLE handle, BYTE source) ;
WeGetClockBusSource	USHORT WeGetClockBusSource (HNDLE handle, BYTE* source) ;
WeSetRcvTrigPacket	USHORT WeSetRcvTrigPacket (HANDLE handle, USHORT num, PacketList* name) ;
WeSetSndTrigPacket	USHORT WeSetSndTrigPacket (HANDLE handle USHORT num, PacketList* name) ;
WeFireTrigPacket	USHORT WeFireTrigPacket (HANDLE handle, USHORT item) ;
WeSetSndClockPacket	USHORT WeSetSndClockgPacket (HANDLE handle, USHORT num, PacketList* name) ;
WeSetRcvClockPacket	USHORT WeSetRcvClockgPacket (HANDLE handle, USHORT num, PacketList* name) ;
WeFireClockPacket	USHORT WeFireClockgPacket (HANDLE handle) ;
WeOutputEXTIOEvent	USHORT WeOutputEXTIOEvent (HANDLE handle, BYTE pulse) ;
WeExecManualArming	USHORT WeExecManualArming (HANDLE hSt, BYTE item) ;
WeSetArmingSource	USHORT WeSetArmingSource (HANDLE hSt, BYTE item) ;
WeGetArmingSource	USHORT WeGetArmingSource (HANDLE hSt, BYTE* item) ;

GUI control

Function name	Syntax
WeShowModuleWindow	USHORT WeShowModuleWindow (HANDLE hMo, HANDLE* hWnd) ;
WeCloseModuleWindow	USHORT WeCloseModuleWindow (HANDLE hMo) ;
WeIsModuleWindow	USHORT WeIsModuleWindow (HANDLE hMo, BYTE* sw) ;
WeShowTrigWindow	USHORT WeShowTrigWindow (HANDLE hSt, HANDLE* hWnd) ;
WeCloseTrigWindow	USHORT WeCloseTrigWindow (HANDLE hSt) ;
WeIsTrigWindow	USHORT WeIsTrigWindow (HANDLE hSt, BYTE* sw) ;
WeShowLinearScaleWindow	USHORT WeShowLinearScaleWindow (HANDLE hMo, HANDLE* hWnd);
WeCloseLinearScaleWindow	USHORT WeCloseLinearScaleWindow (HANDLE hMo) ;
WeIsLinearScaleWindow	USHORT WeIsLinearScaleWindow (HANDLE hMo, BYTE* sw);

Measurement control

Function name	Syntax
WeStart	USHORT WeStart (HANDLE handle) ;
WeStop	USHORT WeStop (HANDLE handle) ;
WeStartSingle	USHORT WeStartSingle (HANDLE hMo, UINT timeOut) ;
WeStartWithEvent	USHORT WeStartWithEvent (HANDLE handle) ;
WeIsRun	USHORT WeIsRun (HANDLE hMo, BYTE status) ;
WeLatchData	USHORT WeLatchData (HANDLE hMo) ;
WeGetAcqDataInfo	USHORT WeGetAcqDataInfo (HANDLE hMo, SHORT ch, int blockNo, AcqDataInfo* inf, USHORT* infNum) ;
WeGetAcqData	USHORT WeGetAcqData (HANDLE hMo, SHORT ch, int blockNo, UINT* recSize, void* buf, USHORT* type) ;
WeGetScaleData	USHORT WeGetScaleData (HANDLE hMo, SHORT ch, int blockNo, ScalingParam *param, UINT* recSize, USHORT type, void* buf) ;
WeGetScaleDataEx	USHORT WeGetScaleDataEx (HANDLE hMo, SHORT ch, int blockNo, UINT* recSize, USHORT ptype, void* buf) ;
WeGetMeasureParam	USHORT WeGetMeasureParam (HANDLE hMo, SHORT ch, int blockNo, UINT start, UINT end, MeasureItem* inf, USHORT* infNum) ;
WeSaveAcqData	USHORT WeSaveAcqData (HANDLE hMo, SHORT ch, int blockNo, chan* filename, USHORT type) ;
WeSaveScaleData	USHORT WeSaveScaleData (HANDLE hMo, SHORT ch, int blockNo, ScalingParam* param, char* filename, USHORT type);
WeSaveScaleDataEx	USHORT WeSaveScaleDataEx (HANDLE hMo, SHORT ch, int blockNo, char* filename, USHORT htype) ;
WeSaveAsciiData	USHORT WeSaveAsciiData (HANDLE hMo, SHORT ch, int blockNo, char* filename, USHORT type) ;
WeSaveScaleAsciiData	USHORT WeSaveScaleAsciiData (HANDLE hMo, SHORT ch, int blockNo, ScalingParam* param, char* filename, USHORT htype) ;
WeSaveScaleAsciiDataEx	USHORT WeSaveScaleAsciiDataEx (HANDLE hMo, SHORT ch, INT blockNo, char* filename, USHORT htype) ;
WeGetCurrentData	USHORT WeGetCurrentData (HANDLE hMo, SHORT ch, UINT* recSize, void* buf, USHORT* type) ;
WeGetScaleCurrentData	USHORT WeGetScaleCurrentData (HANDLE hMo, SHORT ch, USHORT ptype, ScalingParam* param, UINT* recSize, void* buf) ;
WeGetScaleCurrentDataEx	USHORT WeGetScaleCurrentDataEx (HANDLE hMo, SHORT ch, USHORT ptype, UINT* recSize, void* buf) ;
WeGetAcqDataEx	USHORT WeGetAcqDataEx (HANDLE hMo, SHORT ch, int blockNo, UINT start, UINT end, UINT ppNum, USHORT interpolation, UINT* recSize, void* buf, USHORT* type) ;
WeGetAcqDataSize	USHORT WeGetAcqDataSize (HANDLE hMo, SHORT ch, int blockNo, UINT* pointNum, UINT* dataSize, USHORT* type, USHORT* chNum) ;
WeSaveAcqHeader	USHORT WeSaveAcqHeader (HANDLE hMo, SHORT ch, char* filename) ;
WeSavePatternData	USHORT WeSavePatternData (HANDLE hMo, short ch, char* filename) ;

Function name	Syntax
WeLoadPatternData	USHORT WeLoadPatternData (HANDLE hMo, short ch, char* filename) ;
WeStartEx	USHORT WeStartEx (HANDLE handle, UINT blockLen, BYTE blockCount, UINT acqCount, USHORT mode, UINT* pHandle) ;
WeStopEx	USHORT WeStopEx (HANDLE handle, UINT pHandle) ;
WeLoadPatterDataEx	USHORT WeLoadPatternDataEx (HANDLE hMo, char* command, char* filename, short ch, int blockNo) ;
WeSetOverRun	USHORT WeSetOverRun (HANDLE hMo, BYTE sw) ;
WeGetOverRun	USHORT WeGetOverRun (HANDLE hMo, BYTE* sw) ;

Events

Function name	Syntax
WeCreateEvent	USHORT WeCreateEvent (HANDLE hMo, UINT pattern, UINT mode, UINT* evHandle) ;
WeSetEventPattern	USHORT WeSetEventPattern (HANDLE hMo, UINT evHandle, UINT pattern) ;
WeResetEventPattern	USHORT WeResetEventPattern (HANDLE hMo, UINT evHandle, UINT pattern) ;
WeSetEventMode	USHORT WeSetEventMode (HANDLE hMo, UINT evHandle, UINT mode) ;
WeReleaseEvent	USHORT WeReleaseEvent (HANDLE hMo, UINT evHandle) ;

Waveform Parameter Computation

Function name	Syntax
WeExecMeasureParam	USHORT WeExecMeasureParam (void* data, USHORT ptype, double dt, UINT start, UINT end, MeasureItem* item) ;
WeExecMeasureParamAcqData	USHORT WeExecMeasureParamAcqData (void* data, USHORT ptype, double gain, double ofst, double dt, UINT start, UINT end, MeasureItem* item) ;

Filter API for the 4-CH, 100 kS/s D/A Module WE7281

Function name	Syntax
WeWvf2S16GetSize	USHORT WeWvf2S16GetSize(char* infile, short ch, int block, UINT* size);
WeWvf2W32GetSize	USHORT WeWvf2W32GetSize(char* filename, short ch, int block, UINT* size);
WeWvf2W7281GetSize	USHORT WeWvf2W7281GetSize(char* filename, short ch, int block, UINT* size);
WeWvf2S16	USHORT WeWvf2S16(char* filename, short ch, int block, void* buf, UINT* size);
WeWvf2W32	USHORT WeWvf2W32(char* filename, short ch, int block, void* buf, UINT* size);
WeWvf2W7281	USHORT WeWvf2W7281(char* filename, short ch, int block, void* buf, UINT* size);

Others

Function name	Syntax
WeGetHandle	USHORT WeGetHandle (DWORD param, HANDLE* hMo) ;
WeIsNan	USHORT WeIsNan (USHORT type, void* value) ;

7.1 Error Codes

Controller Side

Error Code (Hexadecimal)	Description
1 to 19 (0x1 to x13)	Communication driver error codes.
100 (0x64)	Failed initialization.
101 (0x65)	Buffer is full.
102 (0x66)	No communication interface board.
103 (0x67)	Station does not exist.
104 (0x68)	Event receive error.
105 (0x69)	Abnormal interrupt.
106 (0x6A)	Abnormal received packet.
107 (0x6B)	Buffer size too small.
110 (0x6E)	Requested reply packet not received.
111 (0x6F)	RplySyncExec API error.
112 (0x70)	SettingQuery API error.
113 (0x71)	Cannot process segmented packets.
114 (0x72)	No such function.
115 (0x73)	Not all segmented packets have been received.
200 (0xC8)	Execution error on station.
201 (0xC9)	Execution error on module.
202 (0xCA)	No station name.
203 (0xCB)	Duplicate station names.
204 (0xCC)	Bad station name.
205 (0xCD)	Module does not exist.
206 (0xCE)	Bad module name.
207 (0xCF)	Invalid slot specification.
300 (0x12C)	Function not supported.
301 (0x12D)	Function not supported.
302 (0x12E)	Acquisition data does not exist.
400 (0x190)	Failed to allocate dynamic memory.
401 (0x191)	Failed in sending messages between threads.
402 (0x192)	Failed to create thread.
403 (0x193)	File open error.
404 (0x194)	File read/write error.
405 (0x195)	File access error.
500 (0x1F4)	Handle error.
501 (0x1F5)	ASCII command cannot be found.
502 (0x1F6)	Bad parameter.
503 (0x1F7)	Cannot open the module handle of a child.
504 (0x1F8)	Invalid setting.

Common Error Codes for the Station and Modules

Error Code (Hexadecimal)	Description
4097 (0x1001)	Parameter unnecessary.
4098 (0x1002)	Value was limited.
4099 (0x1003)	Over the range.
4100 (0x1004)	Lost preset information.
8193 (0x2001)	Bad module ID.
8194 (0x2002)	Bad control ID.
8195 (0x2003)	Bad command ID.
8196 (0x2004)	Parameter necessary.
8197 (0x2005)	Parameter type different.
8198 (0x2006)	Too many parameters.
8199 (0x2007)	Not enough parameters.
8200 (0x2008)	Setting conflict.
8201 (0x2009)	Hardware not installed.
8202 (0x200A)	Program error.
8203 (0x200B)	Lost calibration value.
8204 (0x200C)	Failed self test.
8205 (0x200D)	Failed calibration.
8206 (0x200E)	Data exceeds the range.
8207 (0x200F)	Failed measurement.
8208 (0x2010)	Hardware error.
8209 (0x2011)	Temperature error.
8210 (0x2012)	Cooling fan stopped.
8211 (0x2013)	System memory overflow.
8212 (0x2014)	Bad data base format.
8213 (0x2015)	Bad data base version.
8214 (0x2016)	Bad handle was provided.
8215 (0x2017)	Failed to get handle.
8216 (0x2018)	Measurement aborted.
8217 (0x2019)	No measurement data.
8218 (0x201A)	Timeout.
8219 (0x201B)	Data overrun occurred.
8225 to 8232 (0x202#)	Program error in SLOT# (#: Slot number 1 to 8).
8241 to 8248 (0x203#)	Bad version in SLOT# (#: Slot number 1 to 8).

Filter API for the 4-CH, 100 kS/s D/A Module WE7281

Symbol	Error Code (Hexadecimal)	Description
AsciiRSLT_NORMAL	0 (0x0000)	Normal termination.
AsciiRSLT_CANT_OPEN	1 (0x0001)	Failed to open file.
AsciiRSLT_CANT_ALLOC	2 (0x0002)	Failed to allocate memory.
AsciiRSLT_CANT_READ	9 (0x0009)	Failed to retrieve data (includes “?,” for example).Vresolution, Voffset, VUnit, VPlusOverData, VminusOverData, VillegalData, VMaxData, VMinData, Hresolution, HUnit, Data, Time.
AsciiRSLT_BOUNDARY	10 (0x000A)	Boundary specification error.
AsciiRSLT_READ_ERROR	11 (0x000B)	Access error to the wvf file.
AsciiRSLT_LINK_ERROR	13 (0x000D)	Failed to link to the DLL.
AsciiRSLT_UNSUPPORTED_FUNCTION	14 (0x000E)	Function not supported by this DLL.
AsciiRSLT_ZERO_BLOCKSIZE	15 (0x000F)	Block size is zero.
AsciiRSLT_ERROR	255 (0x00FF)	Other error.

8.1 About the Command Names

Each measurement module has ASCII commands that can be used to control its individual functions through the API. These commands are used in conjunction with the WeSetControl, WeSetControlEx, WeGetContorl, and WeGetControllEx functions as described in chapter 6.

ASCII commands are named using the **title names** that appear in the GUI window of a module. The exact character string of **the title** is assigned to the commands, so that the user can easily write the program while looking at the GUI window of the module.

The commands are organized in a hierarchy according to the following basic command rules.

[Tab name:][Group name:]Title name
Example: CH<x>:Trigger:Mask

Tab names are the names of the tab control. However, tab names are not specified except for slot tabs (tab names that include "SLOT" in the name). This is because the tab names change dynamically depending on the slot position of the modules.

Group names can sometimes become nested depending on if the name encloses a set of controls with a frame. In this case, the hierarchy becomes "Group name:...:Group name" such as "AG:Misc:Load:Auto."

The title names are the names given to each control.

8.2 Control parameters

The control parameter is the third parameter of the SetControl function. It is used to specify the conditions or values that are essential in executing the commands. Upper and lower case letters are distinguished for the control parameters. Spaces are ignored.

The setup values are specified with a character string.

Example

```
' Define the variable used to pass the setup value.
Dim val As Variant
' Val variable is used as a string.
val = "1.234"
' Set the offset value.
WeSetControl(hMo, "offset", val)
```

Numerical values are specified using the NR format as defined by ANSIX3.42-1975. If the value is outside the allowed range, it is set to the closest allowed value. If the precision of the value is too great, it is rounded.

The results of queries are returned as a string.

Example

```
' Define the variable for storing the result.
Dim val As Variant
' Get the offset value.
WeGetControl(hMo, "offset", val)
' VarType is a function used to check the variable type.
' In this case, the returned value, vtype, is of string type.
vtype = VarType(val)
```

However, character strings cannot be used to set or query the controls, if one of the following terms appears besides the parameter in the command description.

- WE_UBYTE(8-bit unsigned integer), WE_SBYTE(8-bit signed integer), WE_UWORD(16-bit unsigned integer)
- WE_SWORD(16-bit signed integer), WE_ULONG(32-bit unsigned integer)
- WE_SLONG(32-bit signed integer), WE_FLOAT(32-bit real number), WE_DOUBLE(64-bit real number)

The following example shows the CH<x>:Memory:Output command for the WE7131 module.

CH<x>:Memory:Output

Description

Sets the output pattern or queries the current setting.

Parameter

WE_ULONG*Memory length

For setting the parameter

Set the value using the appropriate type and call Set Control API (see example 1) or the SetControlEx API (see example 2).

Example 1

```
Dim buf(512) As Long
For i = 0 to 512-1
 buf(i) = i
Next i
ret = WeSetControl (hMo, "CH1:Memory:Output", buf)
```

Example 2

```
Dim buf(512) As Long
For i = 0 to 512-1
 buf(i) = i
Next i
ret = WeSetControlEx (hMo, "CH1:Memory:Output", WE_ULONG, 512,
buf(0))
```

For getting the parameter

Define a Variant-type variable and call GetControl API (see example 3) or allocate the buffer for retrieving the result and call GetControlEx API (see example 4).

Example 3

```
Dim buf As Variant
ret = WeGetControl (hMo, "CH1:Memory:Output", buf)
' The buf varible becomes an array of Long type.
```

Example 4

```
Dim buf(512) As Long
ret = WeGetControlEx (hMo, "CH1:Memory:Output", WE_ULONG, 512,
buf(0))
```

8.3 WE7021 GP-IB Controller

ASCII Commands

ASCII Command	Description
ADR	Sets the target address or queries the current setting.
TERM	Sets the terminator or queries the current setting.
DATA	Sets the binary data or queries the current setting.
MSG	Sets the ASCII data or queries the current setting.
TIMEOUT	Sets the timeout time or queries the current setting.
GET	Executes the device trigger.
POLL	Queries serial polling.
REN	Switches between remote and local (0: local, 1: remote).
LLO	Executes local lockout.
IFC	Clears the interface.
DCL	Clears the device.
SDC	Executes SDC.
GTL	Executes Go to Local.
OPTION	Sets the transmit/receive option or queries the current setting.
BLOCK	Sets the block data to IEE488.2 format or queries the current setting.
ERR	Queries the error information.

ADR

Description

Sets the address of the GP-IB device to be controlled or queries the current setting. The GP-IB device is selected using this command before executing other commands.

Parameter

0 to 30

Example (Visual Basic)

```
' Set the control to the GP-IB device at address 1.
ret = WeSetControl (hMo, "ADR", "1")
Dim value As Variant
ret = WeGetControl (hMo, "ADR", value)
```

value → 1

TERM

Description

Sets the terminator or queries the current setting.

Parameter

CR+LF | CR | LF | EOI

Example (Visual Basic)

```
' Set the terminator to LF.
ret = WeSetControl (hMo, "TERM", "LF")
Dim value As Variant
ret = WeGetControl (hMo, "TERM", value)
```

value → LF

DATA

Description

Sets the binary data or queries the current setting.

Parameter

WE_ULONG (the number of transmitted data is fixed to "1")

Example (Visual Basic)

```
' Retrieve block data from the connected device.
Dim stHandle As Long
Dim pBuf As Variant
Dim rSize As Long
rSize = 2002
Dim err As Integer
Dim gpibHandle As Long
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", stHandle)
' Open the first WE7021 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7021", 1, gpibHandle)
ret = WeSetQueryControl (gpibHandle, "DATA", WE_ULONG,  1,  rSize,
WE_UWORD, pBuf)
```

MSG

Description

Sets the ASCII data or queries the current setting.

Parameter

WE_SBYTE

Example (Visual Basic)

```
' Transmit the string to the device.
ret = WeSetControl (hMo, "MSG", "CH1:Offset?")
Dim value As Variant
ret = WeGetControl (hMo, "MSG", value)
```

value → 1.0E+00

TIMEOUT

Description

Sets the timeout time or queries the current setting.

Parameter

0.01 to 60

Example (Visual Basic)

```
' Set the timeout time to 10 s.
ret = WeSetControl (hMo, "TIMEOUT", 10)
Dim value As Variant
ret = WeGetControl (hMo, "TIMEOUT", value)
```

value → 10.000E+00

GET

Description
Executes the device trigger.

Parameter
None

Example (Visual Basic)
```
Dim value As Variant
ret = WeSetControl (hMo, "GET", value)
```

POLL

Description
Queries serial polling.

Parameter
WE_UWORD

Example (Visual Basic)
```
Dim value As Variant
ret = WeGetControl (hMo, "POLL", value)
```
value → 65

REN

Description
Sets the device remote and local or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Executes local lockout.
ret = WeSetControl (hMo, "REN", "On")
Dim value As Variant
ret = WeGetControl (hMo, "REN", value)
```
value → On

LLO

Description
Executes local lockout.

Parameter
None

Example (Visual Basic)
```
Dim value As Variant
ret = WeSetControl (hMo, "LLO", value)
```

IFC

Description
Clears the interface.

Parameter
None

Example (Visual Basic)
```
Dim value As Variant
ret = WeSetControl (hMo, "IFC", value)
```

DCL

Description
Clears the device.

Parameter
None

Example (Visual Basic)
```
Dim value As Variant
ret = WeSetControl (hMo, "DCL", value)
```

SDC

Description
Executes selected device clear.

Parameter
None

Example (Visual Basic)
```
Dim value As Variant
ret = WeSetControl (hMo, "SDC", value)
```

GTL

Description
Executes Go to Local.

Parameter
None

Example (Visual Basic)
```
Dim value As Variant
ret = WeSetControl (hMo, "GTL", value)
```

OPTION

Description

Sets the transmit/receive option or queries the current setting

Parameter

WE_UBYTE (0 - Do not perform pre-processing or post-processing, 1 - Perform pre-processing only, 2 - Perform post-processing only, 3 - perform both pre-processing and post-processing)
Pre-processing during transmission: Talker specification
Post-processing during transmission: Terminator transmission/Talker release
Pre-processing during reception: Listener specification
Post-processing during: Listener release

Example (Visual Basic)

```
' Set the transmit/receive option to 1.
ret = WeSetControl (hMo, "OPTION", 1)
Dim value As Variant
ret = WeGetControl (hMo, "OPTION", value)
```

value → 1

BLOCK

Description

Sets the block data to IEE488.2 format or queries the current setting.

Parameter

WE_UBYTE

Example (Visual Basic)

```
' Set the block data to the connected device.
Dim stHandle As Long
Dim ret As Integer
Dim gpibHandle As Long
Dim value(1000) As Byte
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", stHandle)
' Open the first WE7021 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7021", 1, gpibHandle)
ret = WeSetControl (hMo, "BLOCK", value)
' Using the WE7021 module, retrieve the block data from the
' connected device.
Dim stHandle As Long
Dim ret As Integer
Dim gpibHandle As Long
Dim value As Variant
' Get the station handle of the station named "Station 1."
ret = WeOpenStation ("Station1", stHandle)
' Open the first WE7021 module with a link number of 1.
ret = WeOpenModule (stHandle, "WE7021", 1, gpibHandle)
ret = WeGetControl (hMo, "BLOCK", value)
```

ERR

Description

Queries the error information of the WE7021 module (controller).

Parameter

WWE_UWORD

Note:

For the error codes and their details, see "Module-specific Error Codes" on the next page.

Example (Visual Basic)

```
Dim value As Variant
ret = WeGetControl (hMo, "ERR", value)
```

value → 1

Valid Acquisition Modes

This module is not a data acquisition module.

Acquisition Restrictions

This module is not a data acquisition module.

Valid Common Measurement Control API

API	Valid?
WeStart	No
WeStop	No
WeStartSingle	No
WeStartWithEvent	No
WeIsRun	No
WeLatchData	No
WeGetAcqDataInfo	No
WeGetAcqData	No
WeGetScaleData	No
WeGetMeasureParam	No
WeSaveAcqData	No
WeSaveScaleData	No
WeSaveAsciiData	No
WeGetCurrentData	No
WeGetAcqDataEx	No
WeGetAcqDataSize	No
WeSaveAcqHeader	No
WeSavePatternData	No
WeLoadPatternData	No
WeLoadPatternDataEx	No
WeStartEx	No
WeStopEx	No

Valid Common Events

Event	Event Description
WWE_EV_MEASEND	No
WE_EV_BLOCKEND	No
WE_EV_MEASABORT	No
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

Event	Event Description
0x00010000	Service request generation event

Module-specific Error Codes

The error codes listed below are error messages for the GP-IB Controller Module WE7021. The error messages can be retrieved using the ERR ASCII command (see previous page).

Err Code	Err Description
0x1	Timeout occurred.
0x2	Cannot detect active device.
0x4	Buffer overflow occurred.
0x8	Detected invalid data format.

8.4 WE7081 CAN Bus Interface Module

ASCII Commands

ASCII Command	Description
Operation Mode	Sets the operation mode or queries the current setting.
Bit Rate	Sets bit rate or queries the current setting.
Sample Point:Select	Sets the sample point number or queries the current setting.
Sample Point:Sample Point	Queries the sample point value of the selected sample point number.
Sample Point:Time Quanta	Queries the Time Quanta value of the selected sample point number.
Sample Point:BTR0	Queries the BTR0 value of the selected sample point number.
Sample Point:BTR1	Queries the BTR1 value of the selected sample point number.
Other:BTR0	Sets BTR0 when bit rate is set to Other or queries the current setting.
Other:BTR1	Sets BTR1 when bit rate is set to Other or queries the current setting.
Other:Bit rate	Sets the bit rate value when bit rate is set to Other or queries the current setting.
Other:Sample Point	Queries the sample point value when bit rate is set to Other.
Other:Time Quanta	Queries the Time Quanta value when bit rate is set to Other.
Other: No. of Sample Points	Queries the number of sample points when bit rate is set to Other.
Other: SJW	Queries the SJW value when bit rate is set to Other.
Queue Overflow	Sets whether to continue the measurement when the input/output queue buffer of the CAN chip overflows or queries the current setting.
Acknowledge	Sets whether to make the Acknowledge bit dominant after reading the message or queries the current setting.
ArbitrationLostErrStatus	Queries the information when an arbitration lost error event is received.
BusErrStatus	Queries the information when a bus error event is received.
ErrorLog	Queries the error log.

Note

The following ASCII commands specific to the operation mode is valid only when the corresponding operation mode is active. For example, "Output:Interval" can be used only when the operation mode is set to Output.

Acquisition:

ASCII Command	Description
Acquisition:Mode	Sets the acquisition mode or queries the current setting.
Acquisition:Trigger:Source	Sets the trigger source or queries the current setting.
Acquisition:Trigger:Pretrigger	Sets the amount of pretrigger or queries the current setting.
Acquisition:Trigger:Hold Off	Sets the hold off time or queries the current setting.
Acquisition:Time Base	Sets the time base or queries the current setting.
Acquisition:Sampling Interval	Sets the sampling interval or queries the current setting.
Acquisition:Memory Partition	Sets the number of memory partitions or queries the current setting.
Acquisition:Record Length	Sets the record length or queries the current setting.
Acquisition:No. of Acquisitions	Sets the number of acquisitions or queries the current setting.
Acquisition:Out:Mode	Sets the output mode or queries the current setting.
Acquisition:Out:OneShot:Ext	Sets the output frame format in OneShot mode or queries the current setting.
Acquisition:Out:OneShot:ID	Sets the output frame ID in OneShot mode or queries the current setting.
Acquisition:Out:OneShot:DLC	Sets the output frame DLC in OneShot mode or queries the current setting.
Acquisition:Out:OneShot:Data	Sets the output frame data in OneShot mode or queries the current setting.
Acquisition:Out:OneShot:Send	Sends the output frame when in OneShot mode.
Acquisition:Out:AutoSend	Turns ON/OFF the auto send function during OneShot mode or Manual mode or queries the current setting.

ASCII Command	Description
Acquisition:Out:Period	Sets the transmission mode during Periodic mode or queries the current setting.
Acquisition:Out:No.OfRepeat	Sets the number of output repetitions during Sequential mode or queries the current setting.
Acquisition:Out:SendAll	Sends the data frame of all the channels when in Manual mode.
Acquisition:Out:Output	Turns ON/OFF the output when in Periodic mode, Triggered mode, Remote mode, or Sequential mode or queries the current setting.
Acquisition:Out:Sequence:Target	Sets the target line number of the sequence table of Sequential mode for setting and querying or queries the current setting.
Acquisition:Out:Sequence:On	Turns ON/OFF the sequence table output of Sequential mode or queries the current setting.
Acquisition:Out:Sequence:Ext	Sets the Ext of the sequence table of Sequential mode or queries the current setting.
Acquisition:Out:Sequence:ID	Sets the ID of the sequence table of Sequential mode or queries the current setting.
Acquisition:Out:Sequence:DLC	Sets the DLC of the sequence table of Sequential mode or queries the current setting.
Acquisition:Out:Sequence:Data	Sets the Data of the sequence table of Sequential mode or queries the current setting.
Acquisition:Out:Sequence:Itvl	Sets the Itvl of the sequence table of Sequential mode or queries the current setting.
Acquisition:Out:Sequence:Trg	Sets the Trg of the sequence table of Sequential mode or queries the current setting.
Acquisition:Out:Sequence:Send	Sets the Send of the sequence table of Sequential mode or queries the current setting.
Acquisition:CH:Target	Sets the target channel number for setting and querying or queries the current setting.
Acquisition:CH:On	Turns ON/OFF the acquisition of the target channel or queries the current setting.
Acquisition:CH:Ext	Sets the message format of the target channel or queries the current setting.
Acquisition:CH:ID	Sets the ID of the target channel or queries the current setting.
Acquisition:CH:SB	Sets the start bit of the target channel or queries the current setting.
Acquisition:CH:LN	Sets the data length of the target channel or queries the current setting.
Acquisition:CH:Value Type	Sets the data type of the target channel or queries the current setting.
Acquisition:CH:Endian	Sets the data endian of the target channel or queries the current setting.
Acquisition:CH:Trigger:Type	Sets the trigger type of the target channel or queries the current setting.
Acquisition:CH:Trigger:Level	Sets the trigger level of the target channel or queries the current setting.
Acquisition:CH:Alarm:Type	Sets the alarm type of the target channel or queries the current setting.
Acquisition:CH:Alarm:High	Sets the alarm high level of the target channel or queries the current setting.
Acquisition:CH:Alarm:Low	Sets the alarm low level of the target channel or queries the current setting.
Acquisition:CH:Rq:Itvl	Sets the remote frame transmission interval of the target channel or queries the current setting.
Acquisition:CH:Output:On	Turns ON/OFF the output of the target channel or queries the current setting.
Acquisition:CH:Output:Value	Sets the output frame data of the target channel or queries the current setting.
Acquisition:CH:Output:DLC	Sets the output frame DLC of the target channel or queries the current setting.
Acquisition:CH:Output:Send	Sends the frame of the channel.
Acquisition:Data Size	Queries the data size.
Acquisition:Trigger Combination	Sets the trigger combination or queries the current setting.
Acquisition:Integer NAN	Sets the "Not-A-Number" value for integers or queries the current setting.

Note

Acquisition:CH:xxx functions against a preset target channel number. Therefore, you must set the target channel before using Acquisition:CH:xxx. For example, if you are going to set or query CH12, issue WeSetControl(hMo, "Acquisition:CH:Active", 12) beforehand.

Frame Acquisition:

ASCII Command	Description
Frame:ID Filter:Standard	Sets the receive filter for standard format or queries the current setting.
Frame:ID Filter:Extended	Sets the receive filter for extended format or queries the current setting.
Frame:No. of Frames	Sets the number of acquisition frames or queries the current setting.
Frame:No. of Stored Frames	Queries the number of frames that have been acquired.
Frame:Trigger:Source	Sets the trigger source or queries the current setting.
Frame:Trigger:Type	Sets the internal trigger type or queries the current setting.
Frame:Trigger:Level	Sets the internal trigger level or queries the current setting.
Frame:Trigger:Ext	Sets the internal trigger message format or queries the current setting.
Frame:Trigger:ID	Sets the internal trigger message ID or queries the current setting.
Frame:Trigger:SB	Sets the start bit of the internal trigger or queries the current setting.
Frame:Trigger:LN	Sets the data length of the internal trigger or queries the current setting.
Frame:Trigger:Value Type	Sets the data type of the internal trigger or queries the current setting.
Frame:Trigger:Endian	Sets the data endian of the internal trigger or queries the current setting.
Frame:Trigger:Manual Trigger	Generates a manual trigger.

Output:

ASCII Command	Description
Output:Mode	Sets the output mode or queries the current setting.
Output:Interval	Sets the output interval or queries the current setting.
Output:No. of Output Samples	Sets the number of output samples or queries the current setting.
Output:Trigger:Source	Sets the trigger source or queries the current setting.
Output:Trigger:Type	Sets the internal trigger type or queries the current setting.
Output:Trigger:Level	Sets the internal trigger level or queries the current setting.
Output:Trigger:Ext	Sets the internal trigger message format or queries the current setting.
Output:Trigger:ID	Sets the internal trigger message ID or queries the current setting.
Output:Trigger:SB	Sets the start bit of the internal trigger or queries the current setting.
Output:Trigger:LN	Sets the data length of the internal trigger or queries the current setting.
Output:Trigger:Value Type	Sets the data type of the internal trigger or queries the current setting.
Output:Trigger:Endian	Sets the data endian of the internal trigger or queries the current setting.
Output:Trigger:Manual Trigger	Generates a manual trigger.
Output:Output	Turns ON/OFF the output or queries the current setting.
Output:CH:Target	Sets the channel number for setting and querying or queries the current setting.
Output:CH:On	Turns ON/OFF the acquisition of the target channel or queries the current setting.
Output:CH:Ext	Sets the message format of the target channel or queries the current setting.
Output:CH:ID	Sets the ID of the target channel or queries the current setting.
Output:CH:SB	Sets the start bit of the target channel or queries the current setting.
Output:CH:LN	Sets the data length of the target channel or queries the current setting.

ASCII Command	Description
Output:CH:Value Type	Sets the data type of the target channel or queries the current setting.
Output:CH:Endian	Sets the data endian of the target channel or queries the current setting.
Output:CH:Sweep Pattern	Sets the sweep pattern of the target channel or queries the current setting.
Output:CH:DataStart	Sets the sweep start value of the target channel or queries the current setting.
Output:CH:DataEnd	Sets the sweep end value of the target channel or queries the current setting.
Output:CH:Step	Sets the step for changing the sweep of the target channel or queries the current setting.

Note

Output:CH:xxx functions against a preset target channel number. Therefore, you must set the target channel before using Output:CH:xxx. For example, if you are going to set or query CH12, issue WeSetControl(hMo, "Output:CH:Active", 12) beforehand.

Frame Output:

ASCII Command	Description
Frame Output:Mode	Sets the output mode or queries the current setting.
Frame Output:No. of Stored Frames	Queries the number frames that have been loaded into the memory.
Frame Output:Repeat	Sets the number of output repetitions or queries the current setting.
Frame Output:Trigger:Source	Sets the trigger source or queries the current setting.
Frame Output:Trigger:Type	Sets the internal trigger type or queries the current setting.
Frame Output:Trigger:Level	Sets the internal trigger level or queries the current setting.
Frame Output:Trigger:Ext	Sets the internal trigger message format or queries the current setting.
Frame Output:Trigger:ID	Sets the internal trigger message ID or queries the current setting.
Frame Output:Trigger:SB	Sets the start bit of the internal trigger or queries the current setting.
Frame Output:Trigger:LN	Sets the data length of the internal trigger or queries the current setting.
Frame Output:Trigger:Value Type	Sets the data type of the internal trigger or queries the current setting.
Frame Output:Trigger:Endian	Sets the data endian of the internal trigger or queries the current setting.
Frame Output:Trigger:Manual Trigger	Generates a manual trigger.
Frame Output:Output	Turns ON/OFF the output or queries the current setting.

Operation Mode

Description

Sets the operation mode or queries the current setting.

Parameters

Acquisition | FrameAcquisition | Output | FrameOutput

Example (Visual Basic)

```
' Set the operation mode to frame acquisition mode.
ret = WeSetControl(hMo, "Operation Mode", "FrameAcquisition")
Dim value As Variant
ret = WeGetControl(hMo, "Operation Mode", value)
```

value → "FrameAcquisition"

Bit Rate

Description

Sets bit rate (bps) or queries the current setting.

Parameters

1M | 800k | 500k | 250k | 125k | 100k | 83.3k | 62.5k | 50k | 33.3k | 20k | 10k | Other

Example (Visual Basic)

```
' Set the bit rate to 500 kbps.
ret = WeSetControl(hMo, "Bit Rate", "500k")
Dim value As Variant
ret = WeGetControl(hMo, "Bit Rate", value)
```

value → 500000

Sample Point:Select

Description

Sets the sample point number or queries the current setting.

Parameters

1 to N

Note:

This command is invalid if bit rate is set to Other. For a description of the parameters, see appendix 1, "Sample Point Table" in the user's manual that came with the module.

Example (Visual Basic)

```
' Set the sample point number to 5.
ret = WeSetControl(hMo, "Sample Point:Select", 5)
Dim value As Variant
ret = WeGetControl(hMo, "Sample Point:Select", value)
```

value → 5

Sample Point:Sample Point

Description

Queries the sample point value (%) of the selected sample point number.

Parameters

50 to 100(%)

Note:

This command is invalid if bit rate is set to Other.

Example (Visual Basic)

```
' Query the sample point value.
Dim value As Variant
ret = WeGetControl(hMo, "Sample Point:Sample Point", value)
```

value → 75

Sample Point:Time Quanta

Description
Queries the Time Quanta value of the selected sample point number.

Parameters
8 to 25

Note:
This command is invalid if bit rate is set to Other.

Example (Visual Basic)
```
' Query the Time Quanta value.
Dim value As Variant
ret = WeGetControl(hMo, "Sample Point:Time Quanta", value)
```
value → 15

Sample Point:BTR0

Description
Queries the BTR0 value of the selected sample point number.

Parameters
0x40 to 0x7F

Note:
This command is invalid if bit rate is set to Other.

Example (Visual Basic)
```
' Query the BTR0 value.
Dim value As Variant
ret = WeGetControl(hMo, "Sample Point:BTR0", value)
```
value → 65 (=&h41)

Sample Point:BTR1

Description
Queries the BTR1 value of the selected sample point number.

Parameters
0x00 to 0x7F

Note:
This command is invalid if bit rate is set to Other.

Example (Visual Basic)
```
' Query the BTR1 value.
Dim value As Variant
ret = WeGetControl(hMo, "Sample Point:BTR1", value)
```
value → 24 (=&h18)

Other:BTR0

Description

Sets BTR0 when bit rate is set to Other or queries the current setting.

Parameters

0x00 to 0xFF

Note:

This command is valid only when bit rate is set to Other. For a description of the parameters, see section 1.2, "Operation Mode and Common Setup Items" in the user's manual that came with the module.
Use decimal notation to pass parameters as character strings. To pass integers, use either decimal or hexadecimal notation.

Example (Visual Basic)

```
' Set the BTR0 value to &h42.
ret = WeSetControl(hMo, "Other:BTR0", &h42)
Dim value As Variant
ret = WeGetControl(hMo, "Other:BTR0", value)
```

value → 66 (=&h42)

Other:BTR1

Description

Sets BTR1 when bit rate is set to Other or queries the current setting.

Parameters

0x00 to 0xFF

Note:

This command is valid only when bit rate is set to Other. For a description of the parameters, see section 1.2, "Operation Mode and Common Setup Items" in the user's manual that came with the module.
Use decimal notation to pass parameters as character strings. To pass integers, use either decimal or hexadecimal notation.

Example (Visual Basic)

```
' Set the BTR1 value to &h16.
ret = WeSetControl(hMo, "Other:BTR1", &h16)
Dim value As Variant
ret = WeGetControl(hMo, "Other:BTR1", value)
```

value → 22 (=&h16)

Other:Bit rate

Description

Sets the bit rate (bps) value when bit rate is set to Other or queries the current setting.

Parameters

10.0E3 to 1.0E6

Note:

This command is valid only when bit rate is set to Other. The unit is bps.

Example (Visual Basic)

```
' Query the bit rate when set to Other.
Dim value As Variant
ret = WeGetControl(hMo, "Other:Bit rate", value)
```

value → 500000 (= 500 kbps)

Other:Sample Point

Description

Queries the sample point value when bit rate is set to Other.

Parameters

50 to 100(%)

Note:

This command is valid only when bit rate is set to Other.

Example (Visual Basic)

```
' Query the sample point value when bit rate is set to Other.
Dim value As Variant
ret = WeGetControl(hMo, "Other:Sample Point", value)
```

value → 60 (= 60%)

Other:Time Quanta

Description

Queries the Time Quanta value when bit rate is set to Other.

Parameters

8 to 25

Note:

This command is valid only when bit rate is set to Other.

Example (Visual Basic)

```
' Query the Time Quanta value when bit rate is set to Other.
Dim value As Variant
ret = WeGetControl(hMo, "Other:Time Quanta", value)
```

value → 12

Other:No. of Sample Points

Description

Sets the number of sample points when bit rate is set to Other or queries the current setting.

Parameters

1 | 3

Note:

This command is valid only when bit rate is set to Other.

Example (Visual Basic)

```
' Query the number of sample points when bit rate is set to Other.
Dim value As Variant
ret = WeGetControl(hMo, "Other:Sample Point", value)
```

value → 1

Other:SJW

Description

Queries the SJW value when bit rate is set to Other.

Parameters

1 to 4

Note:

This command is valid only when bit rate is set to Other.

Example (Visual Basic)

```
' Query the SJW value when bit rate is set to Other.
Dim value As Variant
ret = WeGetControl(hMo, "Other:SJW", value)
```

value → 2

Queue Overflow

Description

Sets whether to continue the measurement when the input/output queue buffer of the CAN chip overflows or queries the current setting.

Parameters

Continue | Stop

Example (Visual Basic)

```
' Set the queue overflow handling to continue.
ret = WeSetControl(hMo, "Queue Overflow", "Continue")
Dim value As Variant
ret = WeGetControl(hMo, "Queue Overflow", value)
```

value → "Continue"

Acknowledge

Description

Sets whether to make the Acknowledge bit dominant after reading the message or queries the current setting.

Parameters

On | Off

Example (Visual Basic)

```
' Set Acknowledge to On.
ret = WeSetControl(hMo, "Acknowledge", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acknowledge", value)
```

value → "On"

ArbitrationLostErrStatus

Description

Queries the information when an arbitration lost error event is received.

Parameters

0x00 to 0x1F

Note:

The arbitration lost error code is defined for each lost bit position as shown in the table below.

Code (Hex)	Lost Bit	Code (Hex)	Lost Bit
00	bit 1 of ID	10	bit 15 of ID ;(2)
01	bit 2 of ID	11	bit 16 of ID ;(2)
02	bit 3 of ID	12	bit 17 of ID ;(2)
03	bit 4 of ID	13	bit 18 of ID ;(2)
04	bit 5 of ID	14	bit 19 of ID ;(2)
05	bit 6 of ID	15	bit 20 of ID ;(2)
06	bit 7 of ID	16	bit 21 of ID ;(2)
07	bit 8 of ID	17	bit 22 of ID ;(2)
08	bit 9 of ID	18	bit 23 of ID ;(2)
09	bit 10 of ID	19	bit 24 of ID ;(2)
0A	bit 11 of ID	1A	bit 25 of ID ;(2)
0B	bit SRTR ;(1)	1B	bit 26 of ID ;(2)
0C	bit IDE	1C	bit 27 of ID ;(2)
0D	bit 12 of ID ;(2)	1D	bit 28 of ID ;(2)
0E	bit 13 of ID ;(2)	1E	bit 29 of ID ;(2)
0F	bit 14 of ID ;(2)	1F	bit RTR ;(2)

(1) RTR bit of the standard frame format
(2) Extended frame only

Example (Visual Basic)

```
' Query the error code of the arbitration lost error.
Dim value As Variant
ret = WeGetControl(hMo, "ArbitrationLostErrStatus", value)
```

value → 16 (=&h10)

BusErrStatus

Description

Queries the information when a bus error event is received.

Parameters

0x00 to 0x1F

Note:

The bus error codes are defined as shown in the table below.

Bit Number	Item	Code (Bin): Meaning	
0 to 4	Segment	00010:ID.28 to ID.21	01110:ID.4 to ID.0
		00011:start of frame	01111:ID.12 to ID.5
		00100:bit SRTR	10001:active error flag
		00101:bit IDE	10010:intermission
		00110:ID.20 to ID.18	10011:tolerate dominant bit
		00111:ID.17 to ID.13	10110:passive error flag
		01000:CRC sequence	10111:error delimiter
		01001:reserve bit 0	11000:CRC delimiter
		01010:data field	11001:acknowledge slot
		01011:data length code	11010:end of frame
		01100:bit RTR	11011:acknowledge delimiter
		01101:reserve bit 1	11100:overload flag
5	Direction	0: occurred during transmission	1: occurred during reception
6 to 7	Error code	00:bit error	10:stuff error
		01:form error	11:other type of error

Example (Visual Basic)

```
' Query the error code of the bus error.
Dim value As Variant
ret = WeGetControl(hMo, "BusErrStatus", value)
```

value → 67 (=&h43 =01000011)

Error Log

Description

Queries the error log.

Parameters

Error log character string

Example (Visual Basic)

```
' Query the error log.
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Mode", value)
```

value → "Bus off"

Note:

For a description of the error character strings, see appendix 3, "Error Code" in the user's manual that came with the module.

Acquisition:Mode

Description

Sets the acquisition mode in acquisition mode or queries the current setting.

Parameters

Triggered | Free Run

Example (Visual Basic)

```
' Set the acquisition mode to free run.
ret = WeSetControl(hMo, "Acquisition:Mode", "Free Run")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Mode", value)
```

value → "Free Run"

Acquisition:Trigger:Source

Description

Sets the trigger source in acquisition mode or queries the current setting.

Parameters

Internal | BUSTRG

Example (Visual Basic)

```
' Set the trigger source to BUSTRG.
ret = WeSetControl(hMo, "Acquisition:Trigger:Source", "BUSTRG")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Trigger:Source", value)
```

value → "BUSTRG"

Acquisition:Trigger:Pretrigger

Description

Sets the amount of pretrigger in acquisition mode or queries the current setting.

Parameters

0 to record length – 1

Example (Visual Basic)

```
' Set the amount of pretrigger to 100.
ret = WeSetControl(hMo, "Acquisition:Trigger:Pretrigger", "100")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Trigger:Pretrigger", value)
```

value → 100

Acquisition:Trigger:Hold Off

Description

Sets the hold off time in acquisition mode or queries the current setting.

Parameters

Record length to 8388608

Example (Visual Basic)

```
' Set the hold off time to 100.
ret = WeSetControl(hMo, "Acquisition:Trigger:Hold Off", "100")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Trigger:Hold Off", value)
```

value → 100

Acquisition:Time Base

Description
Sets the time base in acquisition mode or queries the current setting.

Parameters
Internal | BUSCLK

Example (Visual Basic)
```
' Set the time base to BUSCLK.
ret = WeSetControl(hMo, "Acquisition:Time Base", "BUSCLK")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Time Base", value)
```
value → "BUSCLK"

Acquisition:Sampling Interval

Description
Sets the sampling interval in acquisition mode or queries the current setting.

Parameters
0.001 to 10s

Example (Visual Basic)
```
' Set the sampling interval to 10 ms.
ret = WeSetControl(hMo, "Acquisition:Sampling Interval", "0.01")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Sampling Interval", value)
```
value → 0.01

Acquisition:Memory Partition

Description
Sets the number of memory partitions in acquisition mode or queries the current setting.

Parameters
1 to 256

Example (Visual Basic)
```
' Set the number of memory partitions to 2.
ret = WeSetControl(hMo, "Acquisition:Memory Partition", "2")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Memory Partition", value)
```
value → 2

Acquisition:Record Length

Description
Sets the record length in acquisition mode or queries the current setting.

Parameters
When in trigger mode: 100 to 8,388,608/the number of memory partitions
When in free run mode: 1 to 8,388,608

Example (Visual Basic)

```
' Set the record length to 1000.
ret = WeSetControl(hMo, "Acquisition:Record Length", "1000")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Record Length", value)
```

value → 1000

Acquisition:No. of Acquisitions

Description

Sets the number of acquisitions in acquisition mode or queries the current setting.

Parameters

0 to 2,147,483,647

Note:

If you set the value to 0, the number of acquisitions is set to infinite.

Example (Visual Basic)

```
' Set the number of acquisitions to 100.
ret = WeSetControl(hMo, "Acquisition:No. of Acquisitions", "100")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:No. of Acquisitions", value)
```

value → 100

Acquisition:Out:Mode

Description

Sets the output mode in acquisition mode or queries the current setting.

Parameters

OneShot | Manual | Periodic | Triggered | Remote | Sequential

Example (Visual Basic)

```
' Set the output mode to Manual.
ret = WeSetControl(hMo, "Acquisition:Out:Mode", "Manual")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Out:Mode", value)
```

value → "Manual"

Acquisition:Out:OneShot:Ext

Description

Sets the output data frame format when the output mode is set to OneShot in acquisition mode.

Parameters

On | Off

Note:

Specify Off to select the standard format, On to select the extended format.

Example (Visual Basic)

```
' Set the OneShot output data frame format to extended format.
ret = WeSetControl(hMo, "Acquisition:OneShot:Ext", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:OneShot:Ext", value)
```

value → "On"

Acquisition:Out:OneShot:ID

Description

Sets the output data frame ID when the output mode is set to OneShot in acquisition mode.

Parameters

When using standard format: 0 to 0x7ff
When using extended format: 0 to 0x1fffffff

Note:

Use decimal notation to pass parameters as character strings. To pass integers, use either decimal or hexadecimal notation.

Example (Visual Basic)

```
' Set the OneShot output data frame ID to 0x123.
ret = WeSetControl(hMo, "Acquisition:OneShot:ID", &h123)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:OneShot:ID", value)
```

value → 291(=&h123)

Acquisition:Out:OneShot:DLC

Description

Sets the output data frame DLC when the output mode is set to OneShot in acquisition mode.

Parameters

1byte to 8byte or R0

Example (Visual Basic)

```
' Set the OneShot output data frame DLC to 8.
ret = WeSetControl(hMo, "Acquisition:OneShot:DLC", "8byte")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:OneShot:DLC", value)
```

value → 8byte

Acquisition:Out:OneShot:Data

Description

Sets the output data frame data when the output mode is set to OneShot in acquisition mode.

Parameters

Setting is Dim buf(7) As Byte.
Query is Dim buf As Variant.

Example (Visual Basic)

```
' Set the OneShot output data frame data to 0x11, 0x22, 0x33, 0x44,
0x55, 0x66, 0x77, 0x88.
Dim buf(8) As Byte
buf(0) = &h11
buf(1) = &h22
buf(2) = &h33
buf(3) = &h44
buf(4) = &h55
buf(5) = &h66
buf(6) = &h77
buf(7) = &h88
ret = WeSetControl(hMo, "Acquisition:OneShot:DATA", buf)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:OneShot:DATA", value)
```

value → &h11, &h22, &h33, &h44, &h55, &h66, &h77, &h88

Acquisition:Out:OneShot:Send

Description

Sends the data frame when the output mode is set to OneShot in acquisition mode.

Parameters

None

Example (Visual Basic)

```
' Sends the OneShot output data frame.
ret = WeSetControl(hMo, "Acquisition:OneShot:Send", "")
```

Acquisition:Out:AutoSend

Description

Turns ON/OFF auto transmission of the data when the output data is changed when the output mode is set to OneShot or Manual in acquisition mode.

Parameters

Off | On

Note:

This command is valid only when the output mode is set to OneShot or Manual.

Example (Visual Basic)

```
' Turn ON auto transmission of the data when the output data is
changed.
ret = WeSetControl(hMo, "Acquisition:Out:AutoSend", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Out:AutoSend", value)
```

value → On

Acquisition:Out:Period

Description

Sets the data frame transmission interval when the output mode is set to Periodic in acquisition mode.

Parameters

0.1s to 100.0s

Example (Visual Basic)

```
' Set the data frame transmission interval in Periodic mode to 500
ms.
ret = WeSetControl(hMo, "Acquisition:Out:Period", "500E-3")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Out:Period", value)
```

value → 500E-3

Acquisition:Out:No.OfRepeat

Description

Sets the number of output repetitions of the table when the output mode is set to Sequential in acquisition mode.

Parameters

0 to 2147483647

Note:

If you set the value to 0, the number of output repetitions is set to infinite.

Example (Visual Basic)

```
' Set the number of output repetitions to 10.
ret = WeSetControl(hMo, "Acquisition:Out:No.OfRepeat", "10")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Out:No.OfRepeat", value)
```

value → 10

Acquisition:Out:SendAll

Description

Sends the data frame of all the channels when the output mode is set to Manual in acquisition mode.

Parameters

None

Example (Visual Basic)

```
' Send the data frame of all the channels in Manual mode.
ret = WeSetControl(hMo, "Acquisition:Out:SendAll", "")
```

Acquisition:Out:Output

Description

Turns ON/OFF the output when the output mode is set to Periodic, Triggered, Remote, or Sequential in acquisition mode.

Parameters

Off | On

Note:

This command is valid only when the output mode is set to Periodic, Triggered, Remote, and Sequential.

Example (Visual Basic)

```
' Turn ON the output.
ret = WeSetControl(hMo, "Acquisition:Out:Output", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Out:Output", value)
```

value → On

Acquisition:Out:Sequence:Target

Description

Sets the target line number of the sequence table of acquisition mode for setting and querying or queries the current setting.

Parameters

1 to 1024

Note:

The setting and query commands specific to a line in the sequence table (Acquisition:Out:Sequence:xxx) are applied to the line number specified with this command.

Example (Visual Basic)

```
' Set the target line number to 3.
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Target", "3")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition::Out:Sequence:Target", value)
```

value → 3

Acquisition:Out:Sequence:On

Description

Turns ON/OFF the output sequence table output in acquisition mode or queries the current setting.

Parameters

On | Off

Example (Visual Basic)

```
' Turn On the output of line 3.
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Target", "3")
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:On", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition::Out:Sequence:On", value)
```

value → On

Acquisition:Out:Sequence:Ext

Description

Sets the message format of the output sequence table in acquisition mode or queries the current setting.

Parameters

On | Off

Note:

Specify Off to select the standard format, On to select the extended format.

Example (Visual Basic)

```
' Set the message format of line 3 to Ext.
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Target", "3")
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Ext", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Out:Sequence:Ext", value)
```

value → "On"

Acquisition:Out:Sequence:ID

Description

Sets the ID of the output sequence table in acquisition mode or queries the current setting.

Parameters

When using standard format: 0 to 0x7ff
When using extended format: 0 to 0x1fffffff

Example (Visual Basic)

```
' Set the ID of line 3 to 0x123.
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Target", "3")
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:ID", &h123)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition::Out:Sequence:ID", value)
```

value → 291 (=&h123)

Acquisition:Out:Sequence:DLC

Description

Sets the DLC of the output sequence table in acquisition mode or queries the current setting.

Parameters

1byte to 8byte or R0

Note:

The DLC is set to remote frame is R0 is specified.

Example (Visual Basic)

```
' Set the DLC of line 3 to 8.
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Target", "3")
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:DLC", 8)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition::Out:Sequence:DLC", value)
```

value → 8

Acquisition:Out:Sequence:Data

Description

Sets the data of the output sequence table in acquisition mode or queries the current setting.

Parameters

Setting is Dim buf(7) As Byte.
Query is Dim buf As Variant.

Example (Visual Basic)

```
' Set the data of line 3 to 0x11, 0x22, 0x33, 0x44, 0x55, 0x66,
0x77, and 0x88.
Dim buf(8) As Byte
buf(0) = &h11
buf(1) = &h22
buf(2) = &h33
buf(3) = &h44
buf(4) = &h55
buf(5) = &h66
buf(6) = &h77
buf(7) = &h88
ret = WeSetControl(hMo, "Acquisition:OneShot:DATA", buf)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:OneShot:DATA", value)
```

value → &h11, &h22, &h33, &h44, &h55, &h66, &h77, &h88

Acquisition:Out:Sequence:Itvl

Description

Sets the Itvl of the output sequence table in acquisition mode or queries the current setting.

Parameters

0.1s to 100s

Example (Visual Basic)

```
' Set the Itvl of line 3 to 500 ms.
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Target", "3")
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Itvl", 500E-3)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition::Out:Sequence:Itvl", value)
```

value → 500E-3

Acquisition:Out:Sequence:Trg

Description
Turns ON/OFF the trigger wait of the output sequence table in acquisition mode or queries the current setting.

Parameters
On | Off

Example (Visual Basic)
```
' Turn On the trigger wait of line 3.
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Target", "3")
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Trg", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition::Out:Sequence:Trg", value)
```
value → On

Acquisition:Out:Sequence:Send

Description
Sends the data frame of the output sequence table in acquisition mode.

Parameters
None

Example (Visual Basic)
```
' Sends the data frame of line 3.
ret = WeSetControl(hMo, "Acquisition::Out:Sequence:Target", "3")
ret = WeSetControl(hMo, "Acquisition::Out:Sequencet:Send", "")
```

Acquisition:CH:Target

Description
Sets the target channel number for setting and querying in acquisition mode or queries the current setting.

Parameters
1 to 64

Note:

The setting and query commands specific to a channel in acquisition mode (Acquisition:CH:xxx) are applied to the channel number specified with this command.

Example (Visual Basic)
```
' Set the target CH number for setting and querying to 10.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Target", value)
```
value → 10

Acquisition:CH:On

Description

Turns ON/OFF the acquisition of the target channel in acquisition mode or queries the current setting.

Parameters

On | Off

Example (Visual Basic)

```
' Turn ON the acquisition of CH10.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:On", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:On", value)
```

value → On

Acquisition:CH:Ext

Description

Sets the message format of the target channel in acquisition mode or queries the current setting.

Parameters

On | Off

Note:

Specify Off to select the standard format, On to select the extended format.

Example (Visual Basic)

```
' Set the message format of CH10 to Ext.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:On", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:On", value)
```

value → "On"

Acquisition:CH:ID

Description

Sets the ID of the target channel in acquisition mode or queries the current setting.

Parameters

When using standard format: 0 to 0x7ff
When using extended format: 0 to 0x1fffffff

Example (Visual Basic)

```
' Set the CH10 ID to 0x123.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:ID", &h123)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:ID", value)
```

value → 291 (=&h123)

Acquisition:CH:SB

Description

Sets the start bit of the target channel in acquisition mode or queries the current setting.

Parameters

0 to 63

Example (Visual Basic)

```
' Set the start bit of CH10 to 16.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:SB", "16")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:SB", value)
```

value → 16

Acquisition:CH:LN

Description

Sets the data length of the target channel in acquisition mode or queries the current setting.

Parameters

1 to 64

Example (Visual Basic)

```
' Set the data length of CH10 to 8.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:LN", "8")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:LN", value)
```

value → 8

Acquisition:CH:Value Type

Description

Sets the data type of the target channel in acquisition mode or queries the current setting.

Parameters

Unsigned | Signed | Float

Example (Visual Basic)

```
' Set the data type of CH10 to Signed.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Value Type", "Signed")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Value Type", value)
```

value → "Signed"

Acquisition:CH:Endian

Description

Sets the data endian of the target channel in acquisition mode or queries the current setting.

Parameters

Big | Little

Example (Visual Basic)

```
' Set the data endian of CH10 to Big.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Endian", "Big")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Endian", value)
```

value → "Big"

Acquisition:CH:Trigger:Type

Description

Sets the trigger type of the target channel in acquisition mode or queries the current setting. This function is valid in trigger mode.

Parameters

Off | Rise | Fall | Both | High | Low | X

Example (Visual Basic)

```
' Set the trigger type of CH10 to Rise.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Trigger:Type", "Rise")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Trigger:Type", value)
```

value → "Rise"

Acquisition:CH:Trigger:Level

Description

Sets the trigger level of the target channel in acquisition mode or queries the current setting. This function is valid in trigger mode.

Parameters

The selectable range varies depending on the data type and data length.

Example (Visual Basic)

```
' Set the trigger level of CH10 to 1000.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Trigger:Level", 1000)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Trigger:Level", value)
```

value → 1000

Acquisition:CH:Alarm:Type

Description

Sets the alarm type of the target channel in acquisition mode or queries the current setting.
This function is valid in free run mode.

Parameters

Off | Rise | Fall | Both | High | Low | In | Out

Example (Visual Basic)

```
' Set the alarm type of CH10 to High.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Alarm:Type", "High")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Alarm:Type", value)
```

value → "High"

Acquisition:CH:Alarm:High

Description

Sets the alarm high level of the target channel in acquisition mode or queries the current setting.
This function is valid in free run mode.

Parameters

The selectable range varies depending on the data type and data length.

Example (Visual Basic)

```
' Set the alarm high level of CH10 to 2000.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Alarm:High", 2000)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Alarm:High", value)
```

value → 2000

Acquisition:CH:Alarm:Low

Description

Sets the alarm low level of the target channel in acquisition mode or queries the current setting.
This function is valid in free run mode.

Parameters

N-byte value (the selectable range varies depending on the data type and data length)

Example (Visual Basic)

```
' Set the alarm low level of CH10 to 1000.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Alarm:Low", 1000)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Alarm:Low", value)
```

value → 1000

Acquisition:CH:Rq:On

Description

Turns ON/OFF the remote frame output of the target channel in acquisition mode or queries the current setting.

Parameters

On | Off

Example (Visual Basic)

```
' Turn ON the remote frame output of CH10.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Rq:On", On)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Rq:On", value)
```

value → On

Acquisition:CH:Rq:Itvl

Description

Sets the remote frame output interval of the target channel in acquisition mode in multiples of the sampling interval or queries the current setting.

Parameters

1 to 9999

Example (Visual Basic)

```
' Set the remote frame output interval of CH10 to 5.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "10")
ret = WeSetControl(hMo, "Acquisition:CH:Rq:Itvl", "5")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Rq:Itvl", value)
```

value → 5

Acquisition:CH:Output:On

Description

Turns ON/OFF the data frame output of the target channel in acquisition mode or queries the current setting.

Parameters

On | Off

Example (Visual Basic)

```
' Turn ON the data frame output of CH11.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "11")
ret = WeSetControl(hMo, "Acquisition:CH:Output:On", On)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Output:On", value)
value → On
```

Acquisition:CH:Output:Value

Description

Sets the output data of the data frame output of the target channel in acquisition mode or queries the current setting.

Parameters

The selectable range varies depending on the data type and bit length.

Example (Visual Basic)

```
' Set the output data of CH11 to 12.5.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "11")
ret = WeSetControl(hMo, "Acquisition:CH:Output:Value", 12.5)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Output:Value", value)
value → 1.25000E001
```

Acquisition:CH:Output:DLC

Description

Sets the data frame output DLC of the target channel in acquisition mode or queries the current setting.

Parameters

1 to 8

Example (Visual Basic)

```
' Set the output data frame DLC of CH11 to 4.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "11")
ret = WeSetControl(hMo, "Acquisition:CH:Output:Data", 4)
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:CH:Output:Data", value)
value → 4
```

Acquisition:CH:Output:Send

Description

Sends the frame of the target channel in acquisition mode.

Parameters

None

Note:

This command is valid when the output mode is set to Manual, and the output of the target channel is turned ON.

Example (Visual Basic)

```
' Execute the frame transmission of CH4.
ret = WeSetControl(hMo, "Acquisition:CH:Target", "4")
ret = WeSetControl(hMo, "Acquisition:CH:Send", "")
```

Acquisition:Data Size

Description

Queries the data size in acquisition mode.

Parameters

1 | 2 | 4 | 8

Example (Visual Basic)

```
' Query the data size.
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Data Size", value)
```

value → 8

Acquisition:Trigger Combination

Description

Sets the trigger combination in acquisition mode or queries the current setting.
This function is valid in trigger mode.

Parameters

AND | OR

Example (Visual Basic)

```
' Set the trigger combination to OR.
ret = WeSetControl(hMo, "Acquisition:Trigger Combination", "OR")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Trigger Combination", value)
```

value → "OR"

Acquisition:Integer NAN

Description

Sets the "Not-A-Number" value for integers in acquisition mode or queries the current setting.

Parameters

Zero | Fullbit

Example (Visual Basic)

```
' Set the NAN value to Fullbit.
ret = WeSetControl(hMo, "Acquisition:Integer NAN", "Fullbit")
Dim value As Variant
ret = WeGetControl(hMo, "Acquisition:Integer NAN", value)
```

value → "Fullbit"

Frame:ID Filter:Standard

Description

Sets the read data filter for standard format in frame acquisition mode or queries the current setting.

Parameters

Set the filter using 11 characters in binary notation.
The character X denotes Don't Care.
Example) 11111110XXX

Example (Visual Basic)

```
' Set the read data filter for standard format to 11111110XXX.
ret = WeSetControl(hMo, "Frame:ID Filter:Standard", "11111110XXX")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:ID Filter:Standard", value)
```

value → "11111110XXX "

Frame:ID Filter:Extended

Description

Sets the read data filter for extended format in frame acquisition mode or queries the current setting.

Parameters

Set the filter using 29 characters in binary notation.
The character X denotes Don't Care.

Example (Visual Basic)

```
' Set the read data filter for extended format as follows:
00000001110110001OXXXXXXXXXXX
ret = WeSetControl(hMo, "Frame:ID Filter:Extended",
"00000001110110001OXXXXXXXXXX")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:ID Filter:Extended", value)
```

value → "00000001110110001OXXXXXXXXXX"

Frame:No. of Frames

Description

Sets the number of frames to be acquired in frame acquisition mode or queries the current setting.

Parameters

0 to 2147483647

Example (Visual Basic)

```
' Set the number of frames to be acquired to 10000.
ret = WeSetControl(hMo, "Frame:No. of Frames", "10000")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:No. of Frames", value)
```

value → 10000

Frame:No. of Stored Frames

Description
Queries the number of frames that have been acquired in frame acquisition mode.

Parameters
0 to 4294967295
The value returns to 0 after 4294967295.

Example (Visual Basic)
```
' Query the number of frames that have been acquired.
Dim value As Variant
ret = WeGetControl(hMo, "Frame:No. of Stored Frames", value)
```
value → 12345

Frame:Trigger:Source

Description
Sets the trigger source in frame acquisition mode or queries the current setting.

Parameters
Internal | BUSTRG

Example (Visual Basic)
```
' Set the trigger source to BUSTRG.
ret = WeSetControl(hMo, "Frame:Trigger:Source", "BUSTRG")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:Source", value)
```
value → "BUSTRG"

Frame:Trigger:Type

Description
Sets the internal trigger type in frame acquisition mode or queries the current setting.

Parameters
Off | Rise | Fall | Both | High | Low | X

Example (Visual Basic)
```
' Set the internal trigger type to Rise.
ret = WeSetControl(hMo, "Frame:Trigger:Type", "Rise")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:Type", value)
```
value → "Rise"

Frame:Trigger:Level

Description
Sets the internal trigger level in frame acquisition mode or queries the current setting.

Parameters
The selectable range varies depending on the data type.

Example (Visual Basic)

```
' Set the internal trigger level to 1000 V.
ret = WeSetControl(hMo, "Frame:Trigger:Level", "1000")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:Level", value)
```

value → 1000

Frame:Trigger:Ext

Description

Sets the frame format of the internal trigger in frame acquisition mode or queries the current setting.

Parameters

On | Off

Note:

Specify Off to select the standard format, On to select the extended format.

Example (Visual Basic)

```
' Set the message format of the internal trigger to extended
format.
ret = WeSetControl(hMo, "Frame:Trigger:Ext", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:Ext", value)
```

value → "On"

Frame:Trigger:ID

Description

Sets the message ID of the internal trigger in frame acquisition mode or queries the current setting.

Parameters

When using standard format: 0 to 0x7ff
When using extended format: 0 to 0x1fffffff

Example (Visual Basic)

```
' Set the message ID of the internal trigger to 0x123.
ret = WeSetControl(hMo, "Frame:Trigger:ID", &h123)
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:ID", value)
```

value → 291 (=&h123)

Frame:Trigger:SB

Description

Sets the start bit of the internal trigger in frame acquisition mode or queries the current setting.

Parameters

0 to 63

Example (Visual Basic)

```
' Set the start bit of the internal trigger to 32.
ret = WeSetControl(hMo, "Frame:Trigger:SB", "32")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:SB", "32")
```

value → 32

Frame:Trigger:LN

Description

Sets the data length of the internal trigger in frame acquisition mode or queries the current setting.

Parameters

1 to 64

Example (Visual Basic)

```
' Set the date length of the internal trigger to 32.
ret = WeSetControl(hMo, "Frame:Trigger:LN", "32")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:LN", value)
```

value → 32

Frame:Trigger:Value Type

Description

Sets the data type of the internal trigger in frame acquisition mode or queries the current setting.

Parameters

Unsigned | Signed | Float

Example (Visual Basic)

```
' Set the data type of the internal trigger to Signed.
ret = WeSetControl(hMo, "Frame:Trigger:Value Type", "Signed")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:Value Type", value)
```

value → "Signed"

Frame:Trigger:Endian

Description

Sets the data endian of the internal trigger in frame acquisition mode or queries the current setting.

Parameters

Big | Little

Example (Visual Basic)

```
' Set the data endian of the internal trigger to Big.
ret = WeSetControl(hMo, "Frame:Trigger:Endian", "Big")
Dim value As Variant
ret = WeGetControl(hMo, "Frame:Trigger:Endian", value)
```

value → "Big"

Frame:Trigger:Manual Trigger

Description

Generates manual trigger in frame acquisition mode.

Parameters

None

Example (Visual Basic)

The Variant parameter is not initialized, because there are no parameters.

```
Dim value As Variant
' Generate a manual trigger.
ret = WeSetControl(hMo, "Frame:Trigger:Manual Trigger", "")
```

Output:Mode

Description

Sets the output mode in output mode or queries the current setting.

Parameters

Cont | Trigger | One Shot

Example (Visual Basic)

```
' Set the output mode to Trigger.
ret = WeSetControl(hMo, "Output:Mode", "Trigger")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Mode", value)
```

value → "Trigger"

Output:Interval

Description

Sets the output interval in output mode or queries the current setting.

Parameters

0.001 to 10s

Example (Visual Basic)

```
' Set the output interval to 10 ms.
ret = WeSetControl(hMo, "Output:Interval", "0.01")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Interval", value)
```

value → 0.01

Output:No. of Output Samples

Description
Sets the number of output samples in output mode or queries the current setting.

Parameters
0 to 2147483647

Note:
If you set the value to 0, the number of output samples is set to infinite.

Example (Visual Basic)
```
' Set the number of output samples to infinite.
ret = WeSetControl(hMo, "Output:No. of Output Samples", "0")
Dim value As Variant
ret = WeGetControl(hMo, "Output:No. of Output Samples", value)
```
value → 0

Output:Trigger:Source

Description
Sets the trigger source in output mode or queries the current setting.

Parameters
Internal | BUSTRG

Example (Visual Basic)
```
' Set the trigger source to BUSTRG.
ret = WeSetControl(hMo, "Output:Trigger:Source", "BUSTRG")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:Source", value)
```
value → "BUSTRG"

Output:Trigger:Type

Description
Sets the internal trigger type in output mode or queries the current setting.

Parameters
Off | Rise | Fall | Both | High | Low | X

Example (Visual Basic)
```
' Set the internal trigger type to Rise.
ret = WeSetControl(hMo, "Output:Trigger:Type", "Rise")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:Type", value)
```
value → "Rise"

Output:Trigger:Level

Description
Sets the internal trigger level in output mode or queries the current setting.

Parameters

4-byte value (the selectable range of values vary depending on the data type)

Example (Visual Basic)

```
' Set the internal trigger level to 1000.
ret = WeSetControl(hMo, "Output:Trigger:Level", "1000")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:Level", value)
```

value → 1000

Output:Trigger:Ext

Description

Sets the message format of the internal trigger in output mode or queries the current setting.

Parameters

On | Off

Note:

Specify Off to select the standard format, On to select the extended format.

Example (Visual Basic)

```
' Set the message format of the internal trigger to extended
format.
ret = WeSetControl(hMo, "Output:Trigger:Ext", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:Ext", value)
```

value → "On"

Output:Trigger:ID

Description

Sets the message ID of the internal trigger in output mode or queries the current setting.

Parameters

When using standard format: 0 to 0x7ff
When using extended format: 0 to 0x1fffffff

Example (Visual Basic)

```
' Set the message ID of the internal trigger to 0x123.
ret = WeSetControl(hMo, "Output:Trigger:ID", &h123)
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:ID", value)
```

value → 291 (=&h123)

Output:Trigger:SB

Description

Sets the start bit of the internal trigger in output mode or queries the current setting.

Parameters

0 to 63

Example (Visual Basic)

```
' Set the start bit of the internal trigger to 32.
ret = WeSetControl(hMo, "Output:Trigger:SB", "32")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:SB", "32")
value → 32
```

Output:Trigger:LN

Description

Sets the data length of the internal trigger in output mode or queries the current setting.

Parameters

1 to 64

Example (Visual Basic)

```
' Set the date length of the internal trigger to 32.
ret = WeSetControl(hMo, "Output:Trigger:LN", "32")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:LN", value)
value → 32
```

Output:Trigger:Value Type

Description

Sets the data type of the internal trigger in output mode or queries the current setting.

Parameters

Unsigned | Signed | Float

Example (Visual Basic)

```
' Set the data type of the internal trigger to Signed.
ret = WeSetControl(hMo, "Output:Trigger:Value Type", "Signed")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:Value Type", value)
value → "Signed"
```

Output:Trigger:Endian

Description

Sets the data endian of the internal trigger in output mode or queries the current setting.

Parameters

Big | Little

Example (Visual Basic)

```
' Set the data endian of the internal trigger to Big.
ret = WeSetControl(hMo, "Output:Trigger:Endian", "Big")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Trigger:Endian", value)
value → "Big"
```

Output:Trigger:Manual Trigger

Description
Generates manual trigger in output mode.

Parameters
None

Example (Visual Basic)
The Variant parameter is not initialized, because there are no parameters.

```
Dim value As Variant
' Generate a manual trigger.
ret = WeSetControl(hMo, "Output:Trigger:Manual Trigger", "")
```

Output:Output

Description
Turns ON/OFF the output in output mode or queries the current setting.

Parameters
On | Off

Example (Visual Basic)

```
' Start the output.
ret = WeSetControl(hMo, "Output:Output", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Output:Output", value)
```

value → "On"

Output:CH:Target

Description
Sets the active channel number in output mode or queries the current setting.

Parameters
1 to 64

Note:

The setting and query commands specific to a channel in output mode (Output:CH:xxx) are applied to the channel number specified with this command.

Example (Visual Basic)

```
' Set the active channel number to 15.
ret = WeSetControl(hMo, "Output:CH:Target", "15")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:Target", value)
```

value → 15

Output:CH:On

Description

Turns ON/OFF the output of the active channel in output mode or queries the current setting.

Parameters

On | Off

Example (Visual Basic)

```
' Turn ON the output of CH25.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:On", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:On", value)
```

value → "On"

Output:CH:Ext

Description

Sets the message format of the active channel in output mode or queries the current setting.

Parameters

On | Off

Note:

Specify Off to select the standard format, On to select the extended format.

Example (Visual Basic)

```
' Set the message format of CH25 to extended format.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:Ext", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:Ext", value)
```

value → "On"

Output:CH:ID

Description

Sets the active channel ID in output mode or queries the current setting.

Parameters

When using standard format: 0 to 0x7ff
When using extended format: 0 to 0x1fffffff

Example (Visual Basic)

```
' Set the CH25 ID to 0x123.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:ID", &h123)
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:ID", value)
```

value → 291 (=&h123)

Output:CH:SB

Description
Sets the start bit of the active channel in output mode or queries the current setting.

Parameters
0 to 63

Example (Visual Basic)
```
' Set the start bit of CH25 to 32.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:SB", "32")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:SB", value)
```
value → 32

Output:CH:LN

Description
Sets the data length of the active channel in output mode or queries the current setting.

Parameters
1 to 64

Example (Visual Basic)
```
' Set the data length of CH25 to 16.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:LN", "16")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:LN", value)
```
value → 16

Output:CH:Value Type

Description
Sets the data type of the active channel in output mode or queries the current setting.

Parameters
Unsigned | Signed | Float

Example (Visual Basic)
```
' Set the data type of CH25 to Signed.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:Value Type", "Signed ")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:Value Type", value)
```
value → "Signed"

Output:CH:Endian

Description
Sets the data endian of the active channel in output mode or queries the current setting.

Parameters

Big | Little

Example (Visual Basic)

```
' Set the data endian of CH25 to Big.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:Endian", "Big ")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:Endian", value)
```

value → "Big"

Output:CH:Sweep Pattern

Description

Sets the sweep pattern of the active channel in output mode or queries the current setting.

Parameters

Ramp-Repeat | Ramp-Single&Reset | Ramp-Single&Hold | Triangle-Repeat | Off

Example (Visual Basic)

```
' Set the sweep pattern of CH25 to Triangle-Repeat.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:Sweep Pattern", "Triangle-
Repeat")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:Sweep Pattern", value)
```

value → "Triangle-Repeat"

Output:CH:DataStart

Description

Sets the sweep start value of the active channel in output mode or queries the current setting.

Parameters

The selectable range varies depending on the data type and data length.

Example (Visual Basic)

```
' Set the sweep start value of CH25 to 1000.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:DataStart", "1000")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:DataStart", value)
```

value → 1000

Output:CH:DataEnd

Description

Sets the sweep end value of the active channel in output mode or queries the current setting.

Parameters

The selectable range varies depending on the data type and data length.

Example (Visual Basic)

```
' Set the sweep end value of CH25 to 2000.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:DataEnd", "2000")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:DataEnd", value)
```

value → 2000

Output:CH:Step

Description

Sets the step for changing the sweep value of the active channel in output mode or queries the current setting.

Parameters

N-byte value (the selectable range varies depending on the data type and data length)

Example (Visual Basic)

```
' Set the step for changing the sweep value of CH25 to 50.
ret = WeSetControl(hMo, "Output:CH:Target", "25")
ret = WeSetControl(hMo, "Output:CH:Step", "50")
Dim value As Variant
ret = WeGetControl(hMo, "Output:CH:Step", value)
```

value → 50

Frame Output:Mode

Description

Sets the output mode in frame output mode or queries the current setting.

Parameters

Cont | Trigger

Example (Visual Basic)

```
' Set the output mode to Trigger.
ret = WeSetControl(hMo, "Frame Output:Mode", "Trigger")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Mode", value)
```

value → "Trigger"

Frame Output:No. of Stored Frames

Description

Queries the number of frames that have been downloaded to the memory in frame output mode.

Parameters

0 to 419430

Example (Visual Basic)

```
' Query the number frames that have been downloaded to the memory.
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:No. of Stored Frames", value)
```

value → 40000

Frame Output:Repeat

Description

Sets the number of output repetitions (number of times to accept triggers) in frame output mode or queries the current setting.

Parameters

0 to 100

Note:

If you set the value to 0, the number of output repetitions is set to infinite.

Example (Visual Basic)

```
' Set the number of output repetitions to infinite.
ret = WeSetControl(hMo, "Frame Output:Repeat", "0")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Repeat", value)
```

value → 0

Frame Output:Trigger:Source

Description

Sets the trigger source in frame output mode or queries the current setting.

Parameters

Internal | BUSTRG

Example (Visual Basic)

```
' Set the trigger source to BUSTRG.
ret = WeSetControl(hMo, "Frame Output:Trigger:Source", "BUSTRG")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:Source", value)
```

value → "BUSTRG"

Frame Output:Trigger:Type

Description

Sets the internal trigger type in frame output mode or queries the current setting.

Parameters

Off | Rise | Fall | Both | High | Low | X

Example (Visual Basic)

```
' Set the internal trigger type to Rise.
ret = WeSetControl(hMo, "Frame Output:Trigger:Type", "Rise")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:Type", value)
```

value → "Rise"

Frame Output:Trigger:Level

Description

Sets the internal trigger level in frame output mode or queries the current setting.

Parameters

The selectable range varies depending on the data type and data length.

Example (Visual Basic)

```
' Set the internal trigger level to 1000.
ret = WeSetControl(hMo, "Frame Output:Trigger:Level", "1000")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:Level", value)
```

value → 1000

Frame Output:Trigger:Ext

Description

Sets the message format of the internal trigger in frame output mode or queries the current setting.

Parameters

On | Off

Note:

Specify Off to select the standard format, On to select the extended format.

Example (Visual Basic)

```
' Set the message format of the internal trigger to extended
format.
ret = WeSetControl(hMo, "Frame Output:Trigger:Ext", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:Ext", value)
```

value → "On"

Frame Output:Trigger:ID

Description

Sets the message ID of the internal trigger in frame output mode or queries the current setting.

Parameters

When using standard format: 0 to 0x7ff
When using extended format: 0 to 0x1fffffff

Example (Visual Basic)

```
' Set the message ID of the internal trigger to 0x123.
ret = WeSetControl(hMo, "Frame Output:Trigger:ID", &h123)
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:ID", value)
```

value → 291 (=&h123)

Frame Output:Trigger:SB

Description

Sets the start bit of the internal trigger in frame output mode or queries the current setting.

Parameters

0 to 63

Example (Visual Basic)

```
' Set the start bit of the internal trigger to 32.
ret = WeSetControl(hMo, "Frame Output:Trigger:SB", "32")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:SB", "32")
```

value → 32

Frame Output:Trigger:LN

Description

Sets the data length of the internal trigger in frame output mode or queries the current setting.

Parameters

1 to 64

Example (Visual Basic)

```
' Set the date length of the internal trigger to 32.
ret = WeSetControl(hMo, "Frame Output:Trigger:LN", "32")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:LN", value)
```

value → 32

Frame Output:Trigger:Value Type

Description

Sets the data type of the internal trigger in frame output mode or queries the current setting.

Parameters

Unsigned | Signed | Float

Example (Visual Basic)

```
' Set the data type of the internal trigger to Signed.
ret = WeSetControl(hMo, "Frame Output:Trigger:Value Type",
"Signed")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:Value Type", value)
```

value → "Signed"

Frame Output:Trigger:Endian

Description
Sets the data endian of the internal trigger in frame output mode or queries the current setting.

Parameters
Big | Little

Example (Visual Basic)
```
' Set the data endian of the internal trigger to Big.
ret = WeSetControl(hMo, "Frame Output:Trigger:Endian", "Big")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Trigger:Endian", value)
```
value → "Big"

Frame Output:Trigger:Manual Trigger

Description
Generates manual trigger in frame output mode.

Parameters
None

Example (Visual Basic)
The Variant parameter is not initialized, because there are no parameters.
```
Dim value As Variant
' Generate a manual trigger.
ret = WeSetControl(hMo, "Frame Output:Trigger:Manual Trigger", "")
```

Frame Output:Output

Description
Turns ON/OFF the output in frame output mode or queries the current setting.

Parameters
On | Off

Example (Visual Basic)
```
' Start the output.
ret = WeSetControl(hMo, "Frame Output:Output", "On")
Dim value As Variant
ret = WeGetControl(hMo, "Frame Output:Output", value)
```
value → "On"

Valid Acquisition Modes

Acquisition Type	Valid?	Description
Triggered	Yes	Only when the operation mode is acquisition mode
Free Run	Yes	Only when the operation mode is acquisition mode
Gate(Level)	No	
Gate(Edge)	No	

Acquisition Restrictions (When in Acquisition Mode)

Item	Possible Settings
Memory length	8388608/data size/number of channels whose acquisition is turned ON
Data length per block	During trigger mode: 100 to the memory length/the number of memory partitions During free run mode: 1 to the memory length/the number of memory partitions
Number of blocks	During trigger mode: 1 to 256 During free run mode: 1 to the memory length
Number of acquisitions	Data length per block > memory length/2: 1 Data length per block !s!£ memory length/2: 1 to ∞ (specify 0 to mean ∞)

Valid APIs

API	Valid?	Description
WeStart	Yes	Only when the operation mode is acquisition mode or frame acquisition mode.
WeStop	Yes	Only when the operation mode is acquisition mode or frame acquisition mode.
WeStartSingle	Yes	Only when the operation mode is acquisition mode or frame acquisition mode.
WeStartWithEvent	Yes	Only when the operation mode is acquisition mode or frame acquisition mode.
WeIsRun	Yes	Only when the operation mode is acquisition mode or frame acquisition mode.
WeLatchData	Yes	Only when the operation mode is acquisition mode
WeGetAcqDataInfo	Yes	Only when the operation mode is acquisition mode
WeGetAcqData	Yes	Only when the operation mode is acquisition mode (the retrieved data must be bit-extracted using the WeTransAcqData function)
WeGetScaleData	Yes	Only when the operation mode is acquisition mode
WeGetMeasureParam	Yes	Only when the operation mode is acquisition mode
WeSaveAcqData	Yes	Only when the operation mode is acquisition mode
WeSaveScaleData	Yes	Only when the operation mode is acquisition mode
WeSaveAsciiData	Yes	Only when the operation mode is acquisition mode
WeGetCurrentData	Yes	Only when the operation mode is acquisition mode
WeGetAcqDataEx	Yes	Only when the operation mode is acquisition mode (the retrieved data must be bit-extracted using the WeTransAcqData function)
WeGetAcqDataSize	Yes	Only when the operation mode is acquisition mode
WeSaveAcqHeader	Yes	Only when the operation mode is acquisition mode
WeSavePatternData	Yes	Only when the operation mode is frame acquisition mode (used to save the frame data)
WeLoadPatternData	Yes	Only when the operation mode is frame output mode (used to load the frame data)
WeLoadPatternDataEx	No	
WeStartEx	Yes	Only when the operation mode is acquisition mode
WeStopEx	Yes	Only when the operation mode is acquisition mode

Valid Common Events

Event	Valid?	Description
WE_EV_MEASSTART	Yes	Only when the operation mode is acquisition mode or frame acquisition mode.
WE_EV_MEASEND	Yes	Only when the operation mode is acquisition mode or frame acquisition mode.
WE_EV_BLOCKEND	Yes	Only when the operation mode is acquisition mode
WE_EV_MEASABORT	Yes	Only when the operation mode is acquisition mode or frame acquisition mode.
WE_EV_TRIG_HIGH	Yes	Only when the operation mode is acquisition mode
WE_EV_TRIG_LOW	Yes	Only when the operation mode is acquisition mode
WE_EV_TRIG_PULSE	Yes	Only when the operation mode is acquisition mode
WE_EV_CLOSE_MODULE_GUI	Yes	

Module-specific Events

Event	Event Description
0x01000000(&h01000000)	Overrun error occurred
0x02000000(&h02000000)	Arbitration lost error occurred
0x04000000(&h04000000)	Bus error occurred
0x08000000(&h08000000)	Bus off occurred

Module-specific Error Codes

None.

8.5 WE7111 100 MS/s Digital Oscilloscpe

ASCII Commands

Acquisition controls

ASCII Command	Description
Misc:Acq Mode	Sets the acquisition mode or queries the current setting.
Misc:Average Count	Sets the average count (attenuation constant) or queries the current setting.
Time/div	Sets the Time/div setting or queries the current setting.
Filter:20MHz	Sets the input filter or queries the current setting.
Misc:Time Base	Sets the time base source or queries the current setting.
Misc:Record Length	Sets the record length or queries the current setting.

Trace controls

ASCII Command	Description
CH<x>:V/div	Sets the V/div setting or queries the current setting.
CH<x>:Coupling	Sets input coupling or queries the current setting.
CH<x>:Probe	Sets probe attenuation setting or queries the current setting.
CH<x>:Offset	Sets offset voltage or queries the current setting.
CH<x>:Measure	Sets whether or not to perform automated measurement of parameters, or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>.

If the modules in slot 1 and 2 are linked and you wish to change the offset voltage of the module in slot 2, specify "CH2:Offset" in the command line.

Trigger controls

ASCII Command	Description
TriggerMode	Sets the trigger mode or queries the current setting.
Trig:Coupling	Sets the trigger coupling or queries the current setting.
Trig:HF Reject	Sets the HF rejection of the trigger signal or queries the current setting.
Trig:Delay	Sets the trigger delay or queries the current setting.
Trig:Source	Sets the trigger source or queries the current setting.
Trig:Slope	Sets the trigger slope or queries the current setting.
Trig:Level	Sets the trigger level or queries the current setting.
Trig:Position	Sets the trigger position or queries the current setting.
Trig:Hold Off	Sets the trigger hold off mode or queries the current setting.
Trig:Hold Off Time	Sets the trigger hold off time or queries the current setting.

Other controls

ASCII Command	Description
Misc:Auto Setup:Exec	Executes auto setup.
Misc:Calibration:Cal Exec	Executes calibration.
Misc:Calibration:Auto Cal	Sets the automatic calibration setting or queries the current setting.

Misc:Acq Mode

Description

Sets the acquisition mode or queries the current setting.

Parameter

Normal | Envelope | Average

Note:

Envelope can be set only when the time base is set to Internal. In addition, certain combinations of Time/div and record length do not allow the envelope mode. For details, see the User's Manual for the module.

Example (Visual Basic)

```
' Set the acquisition mode to Normal.
ret = WeSetControl (hMo, "Misc:Acq Mode", "Normal")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Acq Mode", value)
```

value → Normal

Misc:Average Count

Description

Sets the average count (attenuation constant) or queries the current setting.

Parameter

2 to 256

Note:

Valid only when the acquisition mode is set to Average.

Example (Visual Basic)

```
' Set the acquisition count to 32.
ret = WeSetControl (hMo, "Misc:Average Count", "32")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Average Count", value)
```

value → 32.0E+00

Time/div

Description

Sets the Time/div setting or queries the current setting.

Parameter

100 ns to 50 s

Note:

Valid only when the time base is set to internal.

Example (Visual Basic)

```
' Set the Time/Div to 100 µs/div.
ret = WeSetControl (hMo, "Time/div", "10E-6")
Dim value As Variant
ret = WeGetControl (hMo, "Time/div", value)
```

value → 10.0E-06

Filter:20MHz

Description

Sets the input filter or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Turn On the input filter.
ret = WeSetControl (hMo, "Filter:20 MHz", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Filter:20 MHz", value)
```

value → On

Misc:Time Base

Description

Sets the time base source or queries the current setting.

Parameter

Internal | EXT CLOCK IN | BUSCLK

Example (Visual Basic)

```
'  Set the time base source to internal.
ret = WeSetControl (hMo, "Misc:Time Base", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Time Base", value)
```

value → Internal

Misc:Record Length

Description

Sets the record length or queries the current setting.

Parameter

1 k to 100 k

Example (Visual Basic)

```
'  Set the record length to 1 kword.
ret = WeSetControl (hMo, "Misc:Record Length", "1k")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Record Length", value)
```

value → 1.0000E+03

CH<x>:V/div

Description

Sets the V/div setting or queries the current setting.

Parameter

5 mV to 5 V (when the probe attenuation is set to 1:1)

Example (Visual Basic)

```
' Set V/div to 100 mV/div.
ret = WeSetControl (hMo, "CH1:V/div", "100E-3")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:V/div", value)
```

value → 100.0E–03

CH<x>:Coupling

Description

Sets input coupling or queries the current setting.

Parameter

AC | DC | GND

Example (Visual Basic)

```
' Set the input coupling to DC.
ret = WeSetControl (hMo, "CH1:Coupling", "DC")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Coupling", value)
```

value → DC

CH<x>:Probe

Description

Sets probe attenuation setting or queries the current setting.

Parameter

1:1 | 10:1 | 100:1 | 1000:1

Example (Visual Basic)

```
' Set the probe attenuation to 10:1.
ret = WeSetControl (hMo, "CH1:Probe", "10:1")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Probe", value)
```

value → 10:1

CH<x>:Offset

Description

Sets offset voltage or queries the current setting.

Parameter

When probe attenuation is set to 1:1

5 mV/div to 50 mV/div	±1 V
100 mV/div to 500 mV/div	±10 V
1 V/div to 5 V/div	±100 V

Note:

Valid only when the input coupling is set to DC. The resolution varies depending on the selected V/div (voltage sensitivity) setting. For details, see the User's Manual for the module.

Example (Visual Basic)

```
' Set the offset to 1.2 V.
ret = WeSetControl (hMo, "CH1:Offset", "1.2")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Offset", value)
```

value → 1.200E+00

CH<x>:Measure

Description

Sets whether or not to perform automated measurement of waveform parameters, or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable the automated measurement.
ret = WeSetControl (hMo, "CH1:Measure", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Measure", value)
```

value → On

Trigger Mode

Description

Sets the trigger mode or queries the current setting.

Parameter

Auto | Auto Level | Normal

Example (Visual Basic)

```
' Set the trigger mode to Normal.
ret = WeSetControl (hMo, "Trigger Mode", "Normal")
Dim value As Variant
ret = WeGetControl (hMo, "Trigger Mode", value)
```

value → Normal

Trig:Coupling

Description

Sets the trigger coupling or queries the current setting.

Parameter

AC | DC

Note:

Cannot be specified when the trigger source is Line or BUSTRG.

Example (Visual Basic)

```
' Set the trigger coupling to DC.
ret = WeSetControl (hMo, "Trig:Coupling", "DC")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Coupling", value)
```

value → DC

Trig:HF Reject

Description

Sets the HF rejection of the trigger signal or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Turn On the HF Rejection of the trigger signal.
ret = WeSetControl (hMo, "Trig:HF Reject", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:HF Reject", value)
```

value → On

Trig:Delay

Description

Sets the trigger delay or queries the current setting.

Parameter

9.99999 s to 200 ns

Example (Visual Basic)

```
' Set the trigger delay to 100 ms.
ret = WeSetControl (hMo, "Trig:Delay", "100E-3")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Delay", value)
```

value → 100.00000E–03

Trig:Source

Description

Sets the trigger source or queries the current setting.

Parameter

Line | BUSTRG | CH1 | CH2 | CH3 | CH4 | CH5 | CH6 | CH7 | CH8

Example (Visual Basic)

```
' Set the trigger source to CH1.
ret = WeSetControl (hMo, "Trig:Source", "CH1")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Source", value)
```

value → CH1

Trig:Slope

Description
Sets the trigger slope or queries the current setting.

Parameter
Rise | Fall | Both

Note:
Cannot be specified when the trigger source is Line or BUSTRG.

Example (Visual Basic)
```
' Set the trigger slope to Fall.
ret = WeSetControl (hMo, "Trig:Slope", "Fall")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Slope", value)
```
value → Fall

Trig:Level

Description
Sets the trigger level or queries the current setting.

Parameter
Voltage corresponding to ±10 div on the vertical scale.

Note:
Cannot be specified when the trigger source is Line or BUSTRG.

Example (Visual Basic)
```
' Set the trigger level to 1 V.
ret = WeSetControl (hMo, "Trig:Level", "1")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Level", value)
```
value → 1.000E+00

Trig:Position

Description
Sets the trigger position or queries the current setting.

Parameter
±5 div

Example (Visual Basic)
```
' Set the trigger position to -2 div.
ret = WeSetControl (hMo, "Trig:Position", "-2")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Position", value)
```
value → –2.0E+00

Trig:Hold Off

Description
Sets the trigger hold off mode or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Enable the trigger hold off mode.
ret = WeSetControl (hMo, "Trig:Hold Off", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Hold Off", value)
```
value → On

Trig:Hold Off Time

Description
Sets the trigger hold off time or queries the current setting.

Parameter
9.999999 s to 200 ns

Note:
Valid only when the hold off mode is On.

Example (Visual Basic)
```
' Set the trigger hold off time to 2 s.
ret = WeSetControl (hMo, "Trig:Hold Off Time", "2")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Hold Off Time", value)
```
value → 2.00000000E+00

Misc:Auto Setup:Exec

Description
Executes auto setup.

Parameter
None

Example (Visual Basic)
```
' The Variant parameter is not initialized, because there are no
' parameters.
Dim value As Variant
' Execute auto setup.
ret = WeSetControl (hMo, "Misc:Auto Setup:Exec", value)
```

Misc:Calibration:Cal Exec

Description
Executes a full calibration.

Parameter

None

Example (Visual Basic)

```
' The Variant parameter is not initialized, because there are no
' parameters.
Dim value As Variant
' Execute calibration.
ret = WeSetControl (hMo, "Misc:Calibration:Cal Exec", value)
```

Misc:Calibration:Auto Cal

Description

Sets the automatic calibration setting or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable automatic calibration.
ret = WeSetControl (hMo, "Misc:Calibration:Auto Cal", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Calibration:Auto Cal", value)
```

value → On

Valid Acquisition Modes

Event	Valid?
Triggered	Yes
Free Run	No
Gate(Level)	No
Gate(Edge)	No

Acquisition Restrictions

Items	Possible Settings	Note
Record length	120k	
Data length per block	1k, 5k, 10k, 30k, 120k	120 kword cannot be specified during Average mode.
Number of block	1	
Number of Acquisition	1	

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	No	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	The A/D data is 16-bit signed integer during the Average mode. For all other modes, the data is 8-bit unsigned integer.
WeGetScaleData	Yes	
WeGetMeasureParam	Yes	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	No	
WeGetAcqDataEx	Yes	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeLoadPatternDataEx	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Event Description
WWE_EV_MEASEND	No
WE_EV_BLOCKEND	No
WE_EV_MEASABORT	No
WE_EV_TRIG_HIGH	Yes
WE_EV_TRIG_LOW	Yes
WE_EV_TRIG_PULSE	Yes
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	No

Module-specific Events

None

Module-specific Error Codes

Error Code	Description
0×1801	The external clock freqency is below 40 Hz.
0×1802	The external clock freqency is above 15 MHz.

8.6 WE7116 2-CH, 20MS/s Digitizer Module

ASCII Commands

ASCII Command	Description
Trigger Mode	Sets the trigger mode or queries the current setting.
Sampling Interval	Sets the sampling interval or queries the current setting.
Record Length	Sets the record length or queries the current setting.
Memory Partition	Sets the memory partitions or queries the current setting.
No. of Acquisitions	Sets the number of acquisitions or queries the current setting.

CH<X>:

ASCII Command	Description
CH<X>:On	Sets the measurement channel or queries the current setting.
CH<X>:Coupling	Sets the input coupling of the channel or queries the current setting.
CH<X>:Range	Sets the range of the channel or queries the current setting.
CH<X>:Offset	Sets the offset voltage of the channel or queries the current setting.
CH<X>:Filter	Sets the low-pass filter of the channel or queries the current setting.
CH<X>:Probe	Sets the probe attenuation of the channel or queries the current setting.

Trig:

ASCII Command	Description
Trig:Source	Sets the trigger source or queries the current setting.
Trig:Type	Sets the trigger type or queries the current setting.
Trig:Level	Sets the trigger level or queries the current setting.
Trig:Hysteresis	Sets the hysteresis width or queries the current setting.
Trig:Upper Level	Sets the upper-limit trigger level for window trigger or queries the current setting.
Trig:Lower Level	Sets the lower-limit trigger level for window trigger or queries the current setting.
Trig:PreTrigger	Sets the amount of pretrigger or queries the current setting.
Trig:HoldOff	Sets the amount of hold off or queries the current setting.

Misc:

ASCII Command	Description
Misc:TimeBase	Sets the time base or queries the current setting.
Misc:offset CAL	Executes DC offset calibration.
Misc:offset CAL:Status	Quries the execution status of DC offset calibration.

Note

Enter the channel number in <x>. If the modules are linked specify a serial number from the parent module.

There are two channels on each WE7116 module.

If two modules are linked and you wish to change the range of channel 2 of the second module, specify "CH4:Range" in the command line.

Trigger Mode

Description
Sets the trigger mode or queries the current setting.

Parameter
Auto | Normal

Example (Visual Basic)
```
' Set the trigger mode to Normal.
ret = WeSetControl (hMo, "Trigger Mode", "Normal")
Dim value As Variant
ret = WeGetControl (hMo, "Trigger Mode", value)
value →Normal
```

Sampling Interval

Description
Sets the sampling interval or queries the current setting.

Parameter
0.00000005 to 0.001

Example (Visual Basic)
```
' Set the sampling interval to 10 usec.
ret = WeSetControl (hMo, "Sampling Interval", "10E-6")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
value →10.000E-06
```

Record Length

Description
Sets the record length or queries the current setting.

Parameter
2 to 4194304/the number of memory partitions

Example (Visual Basic)
```
' Set the record length to 1000.
ret = WeSetControl (hMo, "Record Length", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "Record Length", value)
value →1000
```

Memory Partition

Description
Sets the number of memory partitions or queries the current setting.

Parameter
1 to 1024 (2^n step values)

Example (Visual Basic)

```
' Set the number of memory partitions to 2.
ret = WeSetControl (hMo, "Memory Partition", "2")
Dim value As Variant
ret = WeGetControl (hMo, "Memory Partition", value)
value → 2.0E+00
```

No. of Acquisitions

Description

Sets the number of acquisitions or queries the current setting.

Parameter

1 to the number of memory partitions

Example (Visual Basic)

```
' Set the number of acquisitions to 100.
ret = WeSetControl (hMo, "No. of Acquisitions", "100")
Dim value As Variant
ret = WeGetControl (hMo, "No. of Acquisitions", value)
value → 100
```

CH<X>:On

Description

Turns On/Off the measurement channel or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable the CH1 input signal.
ret = WeSetControl (hMo, "CH1:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:On", value)
value → On
```

CH<X>:Coupling

Description

Sets the input coupling of the channel or queries the current setting.

Parameter

DC | AC | GND

Example (Visual Basic)

```
' Set the coupling of CH1 to DC.
ret = WeSetControl (hMo, "CH1:Coupling", "DC")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Coupling", value)
value → DC
```

CH<X>:Range

Description

Sets the measurement range of the channel or queries the current setting.

Parameter

The possible selections vary depending on the selected probe attenuation.
When the probe attenuation is set to 1:1
100 mV | 200 mV | 500 mV | 1 V | 2 V | 5 V | 10 V | 20 V | 50 V
When the probe attenuation is set to 10:1
1 V | 2 V | 5 V | 10 V | 20 V | 50 V | 100 V | 200 V | 500 V
When the probe attenuation is set to 100:1
10 V | 20 V | 50 V | 100 V | 200 V | 500 V | 1 kV | 2 kV | 5 kV
When the probe attenuation is set to 1000:1
100 V | 200 V | 500 V | 1 kV | 2 kV | 5 kV | 10 kV | 20 kV | 50 kV

Example (Visual Basic)

```
' Set the measurement range of CH1 to 1 V.
ret = WeSetControl (hMo, "CH1:Range", "1V")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Range", value)
value →1 V
```

CH<X>:Offset

Description

Sets the offset voltage of the channel or queries the current setting.

Parameter

The selectable range varies depending on the measurement range and the probe attenuation.

Measurement Range	Probe Attenuation			
	1:1	10:1	100:1	1000:1
100 mV	±0.2 V (0.0001 V steps)	±2.0 V (0.001 V steps)	±20.0 V (0.01 V steps)	±200.0 V (0.1 V steps)
200 mV	±0.4 V (0.0002 V steps)	±4.0 V (0.002 V steps)	±40.0 V (0.02 V steps)	±400.0 V (0.2 V steps)
500 mV	±1.0 V (0.0005 V steps)	±10.0 V (0.005 V steps)	±100.0 V (0.05 V steps)	±1000.0 V (0.5 V steps)
1 V	±2.0 V (0.001 V steps)	±20.0 V (0.01 V steps)	±200.0 V (0.1 V steps)	±2 kV (1 V steps)
2 V	±4.0 V (0.002 V steps)	±40.0 V (0.02 V steps)	±400.0 V (0.2 V steps)	±4 kV (2 V steps)
5 V	±10.0 V (0.005 V steps)	±100.0 V (0.05 V steps)	±1000.0 V (0.5 V steps)	±10 kV (5 V steps)
10 V	±20.0 V (0.01 V steps)	±200.0 V (0.1 V steps)	± 2kV (1 V steps)	±20 kV (10 V steps)
20 V	±40.0 V (0.02 V steps)	±400.0 V (0.2 V steps)	±4 kV (2 V steps)	±40 kV (20 V steps)
50 V	±100.0 V (0.05 V steps)	±1000.0 V (0.5 V steps)	±10 kV (5 V steps)	±100 kV (50 V steps)

Example (Visual Basic)

```
' Set the offset voltage of CH1 to 1.0 V.
ret = WeSetControl (hMo, "CH1:Offset", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Offset", value)
value → 1.0E+00
```

CH<X>:Filter

Description

Sets the low-pass filter of the channel or queries the current setting.

Parameter

OFF | 500 kHz | 1 MHz

Example (Visual Basic)

```
' Set the low-pass filter of CH1 to 1 MHz.
ret = WeSetControl (hMo, "CH1:Filter", "1MHz")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Filter", value)
value → 1.0000E+06
```

CH<X>:Probe

Description

Sets the probe attenuation of the channel or queries the current setting.

Parameter

1:1 | 10:1 | 100:1 | 1000:1

Note:

The measurement range and offset voltage values change when you change the probe attenuation. There is no effect if the probe attenuation is 1:1. However, if the probe attenuation is 10:1, 100:1, or 1000:1, the values are increased 10 times, 100 time, or 1000 times, respectively.

Example (Visual Basic)

```
' Set the probe attenuation of CH1 to 10:1.
ret = WeSetControl (hMo, "CH1:Probe", "10:1")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Probe", value)
value → 10:1
```

Trig:Source

Description

Sets the trigger source or queries the current setting.

Parameter

EXTTRG | BUSTRG | Line | CH1 | CH2 | CH3 | CH4 | CH5 | CH6 | CH7 | CH8 | CH9 | CH10 | CH11 | CH12 | CH13 | CH14 | CH15 | CH16

Example (Visual Basic)

```
' Set the trigger source to internal trigger, and the source
channel to CH1.
ret = WeSetControl (hMo, "Trig:Source", "CH1")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Source", value)
value →CH1
```

Trig:Type

Description

Sets the trigger type or queries the current setting.

Parameter

Rise | Fall | Both | Enter | Exit

Example (Visual Basic)

```
' Set the trigger type to Rise.
ret = WeSetControl (hMo, "Trig:Type", "Rise")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Type", value)
value →Rise
```

Trig:Level

Description

Sets the trigger level when the trigger type is set to Rise, Fall, or Both.

Parameter

The selectable range varies depending on the measurement range and DC offset of the channel set to be the trigger source and the probe attenuation as follows:
Selectable range: ±measurement range × 0.9 + DC offset (1% of measurement range steps)

(Example 1) For measurement range 1 V, DC offset 0 V, and probe attenuation 1:1
Selectable range: -0.9 V to 0.9 V (0.01 V steps)

(Example 2) For measurement range 1 V, DC offset 1 V, and probe attenuation 1:1
Selectable range: 0.1 V to 1.9 V (0.01 V steps)

(Example 3) For measurement range 10 V, DC offset 1 V, and probe attenuation 10:1
Selectable range: -8.0 V to 10.0 V (0.1 V steps)

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Level", "1.0")
' Set the trigger level to 1 V.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Level", value)
value →1.0E+00
```

Trig:Hysteresis

Description

Sets the hysteresis width or queries the current setting.

Parameter

3% | 10%

Example (Visual Basic)

```
' Set the hysteresis width to 10%.
ret = WeSetControl (hMo, "Trig:Hysteresis", "10%")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Hysteresis", value)
value → 10%
```

Trig:Upper Level

Description

Sets the upper-limit trigger level when the trigger type is set to Enter or Exit (window trigger).

Parameter

The selectable range varies depending on the measurement range and DC offset of the channel set to be the trigger source and the probe attenuation as follows:
Selectable range: lower-limit trigger level ± measurement range × 0.1 to measurement range × 0.9 + DC offset (1% of measurement range steps)

Example (Visual Basic)

```
' Set the trigger level to 1 V.
ret = WeSetControl (hMo, "TrigUpper Level", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Upper Level", value)
value → 1.0E+00
```

Trig:Lower Level

Description

Sets the lower-limit trigger level when the trigger type is set to Enter or Exit (window trigger).

Parameter

The selectable range varies depending on the measurement range and DC offset of the channel set to be the trigger source and the probe attenuation as follows:
Selectable range: Measurement range × -0.9 + DC offset to upper-limit trigger level + measurement range × 0.1 (1% of measurement range steps)

Example (Visual Basic)

```
' Set the trigger level to -1 V.
ret = WeSetControl (hMo, "TrigLower Level", "-1.0")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Lower Level", value)
value → -1.0E+00
```

Trig:PreTrigger

Description

Sets the amount of pretrigger or queries the current setting.

Parameter

0 to record length -2

Example (Visual Basic)

```
' Set the amount of pretrigger to 1000.
ret = WeSetControl (hMo, "Trig:PreTrigger", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:PreTrigger", value)
value →1000
```

Trig:Hold Off

Description

Sets the amount of hold off or queries the current setting.

Parameter

When in triggered mode Record length to 4194304

Example (Visual Basic)

```
' Set the hold off time to 1000.
ret = WeSetControl (hMo, "Trig:Hold Off", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Hold Off", value)
value →1000
```

Misc:TimeBase

Description

Sets the time base or queries the current setting.

Parameter

Internal | EXTCLK | BUSCLK

Example (Visual Basic)

```
' Set the time base to internal clock.
ret = WeSetControl (hMo, "Misc:TimeBase", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:TimeBase", value)
value →Internal
```

Misc:offset CAL

Description

Executes DC offset calibration.

Parameter

None

Example (Visual Basic)

```
' The Variant parameter is not initialized, because there are no
parameters.
Dim value As Variant
' Execute DC offset calibration.
ret = WeGetControl (hMo, "Misc:offset CAL", value)
```

Misc:offset CAL:Status

Description

Queries the execution status of DC offset calibration.

Parameter

0 (End state) | 1 (Under offset calibration execution)

Example (Visual Basic)

```
' Query the execution status of DC offset calibration.
Dim value As Variant
ret = WeGetControl (hMo, "Misc:offset CAL:Status", value)
value → 0
```

Valid Acquisition Modes

Acquisition Type	Valid?	Description
Triggered	Yes	
Free Run	No	
Gate(Level)	No	
Gate(Edge)	No	

Acquisition Restrictions

	Description
Memory length	4194304
Data length per block	2 to the memory length/the number of blocks * For the record length range, see the description of the Record Length command.
Number of blocks	1 to 1024
Number of acquisitions	1 to the number of blocks

Valid Common Measurement Control API

API	Valid?	Description
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	No	No supported since free run mode does not exist.
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is raw data (A/D data) in 12-bit signed integer format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is a physical value in double-precision real format.
WeGetAcqDataEx	Yes	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?	Description
WE_EV_MEASEND	Yes	
WE_EV_BLOCKEND	No	
WE_EV_MEASABORT	Yes	
WE_EV_TRIG_HIGH	No	
WE_EV_TRIG_LOW	No	
WE_EV_TRIG_PULSE	No	
WE_EV_CLOSE_MODULE_GUI	Yes	

Module-specific Events

None

Module-specific Error Codes

None

8.7 WE7121 10 MHz Function Generator

ASCII Commands

ASCII Command	Description
Link:Freq	Sets the channel link for the frequency parameter or queries the current setting.
Link:Phase	Sets the channel link for the phase parameter or queries the current setting.
Link:Ampl	Sets the channel link for the amplitude parameter or queries the current setting.
Link:Offset	Sets the channel link for the offset parameter or queries the current setting.
Link:Duty	Sets the channel link for the duty cycle parameter or queries the current setting.
Link:Output	Sets the channel link for the waveform output or queries the current setting.
Phase Sync	Performs phase synchronization.
Manual Trigger	Executes a manual trigger.

CH<x>:

ASCII Command	Description
CH<x>:Invert	Sets the waveform inversion or queries the current setting.
CH<x>:Mode	Sets the output mode or queries the current setting.
CH<x>:Trigger	Sets the trigger source or queries the current setting.
CH<x>:Function	Sets the output waveform (function) or queries the current setting.
CH<x>:Output	Turns ON/OFF the waveform output or queries the current setting.
CH<x>:Freq	Sets the frequency or queries the current setting.
CH<x>:Phase	Sets the phase or queries the current setting.
CH<x>:Ampl	Sets the amplitude or queries the current setting.
CH<x>:Offset	Sets the offset voltage or queries the current setting.
CH<x>:Duty	Sets the duty cycle or queries the current setting.
CH<x>:Burst	Sets the burst count or queries the current setting.
CH<x>:Trigger Freq	Sets the trigger frequency or queries the current setting.
CH<x>:PatternData	Sets the arbitrary waveform data or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>.
If the modules in slot 1 and 2 are linked and you wish to change the offset voltage of the module in slot 2, specify "CH2:Offset" in the command line.

Link:Freq

Description

Sets the channel link for the frequency parameter or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Enable the channel link for the frequency parameter.
ret = WeSetControl (hMo, "Link:Freq", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Link:Freq", value)
```

value → On

Link:Phase

Description

Sets the channel link for the phase parameter or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Enable the channel link for the phase parameter.
ret = WeSetControl (hMo, "Link:Phase", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Link:Phase", value)
```

value → On

Link:Ampl

Description

Sets the channel link for the amplitude parameter or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Enable the channel link for the amplitude parameter.
ret = WeSetControl (hMo, "Link:Ampl", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Link:Ampl", value)
```

value → On

Link:Offset

Description

Sets the channel link for the offset parameter or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Enable the channel link for the offset parameter.
ret = WeSetControl (hMo, "Link:Offset", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Link:Offset", value)
```

value → On

Link:Duty

Description

Sets the channel link for the duty cycle parameter or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Enable the channel link for the duty cycle parameter.
ret = WeSetControl (hMo, "Link:Duty", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Link:Duty", value)
```

value → On

Link:Output

Description

Sets the channel link for the waveform output or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Enable the channel link for the waveform output.
ret = WeSetControl (hMo, "Link:Output", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Link:output", value)
```

value → On

Phase Sync

Description

Performs phase synchronization.

Parameter

None

Example (Visual Basic)

```
' The Variant parameter is not initialized, because there are no
' parameters.
Dim value As Variant
' Perform phase synchronization.
ret = WeSetControl (hMo, "Phase Sync", val)
```

Manual Trigger

Description

Executes manual trigger.

Parameter

On | Off

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Manual Trigger", "Off")
```

CH<x>:Invert

Description

Sets the waveform inversion or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Invert the waveform output.
ret = WeSetControl (hMo, "CH1:Invert", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Invert", value)
```

value → On

CH<x>:Mode

Description

Sets the output mode or queries the current setting.

Parameter

Cont | Trigger | Gate | DC

Example (Visual Basic)

```
' Set the output mode to continuous oscillation.
ret = WeSetControl (hMo, "CH1:Mode", "Cont")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Mode", value)
```

value → Cont

CH<x>:Trigger

Description

Sets the trigger source or queries the current setting.

Parameter

Internal | BUSTRG

Example (Visual Basic)

```
' Set the trigger mode to internal.
ret = WeSetControl (hMo, "CH1:Trigger", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trigger", value)
```

value → Internal

CH<x>:Function

Description

Sets the output waveform (function) or queries the current setting.

Parameter

Sine | Square | Ramp | Triangle | Pulse | Arbitrary

Example (Visual Basic)

```
' Set the output waveform to pulse.
ret = WeSetControl (hMo, "CH1:Function", "Pulse")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Function", value)
```
value → Pulse

CH<x>:Output

Description

Turns ON/OFF the waveform output or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Output the waveform.
ret = WeSetControl (hMo, "CH1:Output", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1: Output", value)
```
value → On

CH<x>:Freq

Description

Sets the frequency or queries the current setting.

Parameter

1 u(μ)Hz to 10 MHz

Example (Visual Basic)

```
' Set the frequency to 1 MHz.
ret = WeSetControl (hMo, "CH1:Freq", "1E6")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Freq", value)
```
value → 1.000000000000E+06

CH<x>:Phase

Description

Sets the phase or queries the current setting.

Parameter

–10000 to 10000 degrees

Example (Visual Basic)

```
' Set the phase to 1224 degrees.
ret = WeSetControl (hMo, "CH1:Phase", "1224")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Phase", value)
```
value → 1.22400E+03

CH<x>:Ampl

Description
Sets the amplitude or queries the current setting.

Parameter
0.02 to 20 V

Example (Visual Basic)
```
' Set the amplitude to 5.123 V.
ret = WeSetControl (hMo, "CH1:Ampl", "5.123")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Ampl", value)
```
value → 5.123E+00

CH<x>:Offset

Description
Sets the offset voltage or queries the current setting.

Parameter
0 to ±10 V

Example (Visual Basic)
```
' Set the offset to 1.45 V.
ret = WeSetControl (hMo, "CH1:Offset", "1.45")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Offset", value)
```
value → 1.450E+00

CH<x>:Duty

Description
Sets the duty cycle or queries the current setting.

Parameter
0 to 100%

Example (Visual Basic)
```
' Set the duty cycle to 98%.
ret = WeSetControl (hMo, "CH1:Duty", "98")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Duty", value)
```
value → 98.00E+00

CH<x>:Burst

Description
Sets the burst count or queries the current setting.

Parameter
1 to 65535

Example (Visual Basic)

```
' Set the burst count to 1000 times.
ret = WeSetControl (hMo, "CH1:Burst", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Burst", value)
```

value → 1000

CH<x>:Trigger Freq

Description

Sets the trigger frequency or queries the current setting.

Parameter

0.001 Hz to 50 kHz

Example (Visual Basic)

```
' Set the trigger frequency to 100 Hz.
ret = WeSetControl (hMo, "CH1:Trigger Freq", "100")
Dim value As Variant
ret = WeGetControl (hMo, "CH1: Trigger Freq", value)
```

value → 100.000E+00

CH<x>:PatternData

Description

Sets the arbitrary waveform data or queries the current setting.

Parameter

WE_UWORD*16384

For details, see the User's Manual for the module.

Example (Visual Basic)

```
' Allocate the buffer memory in PC.
Dim waveBuf(16384) As Integer
' Open the arbitrary waveform data file.
Open "a:\arb.w16" For Binary As #1
' Transfer the arbitrary waveform data to the allocated buffer.
Get #1, , waveBuf
' Send the arbitrary waveform data to channel 1.
ret = WeSetControl(fgHandle, "CH1:PatternData", waveBuf)
' Read the data that was sent.
ret = WeGetControl(fgHandle, "CH1:PatterData", waveBuf)
```

Valid Acquisition Modes

This module is not a data acquisition module.

Acquisition Restrictions

This module is not a data acquisition module.

Valid Common Measurement Control API

API	Valid?	Note
WeStart	No	
WeStop	No	
WeStartSingle	No	
WeStartWithEvent	No	
WeIsRun	No	
WeLatchData	No	
WeGetAcqDataInfo	No	
WeGetAcqData	No	
WeGetScaleData	No	
WeGetMeasureParam	No	
WeSaveAcqData	No	
WeSaveScaleData	No	
WeSaveAsciiData	No	
WeGetCurrentData	No	
WeGetAcqDataEx	No	
WeGetAcqDataSize	No	
WeSaveAcqHeader	No	
WeSavePatternData	No	
WeLoadPatternData	Yes	The file extension is w16.
WeLoadPatternDataEx	No	
WeStartEx	No	
WeStopEx	No	

Valid Common Events

Event	Event Description
WWE_EV_MEASEND	No
WE_EV_BLOCKEND	No
WE_EV_MEASABORT	No
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

None

8.8 WE7131 2 MHz Pattern I/O

ASCII Commands

ASCII Command	Description
Frequency	Sets the clock frequency or queries the current setting.
Mode	Sets operation mode or queries the current setting.
Memory Length	Sets memory length or queries the current setting.
Pre Trigger:Cycle	Sets the pretrigger (cycle) or queries the current setting.
Pre Trigger:Percent	Sets the pretrigger (%) or queries the current setting.
Misc:Time Base	Sets the time base or queries the current setting.
Misc:Output Clock	Sets the clock output ON/OFF or queries the current setting.
Misc:Arming	Sets the arming setting or queries the current setting.

CH<x>:

ASCII Command	Description
CH<x>:Trigger:Mask	Sets the trigger mask or queries the current setting.
CH<x>:Trigger:Pattern	Sets the trigger pattern or queries the current setting.
CH<x>:Trigger:Point	queries the trigger point.
CH<x>:Trigger:Disable	Sets whether or not to enable triggering, or queries the current setting.
CH<x>:Memory:I/O Select	Sets the input/output bits or queries the current setting.
CH<x>:Memory:Output	Sets the output pattern or queries the current setting.
CH<x>:Memory:Input	queries the input pattern.

Note

Enter the channel number in the <x> of CH<x>.
If the modules in slot 1 and 2 are linked and you wish to change the trigger pattern of the module in slot 2, specify "CH2:Trigger:Pattern" in the command line.

Frequency

Description

Sets the clock frequency or queries the current setting.

Parameter

5 mHz to 2 MHz

Example (Visual Basic)

```
' Set the clock frequency to 2 MHz.
ret = WeSetControl (hMo, "Frequency", "2E6")
Dim value As Variant
ret = WeGetControl (hMo, "Frequency", value)
```

value → 2.0000000E+06

Mode

Description

Sets operation mode or queries the current setting.

Parameter

One Shot | Repeat

Example (Visual Basic)

```
' Set the operation mode to Repeat.
ret = WeSetControl (hMo, "Mode", "Repeat")
Dim value As Variant
ret = WeGetControl (hMo, "Mode", value)
```

value → Repeat

Memory Length

Description

Sets memory length or queries the current setting.

Parameter

1 to 32768

Example (Visual Basic)

```
' Set the memory length to 300.
ret = WeSetControl (hMo, "Memory Length", "300")
Dim value As Variant
ret = WeGetControl (hMo, "Memory Length", value)
```

value → 300

Pre Trigger:Cycle

Description

Sets the pretrigger (cycle) or queries the current setting.

Parameter

0 to maximum memory length (cycles)

Example (Visual Basic)

```
' Set the pretrigger to 1000 cycles.
ret = WeSetControl (hMo, "Pre Trigger:Cycle", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "Pre Trigger:Cycle", value)
```

value → 1000

Pre Trigger:Percent

Description

Sets the pretrigger or queries the current setting.

Parameter

0.0 to 100.0%

Example (Visual Basic)

```
' Set the pretrigger to 99.2%.
ret = WeSetControl (hMo, "Pre Trigger:Percent", "99.2")
Dim value As Variant
ret = WeGetControl (hMo, "Pre Trigger:Percent", value)
```

value → 99.2E+00

Misc:Time Base

Description
Sets the time base or queries the current setting.

Parameter
Internal | EXT CLOCK | BUSCLK

Example (Visual Basic)
```
' Set the time base to Internal.
ret = WeSetControl (hMo, "Misc:Time Base", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Time Base", value)
```
value → Internal

Misc:Output Clock

Description
Sets the clock output ON/OFF or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Enable the clock output.
ret = WeSetControl (hMo, "Misc:Output Clock", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Output Clock", value)
```
value → On

Misc:Arming

Description
Sets the arming setting or queries the current setting.

Parameter
Off | BUSTRG

Example (Visual Basic)
```
' Disable arming.
ret = WeSetControl (hMo, "Misc:Arming", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Arming", value)
```
value → Off

CH<x>:Trigger:Mask

Description
Sets the trigger mask or queries the current setting.

Parameter
0 to 4294967295 (0xffffffff)

Example (Visual Basic)

```
' Set the trigger mask to 100.
ret = WeSetControl (hMo, "CH1:Trigger:Mask", "100")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trigger:Mask", value)
```

value → 100

CH<x>:Trigger:Pattern

Description

Sets the trigger pattern or queries the current setting.

Parameter

0 to 4294967295 (0xffffffff)

Example (Visual Basic)

```
' Set the trigger pattern to 100.
ret = WeSetControl (hMo, "CH1:Trigger:Pattern", "100")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trigger:Pattern", value)
```

value → 100

CH<x>:Trigger:Point

Description

Queries the trigger point.

Parameter

0 to maximum memory length

Example (Visual Basic)

```
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trigger:Point", value)
```

value → 100

CH<x>:Trigger:Disable

Description

Sets whether or not to enable the trigger, or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Enable the trigger.
ret = WeSetControl (hMo, "CH1:Trigger:Disable", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trigger:Disable", value)
```

value → On

CH<x>:Memory:I/O Select

Description
Sets the input/output bits or queries the current setting.

Parameter
WE_UBYTE*Memory length

Example (Visual Basic)
```
' Allocate a buffer memory of 512 points.
Dim buf(512) As Byte
For i = 0 to 512-1
    buf(i) = 3      ' Substitute the input/output values.
Next i
' Set the bits.
ret = WeSetControl(hMo, "CH1:Memory:I/O Select", buf)
```

CH<x>:Memory:Output

Description
Sets the output pattern or queries the current setting.

Parameter
WE_ULONG*Memory length

Example (Visual Basic)
```
' Allocate a buffer memory of 512 points.
Dim buf(512) As Long
For i = 0 to 512-1
    buf(i) = I      ' Substitute the output pattern.
Next i
' Set the pattern.
ret = WeSetControl (hMo, "CH1:Memory:Output", buf)
```

CH<x>:Memory:Input

Description
Queries the input pattern.

Parameter
WE_ULONG*Memory length

Example (Visual Basic)
```
' Queries the input pattern.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Memory:Input", value)
```

Valid Acquisition Modes

Event	Valid?
Triggered	Yes
Free Run	No
Gate(Level)	No
Gate(Edge)	No

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	No	
WeGetAcqDataInfo	No	
WeGetAcqData	No	
WeGetScaleData	No	
WeGetMeasureParam	No	
WeSaveAcqData	No	
WeSaveScaleData	No	
WeSaveAsciiData	No	
WeGetCurrentData	No	
WeGetAcqDataEx	No	
WeGetAcqDataSize	No	
WeSaveAcqHeader	No	
WeSavePatternData	Yes	File extension is wpd.
WeLoadPatternData	Yes	File extension is wpo.
WeLoadPatternDataEx	No	
WeStartEx	No	
WeStopEx	No	

Valid Common Events

Event	Event Description
WWE_EV_MEASEND	No
WE_EV_BLOCKEND	No
WE_EV_MEASABORT	No
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

None

8.9 WE7141 100 MHz Universal Counter

ASCII Commands

ASCII Command	Description
Function	Sets the measurement function or queries the current setting.
Gate Time	Sets the gate time or queries the current setting.
Multiplier	Sets the multiplier or queries the current setting.
Ext Gate	Sets the external gate signal input or queries the current setting.
Misc:Arming	Sets the arming setting or queries the current setting.
Misc:D/A Output	Sets the D/A output or queries the current setting.
Misc:D/A Output:Scaling:0V	Sets the measurement value corresponding to 0 V output or queries the current setting.
Misc:D/A Output:Scaling:10V	Sets the measurement value corresponding to 10 V output or queries the current setting.
Misc:Clock	Sets the reference signal or queries the current setting.
CHA:Prescaler	Sets the prescaler value of CHA or queries the current setting.
CHA:Coupling	Sets the input coupling of CHA or queries the current setting.
CHB:Coupling	Sets the input coupling of CHB or queries the current setting.
CHA:Slope	Sets the slope of CHA or queries the current setting.
CHB:Slope	Sets the slope of CHB or queries the current setting.
CHA:Attenuator	Sets the attenuator of CHA or queries the current setting.
CHB:Attenuator	Sets the attenuator of CHB or queries the current setting.
CHA:Trigger Level:Auto	Sets the auto trigger of CHA or queries the current setting.
CHB:Trigger Level:Auto	Sets the auto trigger of CHB or queries the current setting.
CHA:Trigger Level	Sets the trigger level of CHA or queries the current setting.
CHB:Trigger Level	Sets the trigger level of CHB or queries the current setting.
Acq:Acq Mode	Sets the acquisition mode or queries the current setting.
Acq:Sampling Interval	Sets the sampling interval or queries the current setting.
Acq:Alarm:Mode	Sets the alarm mode or queries the current setting.
Acq:Alarm:High	Sets the upper limit of the alarm or queries the current setting.
Acq:Alarm:Low	Sets the lower limit of the alarm or queries the current setting.
Acq:Difference	Sets the totalize count difference output mode or queries the current setting.

Function

Description

Sets the measurement function or queries the current setting.

Parameter

Frequency A | Period A | T Interval A-B | Pulse Width A | Duty Cycle A | Ratio A/B | Totalize A

Example (Visual Basic)

```
' Set the measurement function to Period A.
ret = WeSetControl (hMo, "Function", "Period A")
Dim value As Variant
ret = WeGetControl (hMo, "Function", value)
```

value → Period A

Gate Time

Description

Sets the gate time or queries the current setting.

Parameter

10 ms to 10 s | Ext

Example (Visual Basic)

```
' Set the gate time to 10 ms.
ret = WeSetControl (hMo, "Gate Time", "10E-3")
Dim value As Variant
ret = WeGetControl (hMo, "Gate Time", value)
```

value → 10.0E–03

Multiplier

Description

Sets the multiplier or queries the current setting.

Parameter

×1 | ×10 | ×100 | ×1000

Example (Visual Basic)

```
' Set the multiplier to ×10.
ret = WeSetControl (hMo, "Multiplier", "×10")
Dim value As Variant
ret = WeGetControl (hMo, "Multiplier", value)
```

value → ×10

Ext Gate

Description

Sets the external gate signal input or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Disable the external gate signal input.
ret = WeSetControl (hMo, "Ext Gate", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Ext Gate", value)
```

value → Off

Misc:Arming

Description

Sets the arming setting or queries the current setting.

Parameter

Off | BUSTRG

Example (Visual Basic)

```
' Disable arming.
ret = WeSetControl (hMo, "Misc:Arming", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Arming", value)
```

value → Off

Misc:D/A Output

Description

Sets whether to enable or disable the D/A output, or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Disable the D/A output.
ret = WeSetControl (hMo, "Misc:D/A Output", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:D/A Output", value)
```

value → Off

Misc:D/A Output:Scaling:0V

Description

Sets the measurement value corresponding to 0 V output or queries the current setting.

Parameter

0 to 2^{52}(=4.5035996e15)

Example (Visual Basic)

```
' Set the measurement value corresponding to 0 V output to 1.2.
ret = WeSetControl (hMo, "Misc:D/A Output:Scaling:0V", 1.2")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:D/A Output:Scaling:0V", value)
```

value → 1.200000000E+00

Misc:D/A Output:Scaling:10V

Description

Sets the measurement value corresponding to 10 V output or queries the current setting.

Parameter

0 to 2^{52} (=4.5035996e15)

Example (Visual Basic)

```
' Set the measurement value corresponding to 10 V output to 120 M.
ret = WeSetControl (hMo, "Misc:D/A Output:Scaling:10V", "120E6")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:D/A Output:Scaling:10V", value)
```

value → 120.000000000000000E+06

Misc:Clock

Description
Sets the reference signal or queries the current setting.

Parameter
Int | Ext

Example (Visual Basic)
```
' Set the reference signal to internal clock.
ret = WeSetControl (hMo, "Misc:Clock", "Int")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Clock", value)
```
value → Int

CHA:Prescaler

Description
Sets the prescaler value of CHA or queries the current setting.

Parameter
1/1 | 1/2

Example (Visual Basic)
```
' Set the prescaler to 1/2.
ret = WeSetControl (hMo, "CHA:Prescaler", "1/2")
Dim value As Variant
ret = WeGetControl (hMo, "CHA:Prescaler", value)
```
value → 1/2

CHA:Coupling

Description
Sets the input coupling of CHA or queries the current setting.

Parameter
AC | DC

Example (Visual Basic)
```
' Set the input coupling of CHA to AC.
ret = WeSetControl (hMo, "CHA:Coupling", "AC")
Dim value As Variant
ret = WeGetControl (hMo, "CHA:Coupling", value)
```
value → AC

CHB:Coupling

Description
Sets the input coupling of CHB or queries the current setting.

Parameter
AC | DC

Example (Visual Basic)

```
' Set the input coupling of CHB to AC.
ret = WeSetControl (hMo, "CHB:Coupling", "AC")
Dim value As Variant
ret = WeGetControl (hMo, "CHB:Coupling", value)
```

value → AC

CHA:Slope

Description

Sets the slope of CHA or queries the current setting.

Parameter

Rise | Fall

Example (Visual Basic)

```
' Set the slope of CHA to Rise.
ret = WeSetControl (hMo, "CHA:Slope", "Rise")
Dim value As Variant
ret = WeGetControl (hMo, "CHA:Slope", value)
```

value → Rise

CHB:Slope

Description

Sets the input coupling of CHB or queries the current setting.

Parameter

Rise | Fall

Example (Visual Basic)

```
' Set the slope of CHB to Rise.
ret = WeSetControl (hMo, "CHB:Slope", "Rise")
Dim value As Variant
ret = WeGetControl (hMo, "CHB:Slope", value)
```

value → Rise

CHA:Attenuator

Description

Sets the attenuator of CHA or queries the current setting.

Parameter

×1 | ×10

Example (Visual Basic)

```
' Set the attenuator of CHA to ×10.
ret = WeSetControl (hMo, "CHA:Attenuator", "×10")
Dim value As Variant
ret = WeGetControl (hMo, "CHA:Attenuator", value)
```

value → ×10

CHB:Attenuator

Description
Sets the attenuator of CHB or queries the current setting.

Parameter
×1 | ×10

Example (Visual Basic)
```
' Set the attenuator of CHB to ×10.
ret = WeSetControl (hMo, "CHB:Attenuator", "×10")
Dim value As Variant
ret = WeGetControl (hMo, "CHB:Attenuator", value)
```
value → ×10

CHA:Trigger Level:Auto

Description
Enables/Disables the auto trigger on CHA or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Disable the auto trigger on CHA.
ret = WeSetControl (hMo, "CHA:Trigger Level:Auto", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "CHA:Trigger Level:Auto", value)
```
value → Off

CHB:Trigger Level:Auto

Description
Enables/Disables the auto trigger on CHB or gets the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Disable the auto trigger on CHB.
ret = WeSetControl (hMo, "CHB:Trigger Level:Auto", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "CHB:Trigger Level:Auto", value)
```
value → Off

CHA:Trigger Level

Description
Sets the trigger level of CHA or queries the current setting.

Parameter

When the attenuator is ×1 ±5 V (20 mV resolution)
When the attenuator is ×10 ±40 V (200 mV resolution)

Example (Visual Basic)

```
' Set the trigger level of CHA to 1 V.
ret = WeSetControl (hMo, "CHA:Trigger Level", "1")
Dim value As Variant
ret = WeGetControl (hMo, "CHA:Trigger Level", value)
```

value → 1.00E+00

CHB:Trigger Level

Description

Sets the trigger level of CHB or queries the current setting.

Parameter

When the attenuator is x1 ±5 V (20 mV resolution)
When the attenuator is x10 ±40 V (200 mV resolution)

Example (Visual Basic)

```
' Set the trigger level CHB to 2 V.
ret = WeSetControl (hMo, "CHB:Trigger Level", "2")
Dim value As Variant
ret = WeGetControl (hMo, "CHB:Trigger Level", value)
```

value → 2.00E+00

Acq:Acq Mode

Description

Sets the acquisition mode or queries the current setting.

Parameter

One Shot | Free Run

Example (Visual Basic)

```
' Set the acquisition mode to free run.
ret = WeSetControl (hMo, "Acq:Acq Mode", "Free Run")
Dim value As Variant
ret = WeGetControl (hMo, "Acq:Acq Mode", value)
```

value → Free Run

Acq:Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

10 ms to 100000 s

Example (Visual Basic)

```
' Set the sampling interval to 1 s.
ret = WeSetControl (hMo, "Acq:Sampling Interval", "1")
Dim value As Variant
ret = WeGetControl (hMo, "Acq:Sampling Interval", value)
```

value → 1.00E+00

Acq:Alarm:Mode

Description

Sets the alarm mode or queries the current setting.

Parameter

Off | Rise | Fall | In | Out

Example (Visual Basic)

```
' Disable the alarm mode.
ret = WeSetControl (hMo, "Acq:Alaram:Model", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Acq:Alarm:Mode", value)
```

value → Off

Acq:Alarm:High

Description

Sets the upper limit of the alarm or queries the current setting.

Parameter

0 to 10^{16}

Example (Visual Basic)

```
' Set the upper limit of the alarm to 1 V.
ret = WeSetControl (hMo, "Acq:Alarm:High", "1")
Dim value As Variant
ret = WeGetControl (hMo, "Acq:Alarm:High", value)
```

value → 1.000000000E+00

Acq:Alarm:Low

Description

Sets the lower limit of the alarm or queries the current setting.

Parameter

0 to 10^{16}

Example (Visual Basic)

```
' Set the lower limit of the alarm to 1 V.
ret = WeSetControl (hMo, "Acq:Alarm:Low", "1")
Dim value As Variant
ret = WeGetControl (hMo, "Acq:Alarm:Low", value)
```

value → 1.000000000E+00

Acq:Difference

Description

Sets the totalize count difference measurement or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable the totalize count difference measurement.
ret = WeSetControl (hMo, "Acq:Difference", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Acq:Difference", value)
```

value → On

Valid Acquisition Modes

Event	Valid?
Triggered	Yes
Free Run	Yes
Gate(Level)	No
Gate(Edge)	No

Acquisition Restrictions

Items	Possible Settings	Note
Record length	4096	
Data length per block	1 to 4096	Specify using the WeStartEx parameter.
Number of block	1 to 4096	Specify using the WeStartEx parameter.
Number of Acquisition	1 to ∞ (specify 0 for ∞)*	Specify using the WeStartEx parameter.

* Data length per block > memory length/2: 1
Data length per block ≤ memory length/2: 1 to ∞

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is physical value in double-precision real format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is physical value in double-precision real format.
WeGetAcqDataEx	Yes	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeLoadPatternDataEx	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?	Event Description
WWE_EV_MEASEND	Yes	
WE_EV_BLOCKEND	Yes	
WE_EV_MEASABORT	Yes	
WE_EV_TRIG_HIGH	Yes	Occurs when the alarm mode is set to "In" and data get inside the High/Low setting range, or when the alarm mode is set to "Out" and data get outside the High/Low setting range.
WE_EV_TRIG_LOW	Yes	Occurs when the alarm mode is set to "In" and data first get inside and then outside the High/Low setting range or when the alarm mode is set to "Out" and data first get outside and then inside the High/Low setting range.
WE_EV_TRIG_PULSE	Yes	Occurs when the alarm mode is set to "Rise" and data cross the "High" value from below, or when the alarm mode is set to "Fall" and data cross the "Low" value from above.
WE_EV_ALARM	No	
WE_EV_CLOSE_MODULE_GUI	Yes	

Module-specific Events

None

Module-specific Error Codes

None

8.10 WE7231 30-CH Fast Digital Thermometer

ASCII Commands

Commands Common to All Channels

ASCII Command	Description	GUI*
Reference Channel	Sets the reference channel when measuring the difference between two channels or queries the current setting.	Yes
Sampling Interval	Sets the sampling interval or queries the current setting.	Yes
NULL Meas	Executes measurement of the NULL value.	Yes
Integration Time	Sets the integration time or queries the current setting.	No

* "Yes" indicates that the equivalent operation can be carried out on the operation panel of the module, "No" indicates otherwise.

Burn Out Related Settings

ASCII Command	Description	GUI*
Burn Out:Auto	Sets the auto burn out check or queries the current setting.	Yes
Burn Out:Exec	Executes the burn out check.	Yes
Burn Out:Status	Queries the burn out occurrence status.	Yes
Burn Out:Detail:Status	Queries the details of the burn out occurrence condition.	No

* "Yes" indicates that the equivalent operation can be carried out on the operation panel of the module, "No" indicates otherwise.

Setup Related Settings

ASCII Command	Description	GUI*
Setup:RJC Source	Sets the RJC source or queries the current setting.	Yes
Setup:RJC Reference	Sets the reference value of the external RJC source or queries the current setting.	Yes
Setup:Time Base	Sets the time base or queries the current setting.	Yes
Setup:Unit	Sets the unit or queries the current setting.	Yes

* "Yes" indicates that the equivalent operation can be carried out on the operation panel of the module, "No" indicates otherwise.

Alarm Related Settings

ASCII Command	Description	GUI*1
Alarm: AllGr Hold:	Sets the alarm hold on all groups or queries the current setting.	Yes
Alarm:Gr1:Hold	Sets the alarm hold on group 1 or queries the current setting.	No
Alarm:Gr2:Hold	Sets the alarm hold on group 2 or queries the current setting.	No
Alarm:Gr3:Hold	Sets the alarm hold on group 3 or queries the current setting.	No
Alarm:Gr4:Hold	Sets the alarm hold on group 4 or queries the current setting.	No
Alarm: AllGr Reset:	Releases the held data of alarms of all groups.	Yes
Alarm:Gr1:Reset	Releases the held data of alarms of group 1.	No
Alarm:Gr2:Reset	Releases the held data of alarms of group 2.	No
Alarm:Gr3:Reset	Releases the held data of alarms of group 3.	No
Alarm:Gr4:Reset	Releases the held data of alarms of group 4.	No
Alarm:Gr1:Combination	Sets the alarm combination of group 1 or queries the current setting.	Yes
Alarm:Gr2:Combination	Sets the alarm combination of group 2 or queries the current setting.	Yes
Alarm:Gr3:Combination	Sets the alarm combination of group 3 or queries the current setting.	Yes
Alarm:Gr4:Combination	Sets the alarm combination of group 4 or queries the current setting.	Yes

ASCII Command	Description	GUI*
Alarm:Gr1:Out	Sets the alarm result output of alarms of group 1 or queries the current setting.	Yes
Alarm:Gr2:Out	Sets the alarm result output of alarms of group 2 or queries the current setting.	Yes
Alarm:Gr3:Out	Sets the alarm result output of alarms of group 3 or queries the current setting.	Yes
Alarm:Gr4:Out	Sets the alarm result output of alarms of group 4 or queries the current setting.	Yes
Alarm:Gr1:Status	Queries the occurrence status of alarms of group 1.	Yes/No*2
Alarm:Gr2:Status	Queries the occurrence status of alarms of group 2.	Yes/No*2
Alarm:Gr3:Status	Queries the occurrence status of alarms of group 3.	Yes/No*2
Alarm:Gr4:Status	Queries the occurrence status of alarms of group 4.	Yes/No*2
Alarm:Gr1:HoldStatus	Sets the hold status of alarms of group 1 or queries the current setting.	Yes/No*2
Alarm:Gr2:HoldStatus	Sets the hold status of alarms of group 2 or queries the current setting.	Yes/No*2
Alarm:Gr3:HoldStatus	Sets the hold status of alarms of group 3 or queries the current setting.	Yes/No*2
Alarm:Gr4:HoldStatus	Sets the hold status of alarms of group 4 or queries the current setting.	Yes/No*2
Alarm:Gr1:Detail:Status	Queries the details of the occurrence status of alarms of group 1.	No
Alarm:Gr2:Detail:Status	Queries the details of the occurrence status of alarms of group 2.	No
Alarm:Gr3:Detail:Status	Queries the details of the occurrence status of alarms of group 3.	No
Alarm:Gr4:Detail:Status	Queries the details of the occurrence status of alarms of group 4.	No
Alarm:Gr1:Detail:HoldStatus	Queries the details of the hold status of alarms of group 1.	No
Alarm:Gr2:Detail:HoldStatus	Queries the details of the hold status of alarms of group 2.	No
Alarm:Gr3:Detail:HoldStatus	Queries the details of the hold status of alarms of group 3.	No
Alarm:Gr4:Detail:HoldStatus	Queries the details of the hold status of alarms of group 4.	No

*1 "Yes" indicates that the equivalent operation can be carried out on the operation panel of the module, "No" indicates otherwise.
*2 "Yes/No" indicates that the equivalent operation can or cannot be carried out on the operation panel of the module depending on whether alarm hold is specified.

CH<x>:

ASCII Command	Description	GUI*1
CH<x>:Range	Sets the measurement range (measurement function) or queries the current setting.	Yes
CH<x>:NULL	Sets the use of the null value or queries the current setting.	Yes
CH<x>:Delta	Sets the use of delta or queries the current setting.	Yes
CH<x>:Average	Sets the average or queries the current setting.	Yes
CH<x>:Average:Count	Sets the average count or queries the current setting.	Yes
CH<x>:Alarm:Type	Sets the alarm type or queries the current setting.	Yes
CH<x>:Alarm:High	Sets the alarm high level or queries the current setting.	Yes
CH<x>:Alarm:Low	Sets the alarm low level or queries the current setting.	Yes
CH<x>:Alarm:Group	Sets the group to which the alarm belongs or queries the current setting.	Yes
CH<x>:Alarm:Status	Queries the alarm occurrence status.	Yes/No*2
CH<x>:AlarmHoldStatus	Queries the alarm hold status.	Yes/No*2
CH<x>:Burn Out:Status	Queries the burn out occurrence status.	Yes

*1 "Yes" indicates that the equivalent operation can be carried out on the operation panel of the module, "No" indicates otherwise.
*2 "Yes/No" indicates that the equivalent operation can or cannot be carried out on the operation panel of the module depending on whether alarm hold is specified.

Note

- Enter the channel number in the <x> of CH<x>. If the modules are linked specify a serial number from the parent module.
- There are 30 channels on each WE7231 module. If two modules are linked and you wish to change the range of channel 1 of the second module, specify "CH31:Range" in the command line.

Reference Channel

Description

Sets the reference channel or queries the current setting.

Parameter

CH1 to CH240

If the modules are linked specify a serial number from the parent module.

There are 30 channels on each WE7231 module.

If two modules are linked and you wish to set channel 1 of the second module, specify "CH31."

Example (Visual Basic)

```
' Set the reference channel to CH16.
ret = WeSetControl (hMo, "Reference Channel", "C,g16")
Dim value As Variant
ret = WeGetControl (hMo, "Reference Channel", value)
```

value → CH16

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

2-MΩ Meas. Range Not Used/Used*		Not Used*				Used			
Time Base		Internal		External (CMNCLK)		Internal		External (CMNCLK)	
Fast Scan Mode		ON	OFF	ON	OFF	ON	OFF	ON	OFF
Integration Time	Auto	60 ms	100 ms	100 ms	150 ms	460 ms	900 ms	500 ms	950 ms
	1.0ms	60 ms	100 ms	100 ms	150 ms	460 ms	900 ms	500 ms	950 ms
	4.0ms	120 ms	200 ms	150 ms	250 ms	520 ms	1000 ms	550 ms	1050 ms
	16.6/20.0ms	450 ms	800 ms	600 ms	1100 ms	850 ms	1600 ms	1000 ms	1900 ms
	100ms	2.0 s	3.5 s	2.1 s	3.7 s	2.40 s	4.30 s	2.50 s	4.50 s

Maximum setting: 60 min. Resolution: 10 ms and 1 min for 60 to 990 ms and 1 min to 60 min, respectively.

* If any one of the channels is using a measurement range of 2 MΩ, sampling interval under "Used" is applied.

Example (Visual Basic)

```
' Set the sampling interval to 1 s.
ret = WeSetControl (hMo, "Sampling Interval", "1E+00")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 1.0E+00

NULL Meas

Description

Executes the measurement of the NULL value.

Parameter

None

Example (Visual Basic)

```
Execute the measurement of the NULL value.
Dim value As Variant
ret = WeSetControl (hMo, "NULL Meas", value)
```

Integration Time

Description

Sets the integration time or queries the current setting.
Can be specified only when using the API.
If you select "Auto," the integration time is set according to the sampling interval.
If "16.6/20.0ms" is selected, the integration time is automatically set to 16.6 ms and 20.0 ms for 60 Hz and 50 Hz, respectively.

Parameter

Auto | 1.0ms | 4.0ms | 16.6/20.0ms | 100.0ms

Example (Visual Basic)

```
' Set the integration time to 16.6/20.0ms (fixed).
ret = WeSetControl (hMo, "Integration Time", "16.6/20.0ms")
Dim value As Variant
ret = WeGetControl (hMo, "Integration Time", value)
```

value → 16.6/20.0ms

Burn Out: Auto

Description

Sets the auto burn out check or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Turn OFF the auto burn out check.
ret = WeSetControl (hMo, "Burn Out:Auto", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Burn Out:Auto", value)
```

value → Off

Burn Out:Exec

Description
Checks the burn out occurrence status.

Parameter
None

Example (Visual Basic)
```
' Check the burn out occurrence status.
Dim value As Variant
ret = WeSetControl (hMo, "Burn Out:Exec", value)
```

Burn Out:Status

Description
Queries the result of the combination (OR) of the burn out occurrence status.

Parameter
Off (burn out not detected) | On (burn out detected)

Example (Visual Basic)
```
' Query the result of the combination (OR) of the burn out
' occurrence status.
Dim value As Variant
ret = WeGetControl (hMo, "Burn Out:Status", value)
```
value → Off

Burn Out:Detail:Status

Description
Queries the details of the burn out occurrence condition.

Parameter
WE_ULONG × number of linked modules

Example (Visual Basic)
```
' Query the details of the burn out occurrence condition.
Dim value As Variant
ret = WeGetControl (hMo, "Burn Out:Detail:Status", value)
```
The alarm occurrence status of CH1 to 30 is assigned to bits 0 through 29 and returned.
A value of 0 and 1 correspond to "alarm not occurring" and "alarm occurring," respectively.

Setup:RJC Source

Description
Sets the RJC source or queries the current setting.

Parameter
Internal | External

Example (Visual Basic)

```
' Set the RJC source to External.
ret = WeSetControl (hMo, "Setup:RJC Source", "External")
Dim value As Variant
ret = WeGetControl (hMo, "Setup:RJC Source", value)
```

value → External

Setup:RJC Reference

Description

Sets the reference value of the external RJC source or queries the current setting.

Parameter

The reference value is dependent on the measurement range.
For details, see the user's manual for the module.

Example (Visual Basic)

```
' Set the reference value of the external RJC source to 0.0.
ret = WeSetControl (hMo, "Setup:RJC Reference", "0.0")
Dim value As Variant
ret = WeGetControl (hMo, "Setup:RJC Reference", value)
```

value → 0.0E+00

Setup:Time Base

Description

Sets the time base or queries the current setting.

Parameter

Internal | BUSCLK

Example (Visual Basic)

```
' Set the time base to BUSCLK.
ret = WeSetControl (hMo, "Setup:Time Base","BUSCLK")
Dim value As Variant
ret = WeGetControl (hMo, "Setup:Time Base", value)
```

value → BUSCLK

Setup:Unit

Description

Sets the temperature unit that is to be used or queries the current setting.

Parameter

C | K | F

Example (Visual Basic)

```
' Set the temperature unit to be used to K.
ret = WeSetControl (hMo, "Setup:Unit", "K")
Dim value As Variant
ret = WeGetControl (hMo, "Setup:Unit", value)
```

value → K

Setup:Fast Scan Mode

Description

Sets the fast scan mode or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Turn OFF the fast scan mode.
ret = WeSetControl (hMo, "Setup:Fast Scan Mode", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Setup:Fast Scan Mode", value)
```

value → Off

Alarm:AllGr:Hold
Alarm:Gr1:Hold
Alarm:Gr2:Hold
Alarm:Gr3:Hold
Alarm:Gr4:Hold

Description

Sets the hold function of the alarm result or queries the current setting.
AllGr is used to set all groups, Gr1, Gr2, Gr3, and Gr4, at once. For the query, OFF is returned only when all groups are OFF.

Parameter

Off | On

Example (Visual Basic)

```
' Turns OFF the hold function of the alarm result.
ret = WeSetControl (hMo, "Alarm:AllGr:Hold", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Alarm:AllGr:Hold", value)
```

value → Off

Alarm:AllGr:Reset
Alarm:Gr1:Reset
Alarm:Gr2:Reset
Alarm:Gr3:Reset
Alarm:Gr4:Reset

Description

Releases the hold function of the alarm result.
AllGr is used to release the hold function of all groups, Gr1, Gr2, Gr3, and Gr4, at once.

Parameter

None

Example (Visual Basic)

```
' Release the alarm result of group 1.
Dim value As Variant
ret = WeSetControl (hMo, "Alarm:Gr1:Reset", value)
```

Alarm:Gr1:Combination
Alarm:Gr2:Combination
Alarm:Gr3:Combination
Alarm:Gr4:Combination

Description

Sets the alarm combination or queries the current setting.

Parameter

AND | OR

Example (Visual Basic)

```
' Set the alarm combination of group 2 to AND.
ret = WeSetControl (hMo, "Alarm:Gr2:Combination", "AND")
Dim value As Variant
ret = WeGetControl (hMo, "Alarm:Gr2:Combination", value)
```

value → AND

Alarm:Gr1:Out
Alarm:Gr2:Out
Alarm:Gr3:Out
Alarm:Gr4:Out

Description

Sets the output of the alarm detection result to the bus trigger or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Turn ON the bus trigger output of the alarm detection result of
' group 3.
ret = WeSetControl (hMo, "Alarm:Gr3:Out", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Alarm:Gr3:Out", value)
```

value → On

Alarm:Gr1:Status
Alarm:Gr2:Status
Alarm:Gr3:Status
Alarm:Gr4:Status

Description
Queries the alarm occurrence status.

Parameter
Off | On

Example (Visual Basic)
```
' Query the alarm occurrence status of group 1.
Dim value As Variant
ret = WeGetControl (hMo, "Alarm:Gr1:Status", value)
```
value → ON

Alarm:Gr1:HoldStatus
Alarm:Gr2:HoldStatus
Alarm:Gr3:HoldStatus
Alarm:Gr4:HoldStatus

Description
Queries the occurrence status of the held alarm for the group.

Parameter
Off | On

Example (Visual Basic)
```
' Query the alarm occurrence status of group 1 during hold
' operation.
Dim value As Variant
ret = WeGetControl (hMo, "Alarm:Gr1:HoldStatus", value)
```
value → OFF

Alarm:Gr1:Detail:Status
Alarm:Gr2:Detail:Status
Alarm:Gr3:Detail:Status
Alarm:Gr4:Detail:Status

Description
Queries the details of the alarm occurrence status.

Parameter
WE_ULONG × number of linked modules

Example (Visual Basic)
```
' Query the details of the alarm occurrence status of group 1.
Dim value As Variant
ret = WeGetControl (hMo, "Alarm:Gr1:Detail:Status", value)
```
The alarm occurrence status of CH1 to 30 is assigned to bits 0 through 29 and returned.
A value of 0 and 1 correspond to "alarm not occurring" and "alarm occurring," respectively.

Alarm:Gr1:Detail:HoldStatus
Alarm:Gr2:Detail:HoldStatus
Alarm:Gr3:Detail:HoldStatus
Alarm:Gr4:Detail:HoldStatus

Description
Queries the details of the alarm occurrence status during hold operation.

Parameter
WE_ULONG × number of linked modules

Example (Visual Basic)
```
' Query the details of the alarm occurrence status of group 1
' during hold operation.
Dim value As Variant
ret = WeGetControl (hMo, "Alarm:Gr1:Detail:HoldStatus", value)
```
The alarm occurrence status of CH1 to 30 is assigned to bits 0 through 29 and returned.
A value of 0 and 1 correspond to "alarm not occurring" and "alarm occurring," respectively.

CH<x>:Range

Description
Sets the measurement range (measurement function) of the channel or queries the current setting.

Parameter
The measurement function is automatically determined when the measurement range is selected. For details, see the user's manual for the module.

Example (Visual Basic)
```
' Set the measurement range setting of CH3 to Type T.
ret = WeSetControl (hMo, "CH3:Range", "Type T")
Dim value As Variant
ret = WeGetControl (hMo, "CH3:Range", value)
```
value → Type T

CH<x>:NULL

Description
Turns ON/OFF the NULL value computation of the channel or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Turn ON the NULL value computation of CH5.
ret = WeSetControl (hMo, "CH5:NULL", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH5:NULL", value)
```
value → On

CH<x>:Delta

Description

Turns ON/OFF the delta computation of the channel or queries the current setting.
The reference channel of the delta computation is the channel selected through Reference Channel.
The delta computation can be turned ON only when the measurement condition (temperature, voltage, or resistance) of the specified channel is the same as the Reference Channel.

Parameter

Off | On

Example (Visual Basic)

```
' Turn ON the delta computation of CH6.
ret = WeSetControl (hMo, "CH6:Delta", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH6:Delta", value)
```

value → On

CH<x>:Average

Description

Turns ON/OFF the average computation or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Turn OFF the average computation of CH7.
ret = WeSetControl (hMo, "CH7:Average", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "CH7:Average", value)
```

value → Off

CH<x>:Average:Count

Description

Sets the number of average computations or queries the current setting.

Parameter

2 to 100

Example (Visual Basic)

```
' Set the number of average computation of CH8 to 50.
ret = WeSetControl (hMo, "CH8:Average:Count", "50")
Dim value As Variant
ret = WeGetControl (hMo, "CH8:Average:Count", value)
```

value → 50E+00

CH<x>:Alarm:Type

Description
Sets the alarm type or queries the current setting.

Parameter
Off | Rise | Fall | High | Low | In | Out

Example (Visual Basic)
```
' Set the alarm type of CH9 to High.
ret = WeSetControl (hMo, "CH9:Alarm:Type", "High")
Dim value As Variant
ret = WeGetControl (hMo, "CH9:Alarm:Type", value)
```
value → High

CH<x>:Alarm:High

Description
Sets the alarm High level or queries the current setting.

Parameter
The alarm High level is dependent on the measurement range. For details, see the user's manual for the module.

Example (Visual Basic)
```
' Set the alarm high level of CH10 to 100.0.
ret = WeSetControl (hMo, "CH10:Alarm:High", "100.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH10:Alarm:High", value)
```
value → 100.0E+00

CH<x>:Alarm:Low

Description
Sets the alarm Low level or queries the current setting.

Parameter
The alarm High level is dependent on the measurement range. For details, see the user's manual for the module.

Example (Visual Basic)
```
' Set the alarm Low level of CH11 to 50.0.
ret = WeSetControl (hMo, "CH11:Alarm:Low", "50.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH11:Alarm:Low", value)
```
value → 50.0E+00

CH<x>:Alarm:Group

Description
Sets the group to which the alarm belongs or queries the current setting.

Parameter

1 | 2 | 3 | 4

Example (Visual Basic)

```
' Set the group to which the alarm of CH11 belongs to Gr1.
ret = WeSetControl (hMo, "CH11:Alarm:Group", "1")
Dim value As Variant
ret = WeGetControl (hMo, "CH11:Alarm:Group", value)
```

value → 1.0E+00

CH<x>:Alarm:Status

Description

Queries the current alarm occurrence status.

Parameter

Off (alarm not detected) | On (alarm detected)

Example (Visual Basic)

```
' Query the current alarm occurrence status of CH12.
Dim value As Variant
ret = WeGetControl (hMo, "CH12:Alarm:Status", value)
```

value → On

CH<x>:Alarm:HoldStatus

Description

Queries the occurrence status of the held alarm for the channel.

Parameter

Off (alarm not detected) | On (alarm detected)

Example (Visual Basic)

```
' Query the alarm occurrence status during hold operation of CH13.
Dim value As Variant
ret = WeGetControl (hMo, "CH13:Alarm:HoldStatus", value)
```

value → Off

CH<x>:Burn Out:Status

Description

Queries the occurrence status of burn out.

Parameter

Off (burn out not detected) | On (burn out detected)

Example (Visual Basic)

```
' Query the occurrence status of burn out of CH14.
Dim value As Variant
ret = WeGetControl (hMo, "CH14:Burn Out:Status", value)
```

value → On

Valid Acquisition Modes

Acquisition Type	Valid?	Note
Triggered	No	
Free Run	Yes	
Gate(Level)	No	
Gate(Edge)	No	

Acquisition Restrictions

Item	Possible Settings	Note
Memory length	128	
Data length per block	1 to 64	Specified through a parameter of WeStartEx.
Number of blocks	Ignored.	Ignored because only free run mode is available.
Number of Acquisitions	1 to ∞ (∞ is set with a 0)	Specified through a parameter of WeStartEx.

Note

Since the memory length is 128, if you operate using a sampling rate of 100 ms, an overrun occurs 12.8 s after the start of the measurement. Make sure to read the data before an overrun occurs.

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	No	
WeStartWithEvent	No	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is physical value in single-precision real format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is physical value in single-precision real format.
WeGetAcqDataEx	No	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeLoadPatternDataEx	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?	Note
WE_EV_MEASSTART	Yes	Generated when the start operation completes.
WE_EV_MEASEND	Yes	Generated when the specified number of blocks of data are acquired.
WE_EV_BLOCKEND	Yes	Generated each time data is acquired to the block when acquiring data using memory partitions (blocks).
WE_EV_MEASABORT	Yes	Generated when the start state (RUN state) is cleared.
WE_EV_TRIG_HIGH	Yes	Generated when the alarm type is "High," "Low," "In," or "Out" and the result of the determination is TRUE. However, if combination is set to AND and the alarm type of other channels is Rise or Low, the event is not generated.
WE_EV_TRIG_LOW	Yes	Generated when the alarm type is "High," "Low," "In," or "Out" and the result of the determination is FALSE. However, if combination is set to AND and the alarm type of other channels is Rise or Low, the event is not generated.
WE_EV_TRIG_PULSE	Yes	Generated when the alarm type is "Rise" or "Fall" and the result of the determination is TRUE.
WE_EV_ALARM	Yes	See page 5-3 on the module's user's manual.

Module-specific Events

Event	Note
0x00040000	Alarm result of group 1 The same as WE_EV_TRIG_PULSE
0x00080000	Alarm result of group 2 The same as WE_EV_TRIG_PULSE
0x00100000	Alarm result of group 3 The same as WE_EV_TRIG_PULSE
0x00200000	Alarm result of group 4 The same as WE_EV_TRIG_PULSE
0x00400000	Alarm result of group 1 The same as WE_EV_TRIG_HIGH
0x00800000	Alarm result of group 2 The same as WE_EV_TRIG_HIGH
0x01000000	Alarm result of group 3 The same as WE_EV_TRIG_HIGH
0x02000000	Alarm result of group 4 The same as WE_EV_TRIG_HIGH
0x04000000	Alarm result of group 1 The same as WE_EV_TRIG_LOW
0x08000000	Alarm result of group 2 The same as WE_EV_TRIG_LOW
0x10000000	Alarm result of group 3 The same as WE_EV_TRIG_LOW
0x20000000	Alarm result of group 4 The same as WE_EV_TRIG_LOW
0x40000000	Burn out detected

Module-specific Error Codes

None

8.11 WE7235 4-CH, 100 kS/s Accelerometer Module

ASCII Commands

ASCII Command	Description
Acquisition Mode	Sets the acquisition mode or queries the current setting.
Sampling Interval	Sets the sampling interval or queries the current setting.
Record Length	Sets the record length or queries the current setting.
Memory Partition	Sets the memory partitions or queries the current setting.
No.Of Acquisitions	Sets the number of acquisitions or queries the current setting.

CH<x>:

ASCII Command	Description
CH<x>:On	Sets the measurement channel or queries the current setting.
CH<x>:Bias	Sets the bias current output or queries the current setting.
CH<x>:Coupling	Sets the input coupling or queries the current setting.
CH<x>:Unit	Sets the measurement range unit or queries the current setting.
CH<x>:Range	Sets the range or queries the current setting.
CH<x>:Sensitivity	Sets the sensitivity of the sensor connected to the channel or queries the current setting.
CH<x>:AAF	Sets the input filter type or queries the current setting.
CH<x>:Filter	Sets the input filter frequency or queries the current setting.
CH<x>:Trig Type	Sets the trigger type or queries the current setting.
CH<x>:Trig Level	Sets the trigger level or queries the current setting.

Trig:

ASCII Command	Description
Trig:Trigger:Source	Sets the trigger source or queries the current setting.
Trig:Trigger:Combination	Sets the trigger combination or queries the current setting.
Trig:Trigger:Pretrigger	Sets the amount of pretrigger or queries the current setting.
Trig:Trigger:Hold Off	Sets the amount of hold off or queries the current setting.
Trig:Misc:Time Base	Sets the time base or queries the current setting.
Trig:Misc:FFT Interval	Sets the sampling interval for the FFT or queries the current setting.
Trig:Misc:CH Mode	Sets the number of measurement channels or queries the current setting.
Trig:Overlapped Acquisition	Enables or disables overlapped acquisition or queries the current setting.
Trig:Manual Trigger	Executes a manual trigger.

ConnectionTest

ASCII Command	Description
ConnectionTest:On	Sets the connection test mode or queries the current setting.
ConnectionTest:Exec	Executes the connection test.
ConnectionTest:CH<x>:On	Sets the channel on which to perform the connection test or queries the current setting.
ConnectionTest:CH<x>:Result	Queries the result of the connection test.

Note

- Enter the channel number in the <x> of CH<x>. If the modules are linked specify a serial number from the parent module.
- There are 4 channels on each WE7235 module.
- If two modules are linked and you wish to change the range of channel 2 of the second module, specify "CH6:Range" in the command line.

Acquisition Mode

Description

Sets the acquisition mode or queries the current setting.

Parameter

Triggered | Free Run | Gate(Level) | Gate(Edge)

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Acquisition Mode", "Free Run")
' Set the acquisition mode to free run.
Dim value As Variant
ret = WeGetControl (hMo, "Acquisition Mode", value)
```

value → Free Run

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

0.00001 to 10.0

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Sampling Interval", "1E-3")
' Set the sampling interval to 1 ms.
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 1.000E-03

Record Length

Description

Sets the record length or queries the current setting.

Parameter

The selectable range varies depending on the acquisition mode.

During trigger mode	2 to 4194304/(number of memory partitions × number of channels)
During gate mode	2 to 4194304/number of memory partitions
During free run mode	1 to 4194304/number of memory partitions

However, the record length range changes so that the measurement period does not fall below 5 ms.

Example: When the sampling interval, the number of measurement channels, and the number of memory partitions are set to 100 μs, 4 CH, and 256, respectively.

During trigger mode	50 to 4096
During gate mode	50 to 1048576
During free run mode	50 to 1048576

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Record Length", "1000")
' Set the record length to 1000 points.
Dim value As Variant
ret = WeGetControl (hMo, "Record Length", value)
```

value → 1000

Memory Partition

Description

Sets the number of memory partitions or queries the current setting.

Parameter

1 to 256(2^n steps)

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Memory Partition", "256")
' Set the number of memory partitions to 256.
Dim value As Variant
ret = WeGetControl (hMo, "Memory Partition", value)
```

value → 256.0E+00

No. of Acquisitions

Description

Sets the number of acquisitions or queries the current setting.

Parameter

0 to 32768

Example (Visual Basic)

```
ret = WeSetControl (hMo, "No.Of Acquisitions", "100")
' Set the number of acquisitions to 100.
Dim value As Variant
ret = WeGetControl (hMo, "No.Of Acquisitions", value)
```

value → 100

CH<x>:On

Description

Turns ON/OFF the channel or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
ret = WeSetControl (hMo, "CH1:On", "On")
' Turn ON the measurement of CH1.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:On", value)
```

value → On

CH<x>:Bias

Description
Turns ON/OFF the bias current that is to be supplied to the acceleration sensor that is connected to the channel.

Parameter
Off | On

Example (Visual Basic)
```
ret = WeSetControl (hMo, "CH1:Bias", "On")
' Turn ON the bias current of CH1.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Bias", value)
```
value → On

CH<x>:Coupling

Description
Sets the input coupling or queries the current setting.

Parameter
When the measurement range unit is set to V
DC | AC
When the measurement range unit is set to m/s2 (m/s^2) or G
AC only

Example (Visual Basic)
```
ret = WeSetControl (hMo, "CH1:Coupling", "AC")
' Set the input coupling of CH1 to AC.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Coupling", value)
```
value → AC

CH<x>:Unit

Description
Sets the measurement range unit or queries the current setting.

Parameter
V | m/s2 | G

Example (Visual Basic)
```
ret = WeSetControl (hMo, "CH1:Unit", "m/s2")
' Set the measurement range unit of CH1 to m/s2.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Unit", value)
```
value → m/s2

CH<x>:Range

Description
Sets the measurement range or queries the current setting.

Parameter
When the measurement range unit is set to V
50 mV | 100 mV | 250 mV | 500 mV | 1 V | 2.5 V | 5 V | 10 V | 25 V | 50 V
When the measurement range unit is set to m/s2 (m/s^2) or G
×1 | ×2 | ×5 | ×10 | ×20 | ×50 | ×100

Example (Visual Basic)
```
ret = WeSetControl (hMo, "CH1:Range", "1V")
' Set the measurement range of CH1 to 1 V.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Range", value)
```
value → 1.0E+00

CH<x>:Sensitivity

Description
Sets the sensitivity of the sensor connected to the channel or queries the current setting.

Parameter
Invalid when the measurement range unit is set to V
When the measurement range unit is set to m/s2(m/s^2) or G
0.0 to 9999.0

Example (Visual Basic)
```
ret = WeSetControl (hMo, "CH1:Sensitivity", "1000.0")
' Set the sensor sensitivity of CH1 to 1000.0.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Sensitivity", value)
```
value → 1.0000E+03

CH<x>:AAF

Description
Sets the input filter type or queries the current setting.

Parameter
Off (low-pass filter) | On (anti-aliasing filter)

Example (Visual Basic)
```
ret = WeSetControl (hMo, "CH1:AAF", "On")
' Set the filter of CH1 to anti-aliasing filter.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:AAF", value)
```
value → On

CH<x>:Filter

Description

Sets the input filter frequency or queries the current setting.

Parameter

When the filter is a low-pass filter
Off | 40 Hz | 100 Hz | 400 Hz | 1 kHz | 4 kHz | 10 kHz | 40 kHz
When the filter is an anti-aliasing filter
Off | 20 Hz | 40 Hz | 80 Hz | 200 Hz | 400 Hz | 800 Hz | 2 kHz | 4 kHz | 8 kHz | 20 kHz | 40 kHz

Example (Visual Basic)

```
ret = WeSetControl (hMo, "CH1:Filter", "40kHz")
' Set the filter frequency of CH1 to 40 kHz.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Filter", value)
```

value → 40.0000E+03

CH<x>:Trig Type

Description

Sets the trigger type or queries the current setting.

Parameter

When the acquisition mode is set to trigger mode or gate mode (edge)
Rise | Fall | Both | High | Low | Off
When the acquisition mode is set to gate mode (level)
High | Low | Off

Example (Visual Basic)

```
ret = WeSetControl (hMo, "CH1:Trig Type", "Rise")
' Set the trigger type of CH1 to Rise.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Type", value)
```

value → Rise

CH<x>:Trig Level

Description

Sets the trigger level or queries the current setting.

Parameter

When the measurement range unit is set to V
Measurement range 50 mV –0.05 to 0.05
Measurement range 100 mV –0.1 to 0.1
Measurement range 250 mV –0.25 to 0.25
Measurement range 500 mV –0.5 to 0.5
Measurement range 1 V –1.0 to 1.0
Measurement range 2.5 V –2.0 to 2.0
Measurement range 5 V –5.0 to 5.0
Measurement range 10 V –10.0 to 10.0
Measurement range 25 V –25.0 to 25.0
Measurement range 50 V –50.0 to 50.0
When the measurement range unit is set to m/s2 (m/s^2) or G
1 to 100 (%)

Example (Visual Basic)

```
ret = WeSetControl (hMo, "CH1:Trig Level", "60")
' Set the trigger level CH1 to 60%.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Level", value)
```

value → 60E+00

Trig:Trigger:Source

Description

Sets the trigger source or queries the current setting.

Parameter

Internal | BUSTRG

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Trigger:Source", "Internal")
' Set the trigger source to Internal.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Source", value)
```

value → Internal

Trig:Trigger:Combination

Description

Sets the trigger combination or queries the current setting.

Parameter

AND | OR

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Trigger:Combination", "OR")
' Set the trigger combination to OR.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Combination", value)
```

value → OR

Trig:Trigger:PreTrigger

Description

Sets the amount of pretrigger or queries the current setting.

Parameter

0 to record length – 2

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Trigger:PreTrigger", "1000")
' Set the amount of pretrigger to 1000.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:PreTrigger", value)
```

value → 1000

Trig:Trigger:Hold Off

Description

Sets the amount of hold off or queries the current setting.

Parameter

When the acquisition mode is set to trigger mode and overlapped acquisition is disabled
Specified record length to 4194304
When the acquisition mode is set to trigger mode and overlapped acquisition is enabled
1 to 4194304
When the acquisition mode is set to free run mode or gate mode (edge)
1 to 4194304

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Hold Off", "1000")
' Set the hold off time to 1000.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Hold Off", value)
```

value → 1000

Trig:Misc:Time Base

Description

Sets the time base or queries the current setting.

Parameter

Internal | BUSCLK | FFT

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Misc:Time Base", "BUSCLK")
' Set the time base to BUSCLK.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:Time Base", value)
```

value → BUSCLK

Trig:Misc:FFT Interval

Description
Sets the sampling interval for the FFT or queries the current setting.

Parameter
51.2 Hz | 102.4 Hz | 204.8 Hz | 512 Hz | 1.024 kHz | 2.048 kHz | 5.12 kHz | 10.24 kHz | 20.48 kHz | 51.2 kHz

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Trig:Misc:FFT Interval", "51.2kHz")
' Set the sampling interval for the FFT to 51.2 kHz.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:FFT Interval", value)
```
value → 51.2000E+03

Trig:Misc:CH Mode

Description
Sets the number of measurement channels or queries the current setting.

Parameter
1CH | 2CH | 4CH

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Trig:Misc:CH Mode", "4CH")
' Set the number of measurement channels to 4CH.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:CH Mode", value)
```
value → 4CH

Trig:Overlapped Acquisition

Description
Enables or disables overlapped acquisition or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Trig:Overlapped Acquisition", "On")
' Enable overlapped acquisition.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Overlapped Acquisition", value)
```
value → On

Trig:Manual Trigger

Description
Executes a manual trigger.

Parameter
None

Example (Visual Basic)

```
' Execute a manual trigger.
Dim value As Variant
ret = WeSetControl (hMo, "Trig:Manual Trigger", value)
```

ConnectionTest:On

Description

Enables or disables the execution of the connection test or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
ret = WeSetControl (hMo, "ConnectionTest:On", "On")
' Enable the execution of the connection test.
Dim value As Variant
ret = WeGetControl (hMo, "ConnectionTest:On", value)
```

value → On

ConnectionTest:Exec

Description

Executes the connection test.

Parameter

None

Example (Visual Basic)

```
 ' Execute the connection test.
Dim value As Variant
ret = WeSetControl (hMo, "ConnectionTest:Exec", value)
```

ConnetionTest:CH<x>:On

Description

Sets the channel on which to perform the connection test or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
ret = WeSetControl (hMo, "ConnectionTest:CH1:On", "On")
' Set the connection test target to CH1.
Dim value As Variant
ret = WeGetControl (hMo, "ConnectionTest:CH1:On", value)
```

value → On

ConnetionTest:CH<x>:Result

Description

Queries the result of the connection test.

Parameter

OK | Short | Open | --

Example (Visual Basic)

```
Dim value As Variant
' Queries the result of the connection test.
ret = WeGetControl (hMo, "ConnectionTest:CH1:Result", value)
```

value → OK

Valid Acquisition Modes

Acquisition Type	Valid?	Description
Triggered	Yes	
Free Run	Yes	
Gate(Level)	Yes	
Gate(Edge)	Yes	

Acquisition Restrictions

	Description
Memory length	1048576 (when 4CH mode is selected) 2097152 (when 2CH mode is selected) 4194304 (when 1CH mode is selected)
Data length per block	During trigger/gate mode: 2 to memory length During free run mode: 1 to memory length
* For the record length range, see the description of the Record Length command.	
Number of blocks	During trigger/gate mode: 1 to 256 During free run mode: 1 to memory length
Number of acquisitions	Data length per block > memory length/2: 1 Data length per block ≤ memory length/2: 1 to ∞ (specify 0 to mean ∞)

Valid Common Measurement Control API

API	Valid?	Description
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is raw data (A/D data) in 16-bit signed integer format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is a physical value in double-precision real format.
WeGetAcqDataEx	No	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?	Description
WE_EV_MEASEND	Yes	
WE_EV_BLOCKEND	Yes	
WE_EV_MEASABORT	Yes	
WE_EV_TRIG_HIGH	No	
WE_EV_TRIG_LOW	No	
WE_EV_TRIG_PULSE	No	
WE_EV_CLOSE_MODULE_GUI	Yes	

Module-specific Events

None

Module-specific Error Codes

None

8.12 WE7241 10-CH Digital Thermometer

ASCII Commands

ASCII Command	Description
Sampling Interval	Sets the sampling interval or queries the current setting.
Misc:Burn Out	Sets the burn out or queries the current setting.
Misc:RJC:RJC	Enables/Disables RJC or gets the current setting or queries the current setting.
Misc:RJC	Sets the external temperature of the RJC or queries the current setting.
Misc:RJC:RJC Source	Sets the RJC source or queries the current setting.
Misc:Time Base	Sets the time base or queries the current setting.
Misc:Alarm Combination	Sets the alarm mode or queries the current setting.
Misc:Unit:Channel	Sets the channel for changing the unit of measurement during a temperature measurement or queries the current setting.
Misc:Unit:Channel:All	Sets whether or not to change the unit on all channels during a temperature measurement or queries the current setting.
Misc:Unit	Sets the unit of measurement during temperature measurement or queries the current setting.
Reference Ch	Sets the reference channel or queries the current setting.

CH<x>:

ASCII Command	Description
CH<x>:Range	Sets the range or queries the current setting.
CH<x>:Delta	Sets the difference measurement or queries the current setting.
CH<x>:On	Turns ON/OFF the channel or queries the current setting.
CH<x>:Alarm	Sets the alarm condition or queries the current setting.
CH<x>:Alarm High	Sets the upper limit of the alarm or queries the current setting.
CH<x>:Alarm Low	Sets the lower limit of the alarm or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>.
If two modules are linked and you wish to change the range of channel 9 of the second module, specify "CH19:Range" in the command line.

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

0.2 to 60 s

Example (Visual Basic)

```
' Set the sampling interval to 0.5 s.
ret = WeSetControl (hMo, "Sampling Interval", "0.5")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 500E–03

Misc:Burn Out

Description

Sets the burn out or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Disable burn out.
ret = WeSetControl (hMo, "Misc:Burn Out", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Burn Out", value)
```
value → Off

Misc:RJC:RJC

Description
Enables/Disables RJC or gets the current setting or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Disables the RJC.
ret = WeSetControl (hMo, "Misc:RJC:RJC", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:RJC:RJC", value)
```
value → Off

Misc:RJC

Description
Sets the external temperature of the RJC or queries the current setting.

Parameter
–270.0 to 1820.0°C

Example (Visual Basic)
```
' Set the external temperature of the RJC to 28.1.
ret = WeSetControl (hMo, "Misc:RJC", "28.1")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:RJC:RJC", value)
```
value → 28.1E+00

Misc:RJC:RJC Source

Description
Sets whether to use the external RJC or the internal RJC or queries the current setting.

Parameter
Internal | External

Example (Visual Basic)

```
' Set the RJC to external compensation.
ret = WeSetControl (hMo, "Misc:RJC:RJC Source", "External")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:RJC:RJC Source", value)
```

value → External

Misc:Time Base

Description

Sets the time base or queries the current setting.

Parameter

Internal | BUSCLK

Example (Visual Basic)

```
' Set the time base to internal.
ret = WeSetControl (hMo, "Misc:Time Base", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Time Base", value)
```

value → Internal

Misc:Alarm:Combination

Description

Sets the alarm mode or queries the current setting.

Parameter

AND | OR

Example (Visual Basic)

```
' Set the alarm mode to AND.
ret = WeSetControl (hMo, "Misc:Alarm:Combination", "AND")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Alarm:Combination", value)
```

value → AND

Misc:Unit

Description

Sets the unit of measurement during temperature measurement or queries the current setting.

Parameter

C | K

Example (Visual Basic)

```
' Set the unit to °C.
ret = WeSetControl (hMo, "Misc:Unit", "C")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Unit", value)
```

value → C

Misc:Unit:Channel

Description
Sets the channel for changing the unit of measurement during a temperature measurement or queries the current setting.

Parameter
Slot*-Ch1
where * is 1 to 8

Example (Visual Basic)
```
' Set the channel, on which to change the unit of measurement,
' to Slot2-Ch1.
ret = WeSetControl (hMo, "Misc:Unit:Channel", "Slot2-Ch1")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Unit:Channel", value)
```
value → Slot2-Ch1

Misc:Unit:Channel:All

Description
Sets whether or not to change the unit on all channels during a temperature measurement or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Set the unit to be changed on all channels.
ret = WeSetControl (hMo, "Misc:Unit:Channel:All", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Unit:Channel:All", value)
```
value → On

Reference Ch

Description
Sets the reference channel or queries the current setting.

Parameter
Slot*-Ch1
where * is 1 to 8

Example (Visual Basic)
```
' Set the reference channel to Slot2-Ch1
ret = WeSetControl (hMo, "Reference Ch", "Slot2-Ch1")
Dim value As Variant
ret = WeGetControl (hMo, "Reference Ch", value)
```
value → Slot2-Ch1

CH<x>:Range

Description
Sets the range or queries the current setting.

Parameter

Type K | Type E | Type J | Type T | Type L | Type U | Type N | Type R | Type S | Type B | Type W | KP
50 mV | 100 mV | 200 mV | 500 mV | 1 V | 2 V | 5 V | 10 V | 20 V | 50 V

Example (Visual Basic)

```
' Set the range of CH1 to type T.
ret = WeSetControl (hMo, "CH1:Range", "Type T")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Range", value)
```

value → Type T

CH<x>:Delta

Description

Sets the difference measurement or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Disable difference measurement on CH1.
ret = WeSetControl (hMo, "CH1:Delta", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Delta", value)
```

value → Off

CH<x>:On

Description

Turns ON/OFF the channel or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Disable measurement on CH1.
ret = WeSetControl (hMo, "CH1:On", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:On", value)
```

value → Off

CH<x>:Alarm

Description

Sets the alarm condition or queries the current setting.

Parameter

Rise | Fall | High | Low | In | Out | Off

Example (Visual Basic)

```
' Set the alarm condition to Low.
ret = WeSetControl (hMo, "CH1:Alarm", "Low")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Alarm", value)
```

value → Low

CH<x>:Alarm High

Description

Sets the upper limit of the alarm or queries the current setting.

Parameter

Depends on the range.
For details, see the User' s Manual for the module.

Example (Visual Basic)

```
' Set the upper limit of the alarm to 900.
ret = WeSetControl (hMo, "CH1:Alarm High", "900")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Alarm High", value)
```

value → 900.0E+00

CH<x>:Alarm Low

Description

Sets the lower limit of the alarm or queries the current setting.

Parameter

Depends on the range.
For details, see the User's Manual for the module.

Example (Visual Basic)

```
' Set the lower limit of the alarm to 20.
ret = WeSetControl (hMo, "CH1:Alarm Low", "20")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Alarm Low", value)
```

value → 20.0E+00

Valid Acquisition Modes

Event	Valid?
Triggered	No
Free Run	Yes
Gate(Level)	No
Gate(Edge)	No

Acquisition Restrictions

Items	Possible Settings	Note
Record length	204	
Data length per block	1 to 102	Specify using the WeStartEx parameter.
Number of block	1 to 102	Specify using the WeStartEx parameter.
Number of Acquisition	1 to ∞ (specify 0 for ∞)	Specify using the WeStartEx parameter.

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	No	
WeStartWithEvent	No	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is physical value in single-precision real format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is physical value in single-precision real format.
WeGetAcqDataEx	No	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeLoadPatternDataEx	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?	Event Description
WWE_EV_MEASEND	Yes	
WE_EV_BLOCKEND	Yes	
WE_EV_MEASABORT	Yes	
WE_EV_TRIG_HIGH	Yes	Occurs (1) when the alarm mode is set to In and the measured value enters the High/Low range, (2) when the alarm mode is set to Out and the measured value exits the High/Low range, (3) when the alarm mode is set to High and the measured value exceeds the High value, or (4) when the alarm mode is set to Low and the measured value falls below the Low value.
WE_EV_TRIG_LOW	Yes	Occurs (1) when the alarm mode is set to In and the measured value leaves the High/Low range after entering it, (2) when the alarm mode is set to Out and the measured value enters the High/Low range after exiting it, (3) when the alarm mode is set to High and the measured value falls below the High value after exceeding it, or (4) when the alarm mode is set to Low and the measured value exceeds the Low value after falling below it.
WE_EV_TRIG_PULSE	Yes	Occurs when the alarm mode is set to Rise and the measured value crosses the High value from below the value to above the value or when the alarm mode is set to Fall and the measured value crosses the Low from above the value to below the value.
WE_EV_ALARM	No	
WE_EV_CLOSE_MODULE_GUI	Yes	

Module-specific Events

None

Module-specific Error Codes

None

8.13 WE7245 4-CH, 100 kS/s Strain

ASCII Commands

ASCII Command	Description
Acquisition Mode	Sets the acquisition mode or queries the current setting.
Sampling Interval	Sets the sampling interval or queries the current setting.
Record Length	Sets the record length or queries the current setting.
Memory Partition	Sets the memory partition or queries the current setting.
No. of Acquisitions	Sets the number of acquisitions or queries the current setting.
Linear Scaling	Sets the linear scaling or queries the current setting.
GetX1	Gets the current measured value and stores the value in X1.
GetX2	Gets the current measured value and stores the value in X2.
Balance	Executes the balancing
Balance Status	Queries the results of the balance operation.
Shunt	Performs shunt calibration.

CH<x>:

ASCII Command	Description
CH<x>:On	Turns On/Off the channel or gets the current setting or queries the current setting.
CH<x>:Range Unit	Sets the range unit or queries the current setting.
CH<x>:Range	Sets the range or queries the current setting.
CH<x>:Exciation	Sets the bridge voltage or queries the current setting.
CH<x>:GaugeFactor	Sets the gauge factor or queries the current setting.
CH<x>:Trig Type	Sets the trigger type or queries the current setting.
CH<x>:Trig Level	Sets the trigger level or queries the current setting.
CH<x>:Filter	Sets the input filter or queries the current setting.
CH<x>:Balance Status	Queries the results of the balance operation.
CH<x>:Scale	Turns ON/OFF linear scaling and shunt calibration or queries the current setting.
CH<x>:X1	Sets X1 (when using P1-P2 linear scaling) or queries the current setting.
CH<x>:Y1	Sets Y1 (when using P1-P2 linear scaling) or queries the current setting.
CH<x>:X2	Sets X2 (when using P1-P2 linear scaling) or queries the current setting.
CH<x>:Y2	Sets Y2 (when using P1-P2 linear scaling) or queries the current setting.
CH<x>:A	Sets A (when using Ax+B linear scaling) or queries the current setting.
CH<x>:B	Sets B (when using Ax+B linear scaling) or queries the current setting.
CH<x>:Unit	Sets the linear scale unit string or queries the current setting.

Trig:

ASCII Command	Description
Trig:Trigger:Source	Sets the trigger source or queries the current setting.
Trig:Trigger:Combination	Sets the trigger mode or queries the current setting.
Trig:Trigger:Pretrigger	Sets the amount of pretrigger or queries the current setting.
Trig:Trigger:Hold Off	Sets the trigger hold off or queries the current setting.
Trig:Overlapped Acquisition	Sets the overlap acquisition or queries the current setting.
Trig:Misc:Time Base	Sets the time base or queries the current setting.
Trig:Misc:CH Mode	Sets the channel mode or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>. The number of channels per module is four for the WE7245. If two modules are linked and you wish to change the range of CH2 on the second module, specify "CH6:Range" in the command line.

Acquisition Mode

Description

Sets the acquisition mode or queries the current setting.

Parameter

Triggered | Free Run | Gate(Level) | Gate(Edge)

Example (Visual Basic)

```
' Set the acquisition mode to free run.
ret = WeSetControl (hMo, "Acquisition Mode", "Free Run")
Dim value As Variant
ret = WeGetControl (hMo, "Acquisition Mode", value)
```

value → Free Run

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

0.00001 s to 10 s

Example (Visual Basic)

```
' Set the sampling interval to 10 ms.
ret = WeSetControl (hMo, "Sampling Interval", "10E-3")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 10.00E–03

Record Length

Description

Sets the record length or queries the current setting.

Parameter

2 to 4194304/(number of memory partitions x number of measurement channels) for the trigger mode

1 to 4194304 for the free run mode/number of measurement channels

2 to 4194304 for the gate mode/number of measurement channels

However, the record length range changes so that the measurement period does not fall below 5 ms.

Example: When the sampling interval, the number of measurement channels, and the number of memory partitions are set to 10 ms, 4 CH, and 256, respectively.

500 to 4096 for the trigger mode

500 to 4194304 for the free run mode

500 to 4194304 for the gate mode

Example (Visual Basic)

```
' Set the record length to 1000 points.
ret = WeSetControl (hMo, "Record Length", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "Record Length", value)
```

value → 1000

Memory Partition

Description
Sets the memory partition or queries the current setting.

Parameter
1 to 256 (2^n steps)

Example (Visual Basic)
```
' Set the number of memory partitions to 2.
ret = WeSetControl (hMo, "Memory Partition", "2")
Dim value As Variant
ret = WeGetControl (hMo, "Memory Partition", value)
```
value → 2.0E+00

No. of Acquisitions

Description
Sets the number of acquisitions or queries the current setting.

Parameter
0 to 32768 (0 for ∞)

Example (Visual Basic)
```
' Set the number of acquisitions to 100.
ret = WeSetControl (hMo, "No. of Acquisitions", "100")
Dim value As Variant
ret = WeGetControl (hMo, "No. of Acquisitions", value)
```
value → 100

Linear Scaling

Description
Sets the linear scaling or queries the current setting.

Parameter
off | Ax+B | P1-P2 | Shunt

Example (Visual Basic)
```
' Set the linear scaling to P1-P2.
ret = WeSetControl (hMo, "Linear Scaling", "P1-P2")
Dim value As Variant
ret = WeGetControl (hMo, "Linear Scaling", value)
```
value → P1-P2

GetX1

Description
Gets the current measured value and stores the value in X1 when using the P1-P2 linear scaling.

Parameter
None

Example (Visual Basic)

```
' The Variant variable is not initialized because the parameter is
not present.
Dim value As Variant
' The current measured value is acquired in X1.
ret = WeGetControl (hMo, "GetX1", value)
```

GetX2

Description

Gets the current measured value and stores the value in X2 when using the P1-P2 linear scaling.

Parameter

None

Example (Visual Basic)

```
' The Variant variable is not initialized because the parameter is
not present.
Dim value As Variant
' The current measured value is acquired in X2.
ret = WeGetControl (hMo, "GetX2", value)
```

Balance

Description

Executes balancing. Balancing is performed on all channels that have the measurement range set to a strain range.

Parameter

None

Example (Visual Basic)

```
' The Variant variable is not initialized because the parameter is
not present.
Dim value As Variant
' Execute the balance.
ret = WeGetControl (hMo, "Balance", value)
```

Balance Status

Description

Queries the results of the balance operation. The result is returned for all channels that have the measurement range set to a strain range. To query the result of a specific channel, use the CH<x>:Balance Status command.

Parameter

OFF(Balancing complete) | ON(Balancing failed)

Example (Visual Basic)

```
' Queries the results of the balance operation.
Dim value As Variant
ret = WeGetControl (hMo, "BalanceStatus", value)
```

value → ON

Shunt

Description

Performs shunt calibration.

Parameter

None

Example (Visual Basic)

```
' The Variant variable is not initialized because the parameter is
not present.
Dim value As Variant
' Perform the shunt calibration.
ret = WeGetControl (hMo, "shunt", value)
```

CH<x>:On

Description

Turns On/Off the channel or gets the current setting or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable acquisition on CH1.
ret = WeSetControl (hMo, "CH1:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:On", value)
```

value → On

CH<x>:Range Unit

Description

Sets the range unit or queries the current setting.

Parameter

uSTR | mV/V | V

Example (Visual Basic)

```
' Set CH1's range unit to uSTR.
ret = WeSetControl (hMo, "CH1:RangeUnit", "uSTR")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:RangeUnit", value)
```

value → uSTR

CH<x>:Range

Description
Sets the range or queries the current setting.

Parameter
When the RangeUnit is set to uSTR:
1000 uSTR | 2000 uSTR | 5000 uSTR | 10000 uSTR | 20000 uSTR
When the RangeUnit is set to mV/V:
0.5 mV/V | 1 mV/V | 2.5 mV/V | 5 mV/V | 10 mV/V
When the RangeUnit is set to V:
100 mV | 200 mV | 500 mV | 1 V | 2 V | 5 V | 10 V | 20 V

Example (Visual Basic)
```
' Set CH1's range to 10000 uSTR.
ret = WeSetControl (hMo, "CH1:Range", "10000 uSTR")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Range", value)
```
value → 10000 uSTR

CH<x>:Excitation

Description
Sets the bridge voltage or queries the current setting.

Parameter
When the RangeUnit is set to uSTR or mV/V:
2 V | 5 V | 10 V
When the RangeUnit is set to V:
2 V | 5 V | 10 V | OFF

Example (Visual Basic)
```
' Set CH1's bridge voltage to 2 V.
ret = WeSetControl (hMo, "CH1:Excitation", "2 V")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Excitation", value)
```
value → 2.0E+00

CH<x>:Gauge Factor

Description
Sets the gauge factor or queries the current setting.

Parameter
1.800 to 2.300 (0.001 step)

Example (Visual Basic)
```
' Set CH1's gauge factor to 2.000.
ret = WeSetControl (hMo, "CH1:Gauge Factor", "2.000")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Gauge Factor", value)
```
value → 2.000E+00

CH<x>:Trig Type

Description

Sets the trigger type or queries the current setting.

Parameter

Rise | Fall | Both | High | Low | Off

Example (Visual Basic)

```
' Set CH1's trigger type to Fall.
ret = WeSetControl (hMo, "CH1:Trig Type", "Fall")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Type", value)
```

value → Fall

CH<x>:Trig Level

Description

Sets the trigger level or queries the current setting.

Parameter

When the RangeUnit is set to uSTR:
- –1000 uSTR to 1000 uSTR for 1000 uSTR range
- –2000 uSTR to 2000 uSTR for 2000 uSTR range
- –5000 uSTR to 5000 uSTR for 5000 uSTR range
- –10000 uSTR to 10000 uSTR for 10000 uSTR range
- –20000 uSTR to 20000 uSTR for 20000 uSTR range

When the RangeUnit is set to mV/V:
- –0.5 mV/V to 0.5 mV/V for 0.5 mV/V range
- –1 mV/V to 1 mV/V for 1 mV/V range
- –2.5 mV/V to 2.5 mV/V for 2.5 mV/V range
- –5 mV/V to 5 mV/V for 5 mV/V range
- –10 mV/V to 10 mV/V for 10 mV/V range

When the RangeUnit is set to V:
- –0.1 V to 0.1 V for 100 mV range
- –0.2 V to 0.2 V for 200 mV range
- –0.5 V to 0.5 V for 500 mV range
- –1 V to 1 V for 1 V range
- –2 V to 2 V for 2 V range
- –5 V to 5 V for 5 V range
- –10 V to 10 V for 10 V range
- –20 V to 20 V for 20 V range

Example (Visual Basic)

```
' Set the trigger level to 1 V.
ret = WeSetControl (hMo, "CH1:Trig Level", "1")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Level", value)
```

value → 1.0E+00

CH<x>:Filter

Description
Sets the frequency of tne input filter or queries the current setting.

Parameter
Off | 10 Hz | 30 Hz | 100 Hz | 300 Hz | 1 kHz | 3 kHz | 10 kHz

Example (Visual Basic)
```
' Set CH1 input filter frequency to 10 Hz.
ret = WeSetControl (hMo, "CH1:Filter", "10Hz")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Filter", value)
```
value → 10.0E+00

CH<x>:Balance Status

Description
Queries the results of the balance operation for each channel.

Parameter
0 (Balancing complete) | 1 (Balancing failed)

Example (Visual Basic)
```
' Queries the results of the balance operation for CH1.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:BalanceStatus", value)
```
value → 0

CH<x>:Scale

Description
Sets the linear scaling and shunt calibration's ON/OFF or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Set the linear scaling and shunt calibration to ON.
ret = WeGetControl (hMo, "CH1:Scale", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Scale", value)
```
value → On

CH<x>:X1

Description
Sets the X1 value when using P1-P2 linear scaling or queries the current setting.

Parameter
–1E30 to 1E30

Example (Visual Basic)

```
' Set CH1's X1 to 1.0.
ret = WeGetControl (hMo, "CH1:X1", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:X1", value)
value → 1.0E+000
```

CH<x>:Y1

Description

Sets the Y1 value when using P1-P2 linear scaling or queries the current setting.

Parameter

–1E30 to 1E30

Example (Visual Basic)

```
' Set CH1's Y1 to 1.0.
ret = WeGetControl (hMo, "CH1:Y1", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Y1", value)
value → 1.0E+000
```

CH<x>:X2

Description

Sets the X2 value when using P1-P2 linear scaling or queries the current setting.

Parameter

–1E30 to 1E30

Example (Visual Basic)

```
' Set CH1's X2 to 1.0.
ret = WeGetControl (hMo, "CH1:X2", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:X2", value)
value → 1.0E+000
```

CH<x>:Y2

Description

Sets the Y2 value when using P1-P2 linear scaling or queries the current setting.

Parameter

–1E30 to 1E30

Example (Visual Basic)

```
' Set CH1's Y2 to 1.0.
ret = WeGetControl (hMo, "CH1:Y2", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Y2", value)
value → 1.0E+000
```

CH<x>:A

Description
Sets the A value when using Ax+B linear scaling or queries the current setting.

Parameter
–1E30 to 1E30

Example (Visual Basic)
```
' Set CH1's A to 1.0.
ret = WeGetControl (hMo, "CH1:A", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:A", value)
```
value → 1.0E+000

CH<x>:B

Description
Sets the B value when using Ax+B linear scaling or queries the current setting.

Parameter
–1E30 to 1E30

Example (Visual Basic)
```
' Set CH1's B to 1.0.
ret = WeGetControl (hMo, "CH1:B", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:B", value)
```
value → 1.0E+000

CH<x>:Unit

Description
Sets the unit string for linear scaling or queries the current setting.

Parameter
Strings cannot consist of more than 16 characters

Example (Visual Basic)
```
' Set CH1's unit string to uSTR.
ret = WeGetControl (hMo, "CH1:Unit", "uSTR")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Unit", value)
```
value → uSTR

Trig:Trigger:Source

Description
Sets the trigger source or queries the current setting.

Parameter
Internal | BUSTRG

Example (Visual Basic)

```
' Set the trigger source to internal.
ret = WeSetControl (hMo, "Trig:Trigger:Source", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Source", value)
```

value → Internal

Trig:Trigger:Combination

Description

Sets the trigger mode or queries the current setting.

Parameter

AND | OR

Example (Visual Basic)

```
' Set the trigger mode to AND.
ret = WeSetControl (hMo, "Trig:Trigger:Combination", "AND")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Combination", value)
```

value → AND

Trig:Trigger:Pretrigger

Description

Sets the amount of pretrigger or queries the current setting.

Parameter

0 to the specified record length –2

Example (Visual Basic)

```
' Set the amount of pretrigger to 100.
ret = WeSetControl (hMo, "Trig:Trigger:Pretrigger", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Pretrigger", value)
```

value → 100

Trig:Trigger:Hold Off

Description

Sets the trigger hold off or queries the current setting.

Parameter

- When the acquisition mode is set to trigger mode and overlapped acquisition is disabled: Specified record length to 4194304
- When the acquisition mode is set to trigger mode and overlapped acquisition is enabled: 1 to 4194304
- When the acquisition mode is set to free run mode or gate mode (edge): 1 to 4194304

Example (Visual Basic)

```
' Set the hold off time to 100.
ret = WeSetControl (hMo, "Trig:Trigger:Hold Off", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Hold Off", value)
```

value → 100

Trig:Overlapped Acquisition

Description

Sets the overlap acquisition or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Allow overlap acquisition.
ret = WeSetControl (hMo, "Trig:Overlapped Acquisition", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Overlapped Acquisition", value)
```

value → On

Trig:Misc:Time Base

Description

Sets the time base or queries the current setting.

Parameter

Internal | BUSCLK

Example (Visual Basic)

```
' Set the time base to internal.
ret = WeSetControl (hMo, "Trig:Misc:Time Base", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:Time Base", value)
```

value → Internal

Trig:Misc:CH Mode

Description

Sets the channel mode or queries the current setting.

Parameter

1CH | 2CH | 4CH

Example (Visual Basic)

```
' Set the channel mode to four channel.
ret = WeSetControl (hMo, "Trig:Misc:CH Mode", "4CH")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:CH Mode", value)
```

value → 4CH

Valid Acquisition Modes

Event	Valid?
Triggered	Yes
Free Run	Yes
Gate(Level)	Yes
Gate(Edge)	Yes

Acquisition Restrictions

Items	Possible Settings
Record length	4194304/N (where N is the number of measurement channels (1, 2, or 4).
Data length per block	During trigger mode: 2 to the memory length/memory partition During free run mode: 1 to the memory length
Number of block	During trigger/gate mode: 1 to 256 During free run mode: 1 to the memory length
Number of Acquisition	Data length per block > memory length/2: 1 Data length per block ≤ memory length/2: 1 to ∞ (specify 0 for ∞)

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is raw data in 16-bit signed integer format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is voltage in double-precision real format.
WeGetAcqDataEx	Yes	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?
WE_EV_MEASEND	Yes
WE_EV_BLOCKEND	Yes
WE_EV_MEASABORT	Yes
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

Error Code	Description
0×1801	Failed to balance the bridge.
0×1802	Failed to get the measured value (for reasons such as measurement not in progress (stopped)).

8.14 WE7251 10-CH, 100 kS/s Digitizer

ASCII Commands

ASCII Command	Description
Acquisition Mode	Sets the acquisition mode or queries the current setting.
Sampling Interval	Sets the sampling interval or queries the current setting.
Record Length	Sets the record length or queries the current setting.
Filter 1 kHz	Sets the 1-kHz filter or queries the current setting.
Memory Partition	Sets the memory partition or queries the current setting.
No. of Acquisitions	Sets the number of acquisitions or queries the current setting.

CH<x>:

ASCII Command	Description
CH<x>:On	Turns On/Off the channel or gets the current setting
CH<x>:Range	Sets the channel's input range or queries the current setting.
CH<x>:Trig Type	Sets the channel's trigger type or queries the current setting.
CH<x>:Trig Low	Sets the channel's trigger Low level or queries the current setting.
CH<x>:Trig High	Sets the channel's trigger High level or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>.
If two modules are linked and you wish to change the range of channel 9 of the second module, specify "CH19:Range" in the command line.

Trig:

ASCII Command	Description
Trig:Source	Sets the trigger source or queries the current setting.
Trig:Combination	Sets the trigger mode or queries the current setting.
Trig:Pretrigger	Sets the amount of pretrigger or queries the current setting.
Trig:Hold Off	Sets the hold off time or queries the current setting.
Trig:Misc:Time Base	Sets the time base or queries the current setting.
Trig:Overlapped Acquisition	Sets the overlap acquisition or queries the current setting.

Acquisition Mode

Description

Sets the acquisition mode or queries the current setting.

Parameter

Triggered | Free Run | Gate

Example (Visual Basic)

```
' Set the acquisition mode to Gate.
ret = WeSetControl (hMo, "Acquisition Mode", "Gate")
Dim value As Variant
ret = WeGetControl (hMo, "Acquisition Mode", value)
```

value → Gate

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

0.00001 s to 10 s

Example (Visual Basic)

```
' Set the sampling interval to 10 ms.
ret = WeSetControl (hMo, "Sampling Interval", "10E-3")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 10.00E–03

Record Length

Description

Sets the record length or queries the current setting.

Parameter

- When modules are not linked
  100 to 1048576/(number of memory partitions x number of channels) for the trigger mode
  1 to 1048576 for the free run mode
  2 to 1048576 for the gate mode
- When modules are linked
  100 to 104857/number of memory partitions for the trigger mode
  1 to 104857 for the free run mode
  2 to 104857 for the gate mode

Example (Visual Basic)

```
' Set the record length to 100 points.
ret = WeSetControl (hMo, "Record Length", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Record Length", value)
```

value → 100

Filter 1kHz

Description

Sets the filter or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Disable the 1-kHz filter.
ret = WeSetControl (hMo, "Filter 1 kHz", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "Filter 1 kHz", value)
```

value → Off

Memory Partition

Description

Sets the number of memory partitions or queries the current setting.

Parameter

1 to 256

Example (Visual Basic)

```
' Set the number of memory partitions to 2.
ret = WeSetControl (hMo, "Memory Partition", "2")
Dim value As Variant
ret = WeGetControl (hMo, "Memory Partition", value)
```

value → 2.0E+00

No. of Acquisitions

Description

Sets the number of acquisitions or queries the current setting.

Parameter

0 to 32768

Example (Visual Basic)

```
' Set the number of acquisitions to 100.
ret = WeSetControl (hMo, "No. of Acquisitions", "100")
Dim value As Variant
ret = WeGetControl (hMo, "No. of Acquisitions", value)
```

value → 100

CH<x>:On

Description

Turns On/Off the channel or gets the current setting or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable acquisition on CH1.
ret = WeSetControl (hMo, "CH1:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:On", value)
```

value → On

CH<x>:Range

Description

Sets the channel' s measurement range or queries the current setting.

Parameter

1 V | 2 V | 5 V | 10 V | 20 V

Example (Visual Basic)

```
' Set CH1's range to 20 V.
ret = WeSetControl (hMo, "CH<x>:Range", "20V")
Dim value As Variant
ret = WeGetControl (hMo, "CH<x>:Range", value)
```
value → 20.0E+00

CH<x>:Trig Type

Description

Sets the trigger type or queries the current setting.

Parameter

Rise | Fall | Both | High | Low | In | Out | Off

Example (Visual Basic)

```
' Set CH1's trigger type to In.
ret = WeSetControl (hMo, "CH1:Trig Type", "In")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Type", value)
```
value → In

CH<x>:Trig Low

Description

Sets the trigger Low level or queries the current setting.

Parameter

–20 to 20

Example (Visual Basic)

```
' Set CH1's trigger Low level to 1 V.
ret = WeSetControl (hMo, "CH1:Trig Low", "1")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Low", value)
```
value → 1.0E+00

CH<x>:Trig High

Description

Sets the trigger High level or queries the current setting.

Parameter

–20 to 20

Example (Visual Basic)

```
' Set CH1's trigger High level to 2 V.
ret = WeSetControl (hMo, "CH1:Trig High", "2")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig High", value)
```
value → 2.0+00

Trig:Source

Description
Sets the trigger source or queries the current setting.

Parameter
Internal | BUSTRG

Example (Visual Basic)
```
' Set the trigger source to internal.
ret = WeSetControl (hMo, "Trig:Source", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Source", value)
```
value → Internal

Trig:Combination

Description
Sets the trigger mode or queries the current setting.

Parameter
AND | OR

Example (Visual Basic)
```
' Set the trigger mode to AND.
ret = WeSetControl (hMo, "Trig:Combination", "AND")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Combination", value)
```
value → AND

Trig:Pretrigger

Description
Sets the amount of pretrigger or queries the current setting.

Parameter
0 to the specified record length – 2

Example (Visual Basic)
```
' Set the amount of pretrigger to 100.
ret = WeSetControl (hMo, "Trig:Pretrigger", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Pretrigger", value)
```
value → 100

Trig:Hold Off

Description
Sets the hold off time or queries the current setting.

Parameter

- When the acquisition mode is set to trigger mode and overlapped acquisition is disabled: Specified record length to 1048576
- When the acquisition mode is set to trigger mode and overlapped acquisition is enabled: 1 to 1048576
- When the acquisition mode is set to free run mode: 1 to 1048576

Example (Visual Basic)

```
' Set the hold off time to 100.
ret = WeSetControl (hMo, "Trig:Hold Off", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Hold Off", value)
```

value → 100

Trig:Misc:Time Base

Description

Sets the time base or queries the current setting.

Parameter

Internal | BUSCLK

Example (Visual Basic)

```
' Set the time base to BUSCLK.
ret = WeSetControl (hMo, "Trig:Misc:Time Base", "BUSCLK")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:Time Base", value)
```

value → BUSCLK

Trig:Overlapped Acquisition

Description

Sets the overlap acquisition or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Allow overlap acquisition.
ret = WeSetControl (hMo, "Trig:Overlapped Acquisition", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Overlapped Acquisition", value)
```

value → On

Valid Acquisition Modes

Event	Valid?
Triggered	Yes
Free Run	Yes
Gate(Level)	Yes
Gate(Edge)	No

Acquisition Restrictions

Items	Possible Settings
Record length	1048576/N (where N is the number of measurement channels. The value is 10 when the modules are linked.)
Data length per block	During trigger mode: 100 to the memory length During gate mode: 2 to the memory length During free run mode: 1 to the memory length
Number of block	During trigger/gate mode: 1 to 256, during free run mode: 1 to the memory length
Number of Acquisition	Data length per block > memory length/2: 1 Data length per block ≤ memory length/2: 1 to ∞ (specify 0 for ∞)

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is raw data in 16-bit signed integer format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is voltage in double-precision real format.
WeGetAcqDataEx	Yes	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeLoadPatternDataEx	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?
WWE_EV_MEASEND	Yes
WE_EV_BLOCKEND	Yes
WE_EV_MEASABORT	Yes
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

None

8.15 WE7261/WE7262 32-Bit Digital I/O

ASCII Commands

I/O:

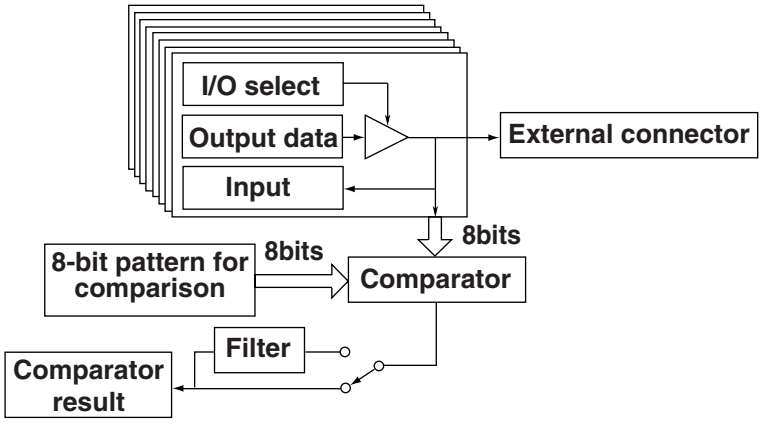
ASCII Command	Description
I/O:I/O Select:Port1	Sets the I/O setting of Port 1 or queries the current setting.
I/O:I/O Select:Port2	Sets the I/O setting of Port 2 or queries the current setting.
I/O:Output:Port1	Sets the output pattern of Port 1 or queries the current setting.
I/O:Output:Port2	Sets the output pattern of Port 2 or queries the current setting.
I/O:Input:Port1	Queries the input pattern of Port 1.
I/O:Input:Port2	Queries the input pattern of Port 2.
I/O:Update	Sets the hardware according to the registered data or queries the current setting.
I/O:Sampling Interval	Sets the sampling interval or queries the current setting.
I/O:Repeat	Sets the repeat operation or queries the current setting.
I/O Select	Sets the I/O setting or queries the current setting.
Output	Sets the Output or queries the current setting.
Input	Queries the Input.

Cntr:

ASCII Command	Description
Cntr:Gate Mode	Sets the gate mode or queries the current setting.
Cntr:Gate Time	Sets the gate time or queries the current setting.
Cntr:Counter:Port1	Queries the Port 1 counter.
Cntr:Counter:Port2	Queries the Port 2 counter.
Cntr:Counter:Port1:Over	Queries the over range condition of the Port 1 counter.
Cntr:Counter:Port2:Over	Queries the over range condition of the Port 2 counter.
Cntr:Start	Starts the counter.
Cntr:Stop	Stops the counter.

Comparator:

These are commands for the comparator functions that are only controllable through the API. The structure of the DIO comparator is 4 sets of 8 bits. It monitors the input and output states independent of the sampling timing. There are no "don't care" states.



ASCII Command	Description
Filter	Sets the filter or queries the current setting.
Comparator Sw	Sets the comparator switch (On/Off) or queries the current setting.
Comparator	Sets the comparator or queries the current setting.
Comparator:Port1:Upper	Queries the result of the Port 1 upper comparator.
Comparator:Port1:Lower	Queries the result of the Port 1 lower comparator.
Comparator:Port2:Upper	Queries the result of the Port 2 upper comparator.
Comparator:Port2:Lower	Queries the result of the Port 2 lower comparator.

Events:

An appropriate event is generated at the time at which the comparator of the upper or lower 8 bits of port 1 and the upper or lower 8 bits of port 2 detects a pattern match.

Trigger Output:

A trigger is generated on the bus trigger line at the time at which the comparator of the upper or lower 8 bits of port 1 and the upper or lower 8 bits of port 2 detects a pattern match.

I/O:I/O Select:Port1

Description

Sets the I/O setting of Port 1 or queries the current setting.

Parameter

0 to 65535(0xffff)

Note:

Issuing the I/O:Update command sets the hardware.

Example (Visual Basic)

```
' Set the I/O setting of Port 1 to 1000.
ret = WeSetControl (hMo, "I/O:I/O Select:Port1", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "I/O:I/O Select:Port1", value)
```

value → 1000

I/O:I/O Select:Port2

Description

Sets the I/O setting of Port 2 or queries the current setting.

Parameter

0 to 65535(0xffff)

Note:

Issuing the I/O:Update command sets the hardware.

Example (Visual Basic)

```
' Set the I/O setting of Port 2 to 1000.
ret = WeSetControl (hMo, "I/O:I/O Select:Port2", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "I/O:I/O Select:Port2", value)
```

value → 1000

I/O:Output:Port1

Description

Sets the output pattern of Port 1 or queries the current setting.

Parameter

0 to 65535(0xffff)

Note:

Issuing the I/O:Update command sets the hardware.

Example (Visual Basic)

```
' Set the output pattern of Port 1 to 1000.
ret = WeSetControl (hMo, "I/O:Output:Port1", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "I/O:Output:Port1", value)
```

value → 1000

I/O:Output:Port2

Description

Sets the output pattern of Port 2 or queries the current setting.

Parameter

0 to 65535(0xffff)

Note:

Issuing the I/O:Update command sets the hardware.

Example (Visual Basic)

```
' Set the output pattern of Port 2 to 1000.
ret = WeSetControl (hMo, "I/O:Output:Port2", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "I/O:Output:Port2", value)
```

value → 1000

I/O:Input:Port1

Description

Queries the input pattern of Port 1.

Parameter

0 to 65535(0xffff)

Example (Visual Basic)

```
Dim value As Variant
' Query the input pattern of Port 1.
ret = WeGetControl (hMo, "I/O:Input:Port1", value)
```

value → 1000

I/O:Input:Port2

Description

Queries the input pattern of Port 2.

Parameter

0 to 65535(0xffff)

Example (Visual Basic)

```
Dim value As Variant
' Query the input pattern of Port 2.
ret = WeGetControl (hMo, "I/O:Input:Port2", value)
```

value → 1000

I/O:Update

Description

Sets the hardware according to the registered data or queries the current setting.

Parameter

None

Example (Visual Basic)

```
Dim value As Variant
ret = WeSetControl (hMo, "I/O:Update", value)
```

I/O:Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

0.01 to 10.00

Example (Visual Basic)

```
' Set the sampling interval to 5 s.
ret = WeSetControl (hMo, "I/O:Sampling Interval", "5.0")
Dim value As Variant
ret = WeGetControl (hMo, "I/O:Sampling Interval", value)
```

value → 5.00E+00

I/O:Repeat

Description

Sets the repeat operation or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable the I/O repeat operation.
ret = WeSetControl (hMo, "I/O:Repeat", "On")
Dim value As Variant
ret = WeGetControl (hMo, "I/O:Repeat", value)
```

value → On

I/O Select

Description

Sets the I/O Select or queries the current setting.

Parameter

WE_ULONG = 0 to 4294967295(0xffffffff)

Note:

Directly sets the hardware. Sets Port 1 and Port 2 simultaneously.

Example (Visual Basic)

```
' Set the I/O Select to 1000.
ret = WeSetControl (hMo, "I/O Select", 1000)
Dim value As Variant
ret = WeGetControl (hMo, "I/O Select", value)
```

value → 1000

Output

Description

Sets the Output or queries the current setting.

Parameter

WE_ULONG = 0 to 4294967295(0xffffffff)

Note:

Directly sets the hardware. Sets Port 1 and Port 2 simultaneously.

Example (Visual Basic)

```
' Sets the Output to 1000.
ret = WeSetControl (hMo, "Output", 1000)
Dim value As Variant
ret = WeGetControl (hMo, "Output", value)
```

value → 1000

Input

Description

Queries the Input.

Parameter

WE_ULONG = 0 to 4294967295(0xffffffff)

Example (Visual Basic)

```
Dim value As Variant
ret = WeGetControl (hMo, "Input", value)
```

value → 1000

Cntr:Gate Mode

Description

Sets the gate mode or queries the current setting.

Parameter

Manual | Timer | External

Example (Visual Basic)

```
' Set the gate mode to timer.
ret = WeSetControl (hMo, "Cntr:Gate Mode", "Timer")
Dim value As Variant
ret = WeGetControl (hMo, "Cntr:Gate Mode", value)
```

value → Timer

Cntr:Gate Time

Description

Sets the gate time or queries the current setting.

Parameter

0.001 s to 600.0000 s

Example (Visual Basic)

```
' Set the gate time to 3 s.
ret = WeSetControl (hMo, "Cntr:Gate Time", "3.0")
Dim value As Variant
ret = WeGetControl (hMo, "Cntr:Gate Time", value)
```

value → 3.0000E+00

Cntr:Counter:Port1

Description

Queries the Port 1 counter.

Parameter

0 to 0xFFFE0000

Example (Visual Basic)

```
Dim value As Variant
ret = WeGetControl (hMo, "Cntr:Counter:Port1", value)
```

value → 1000

Cntr:Counter:Port2

Description

Queries the Port 2 counter.

Parameter

0 to 0xFFFE0000

Example (Visual Basic)

```
Dim value As Variant
ret = WeGetControl (hMo, "Cntr:Counter:Port2", value)
```

value → 1000

Cntr:Counter:Port1:Over

Description
Queries the over range condition of the Port1 counter.

Parameter
Off | On

Example (Visual Basic)
```
Dim value As Variant
ret = WeGetControl (hMo, "Cntr:Counter:Port1:Over", value)
```
value → On

Cntr:Counter:Port2:Over

Description
Queries the over range condition of the Port 2 counter.

Parameter
Off | On

Example (Visual Basic)
```
Dim value As Variant
ret = WeGetControl (hMo, "Cntr:Counter:Port2:Over", value)
```
value → On

Cntr:Start

Description
Starts the counter.

Parameter
None

Example (Visual Basic)
```
Dim value As Variant
ret = WeSetControl (hMo, "Cntr:Start", value)
```

Cntr:Stop

Description
Stops the counter.

Parameter
None

Example (Visual Basic)
```
Dim value As Variant
ret = WeSetControl (hMo, "Cntr:Stop", value)
```

Filter

Description
Sets the chattering elimination filter or queries the current setting.

Parameter
Off | On

Note:
Eliminates chattering noise during pattern matching. When turned on, signal components with a period less than 1 ms are eliminated.

Example (Visual Basic)
```
' Enable the chattering elimination filter.
ret = WeSetControl (hMo, "Filter", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Filter", value)
```
value → On

Comparator Sw

Description
Sets the comparator switch or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Enable the comparator switch.
ret = WeSetControl (hMo, "Comparator Sw", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Comparator Sw", value)
```
value → On

Comparator

Description
Sets the comparator or queries the current setting.

Parameter
WE_ULONG = 0 to 4294967295(0xffffffff)

24 to 31	Port1 Upper
16 to 23	Port1 Lower
8 to 15	Port2 Upper
0 to 7	Port2 Lower

Example (Visual Basic)
```
' Set the comparator to 1000.
ret = WeSetControl (hMo, "Comparator", 1000)
Dim value As Variant
ret = WeGetControl (hMo, "Comparator", value)
```
value → 1000

Comparator:Port1:Upper

Description
Queries the result of the Port 1 upper comparator.

Parameter
Off | On

Example (Visual Basic)
```
Dim value As Variant
ret = WeGetControl (hMo, "Comparator:Port1:Upper", value)
```
value → Off

Comparator:Port1:Lower

Description
Queries the result of the Port 1 lower comparator.

Parameter
Off | On

Example (Visual Basic)
```
Dim value As Variant
ret = WeGetControl (hMo, "Comparator:Port1:Lower", value)
```
value → Off

Comparator:Port2:Upper

Description
Queries the result of the Port 2 upper comparator.

Parameter
Off | On

Example (Visual Basic)
```
Dim value As Variant
ret = WeGetControl (hMo, "Comparator:Port2:Upper", value)
```
value → Off

Comparator:Port2:Lower

Description
Queries the result of the Port 2 lower comparator.

Parameter
Off | On

Example (Visual Basic)
```
Dim value As Variant
ret = WeGetControl (hMo, "Comparator:Port2:Lower", value)
```
value → Off

Valid Acquisition Modes

Event	Valid?	Note
Triggered	Yes	For the counter function
Free Run	Yes	For the I/O function
Gate(Level)	No	
Gate(Edge)	No	

Acquisition Restrictions

Items	Possible Settings	Note
Record length	8192	
Data length per block	1 to 8192	Specify using the WeStartEx parameter.
Number of block	1 to 8192	Specify using the WeStartEx parameter.
Number of Acquisition	1 to ∞ (specify 0 for ∞)	Specify using the WeStartEx parameter.

* Data length per block > memory length/2: 1
Data length per block ≤ memory length/2: 1 to ∞

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	Valid for the I/O function.
WeStop	Yes	Valid for the I/O function.
WeStartSingle	Yes	Valid for the I/O function.
WeStartWithEvent	Yes	Valid for the I/O function.
WeIsRun	Yes	Valid for the counter function. For the I/O function, use the I/O:IsRun ASCII command.
WeLatchData	Yes	Valid for the I/O function.
WeGetAcqDataInfo	Yes	Valid for the I/O function.
WeGetAcqData	Yes	Valid for the I/O function. Data are bit data of 2-byte length.
WeGetScaleData	No	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	Valid for the I/O function.
WeSaveScaleData	No	
WeSaveAsciiData	No	
WeGetCurrentData	No	
WeGetAcqDataEx	No	
WeGetAcqDataSize	Yes	Valid for the I/O function.
WeSaveAcqHeader	Yes	Valid for the I/O function.
WeSavePatternData	No	
WeLoadPatternData	No	
WeLoadPatternDataEx	No	
WeStartEx	Yes	Valid for the I/O function.
WeStopEx	Yes	Valid for the I/O function.

Valid Common Events

Event	Valid?
WWE_EV_MEASEND	Yes
WE_EV_BLOCKEND	Yes
WE_EV_MEASABORT	Yes
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

Event	Event Description
0×01000000	Port1 counter over range
0×02000000	Port2 counter over range
0×10000000	Port1 lower pattern matched
0×20000000	Port1 upper pattern matched
0×40000000	Port2 lower pattern matched
0×80000000	Port2 upper pattern matched

Module-specific Error Codes

None

8.16 WE7271/WE7272 4-CH, 100 kS/s Isolated Digitizer

Note

When getting the module handle for WE7272 using the WeOpenModule API, specify "WE7271" for the name.

ASCII Commands

ASCII Command	Description
Acquisition Mode	Sets the acquisition mode or queries the current setting.
Sampling Interval	Sets the sampling interval or queries the current setting.
Record Length	Sets the record length or queries the current setting.
Memory Partition	Sets the memory partition or queries the current setting.
No. of Acquisitions	Sets the number of acquisitions or queries the current setting.

CH<x>:

ASCII Command	Description
CH<x>:On	Turns On/Off the channel or gets the current setting or queries the current setting.
CH<x>:Range	Sets the channel' s input range or queries the current setting.
CH<x>:Trig Type	Sets the channel' s trigger type or queries the current setting.
CH<x>:Trig Level	Sets the channel' s trigger level or queries the current setting.
CH<x>:Filter	Sets the channel' s input filter or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>. The number of channels per module is four for the WE7271 or WE7272. If two modules are linked and you wish to change the range of CH3 on the second module, specify "CH7:Range" in the command line.

Trig:

ASCII Command	Description
Trig:Trigger:Source	Sets the trigger source or queries the current setting.
Trig:Trigger:Combination	Sets the trigger mode or queries the current setting.
Trig:Trigger:Pretrigger	Sets the amount of pretrigger or queries the current setting.
Trig:Trigger:Hold Off	Sets the trigger hold off or queries the current setting.
Trig:Overlapped Acquisition	Sets the overlap acquisition or queries the current setting.
Trig:Misc:Time Base	Sets the time base or queries the current setting.
Trig:Misc:CH Mode	Sets the channel mode or queries the current setting.

Acquisition Mode

Description

Sets the acquisition mode or queries the current setting.

Parameter

Triggered | Free Run | Gate(Level) | Gate(Edge)

Example (Visual Basic)

```
' Set the acquisition mode to free run.
ret = WeSetControl (hMo, "Acquisition Mode", "Free Run")
Dim value As Variant
ret = WeGetControl (hMo, "Acquisition Mode", value)
```

value → Free Run

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

0.00001 s to 10 s

Example (Visual Basic)

```
' Set the sampling interval to 10 ms.
ret = WeSetControl (hMo, "Sampling Interval", "10E-3")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 10.00E–03

Record Length

Description

Sets the record length or queries the current setting.

Parameter

2 to 4194304/(number of memory partitions x number of measurement channels) for the trigger mode
1 to 4194304 for the free run mode/number of measurement channels
2 to 4194304 for the gate mode/number of measurement channels

Note:

For the minimum record length, "the record length x sampling interval" cannot be less than 5 ms. When using the internal clock as the time base, the record length is set to the minimum allowed value when a value below this minimum value is specified. However, this restriction is not enforced when the time base signal of the measuring station (CMNCLK) is used. The value is set to the specified value, but proper measurements may be impeded. The minimum allowed record length is "5 ms/sampling interval." For example, the minimum allowed record length when the sampling interval is "10 μs" is "500."

Example (Visual Basic)

```
' Set the record length to 100 points.
ret = WeSetControl (hMo, "Record Length", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Record Length", value)
```

value → 100

Memory Partition

Description

Sets the memory partition or queries the current setting.

Parameter

1 to 256

Example (Visual Basic)

```
' Set the number of memory partitions to 2.
ret = WeSetControl (hMo, "Memory Partition", "2")
Dim value As Variant
ret = WeGetControl (hMo, "Memory Partition", value)
```
value → 2.0E+00

No. of Acquisitions

Description

Sets the number of acquisitions or queries the current setting.

Parameter

0 to 32768 (0 for ∞)

Example (Visual Basic)

```
' Set the number of acquisitions to 100.
ret = WeSetControl (hMo, "No. of Acquisitions", "100")
Dim value As Variant
ret = WeGetControl (hMo, "No. of Acquisitions", value)
```
value → 100

CH<x>:On

Description

Turns On/Off the channel or gets the current setting or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable acquisition on CH1.
ret = WeSetControl (hMo, "CH1:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:On", value)
```
value → On

CH<x>:Range

Description

Sets the range or queries the current setting.

Parameter

1V | 2V | 5V | 10V | 20V | 35V

Example (Visual Basic)

```
' Set CH1's range to 20 V.
ret = WeSetControl (hMo, "CH1:Range", "20.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Range", value)
```
value → 20.0E+00

CH<x>:Trig Type

Description
Sets the trigger type or queries the current setting.

Parameter
Rise | Fall | Both | High | Low | Off

Example (Visual Basic)
```
' Set CH1's trigger type to Low.
ret = WeSetControl (hMo, "CH1:Trig Type", "Low")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Type", value)
```
value → Low

CH<x>:Trig Level

Description
Sets the trigger level or queries the current setting.

Parameter
–35 to 35

Example (Visual Basic)
```
' Set the trigger level to 1 V.
ret = WeSetControl (hMo, "CH1:Trig Level", "1")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Level", value)
```
value → 1.00E+00

CH<x>:Filter

Description
Sets the input filter or queries the current setting.

Parameter
Off | 500Hz | 5kHz

Example (Visual Basic)
```
' Turn Off CH1's input filter.
ret = WeSetControl (hMo, "CH1:Filter", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Filter", value)
```
value → Off

Trig:Trigger:Source

Description
Sets the trigger source or queries the current setting.

Parameter
Internal | BUSTRG

Example (Visual Basic)

```
' Set the trigger source to internal.
ret = WeSetControl (hMo, "Trig:Trigger:Source", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Source", value)
```

value → Internal

Trig:Trigger:Combination

Description

Sets the trigger mode or queries the current setting.

Parameter

AND | OR

Example (Visual Basic)

```
' Set the trigger mode to AND.
ret = WeSetControl (hMo, "Trig:Trigger:Combination", "AND")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Combination", value)
```

value → AND

Trig:Trigger:Pretrigger

Description

Sets the amount of pretrigger or queries the current setting.

Parameter

0 to the specified record length –2

Example (Visual Basic)

```
' Set the amount of pretrigger to 100.
ret = WeSetControl (hMo, "Trig:Trigger:Pretrigger", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Pretrigger", value)
```

value → 100

Trig:Trigger:Hold Off

Description

Sets the trigger hold off or queries the current setting.

Parameter

- When the acquisition mode is set to trigger mode and overlapped acquisition is disabled: Specified record length to 4194304
- When the acquisition mode is set to trigger mode and overlapped acquisition is enabled: 1 to 4194304
- When the acquisition mode is set to free run mode or gate mode (edge): 1 to 4194304

Example (Visual Basic)

```
' Set the hold off time to 100.
ret = WeSetControl (hMo, "Trig:Trigger:Hold Off", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Hold Off", value)
```

value → 100

Trig:Misc:Time Base

Description

Sets the time base or queries the current setting.

Parameter

Internal | BUSCLK

Example (Visual Basic)

```
' Set the time base to bus clock.
ret = WeSetControl (hMo, "Trig:Misc:Time Base", "BUSCLK")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:Time Base", value)
```

value → BUSCLK

Trig:Misc:CH Mode

Description

Sets the channel mode or queries the current setting.

Parameter

1CH | 2CH | 4CH

Example (Visual Basic)

```
' Set the channel mode to one channel.
ret = WeSetControl (hMo, "Trig:Misc:CH Mode", "1CH")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:CH Mode", value)
```

value → 1CH

Trig:Overlapped Acquisition

Description

Enables or disables overlapped acquisition or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable overlapped acquisition.
ret = WeSetControl (hMo, "Trig:Overlapped Acquisition", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Overlapped Acquisition", value)
```

value → On

Valid Acquisition Modes

Event	Valid?
Triggered	Yes
Free Run	Yes
Gate(Level)	Yes
Gate(Edge)	Yes

Acquisition Restrictions

Items	Possible Settings
Record length	4194304/N (where N is the number of measurement channels (1, 2, or 4).
Data length per block	During trigger mode: 2 to the memory length During free run mode: 1 to the memory length
Number of block	During trigger/gate mode: 1 to 256 During free run mode: 1 to the memory length
Number of Acquisition	Data length per block > memory length/2: 1 Data length per block ≤ memory length/2: 1 to ∞ (specify 0 for ∞)

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is raw data in 16-bit signed integer format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is voltage in double-precision real format.
WeGetAcqDataEx	No	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeLoadPatternDataEx	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?
WWE_EV_MEASEND	Yes
WE_EV_BLOCKEND	Yes
WE_EV_MEASABORT	Yes
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

None

8.17 WE7273 8-CH, 100 kS/s Isolated Digitizer

ASCII Commands

ASCII Command	Description
Acquisition Mode	Sets the acquisition mode or queries the current setting.
Sampling Interval	Sets the sampling interval or queries the current setting.
Record Length	Sets the record length or queries the current setting.
Memory Partition	Sets the memory partition or queries the current setting.
No. of Acquisitions	Sets the number of acquisitions or queries the current setting.

CH<x>:

ASCII Command	Description
CH<x>:On	Turns On/Off the channel or gets the current setting or queries the current setting.
CH<x>:Coupling	Sets the channel's input coupling or queries the current setting.
CH<x>:Range	Sets the channel's measurement range or queries the current setting.
CH<x>:Filter	Sets the channel's input filter or queries the current setting.
CH<x>:Trig Type	Sets the channel's trigger type or queries the current setting.
CH<x>:Trig Level	Sets the channel's trigger level or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>. The number of channels per module is eight for the WE7273. If two modules are linked and you wish to change the range of CH2 on the second module, specify "CH10:Range" in the command line.

Trig:

ASCII Command	Description
Trig:Trigger:Source	Sets the trigger source or queries the current setting.
Trig:Trigger:Combination	Sets the trigger mode or queries the current setting.
Trig:Trigger:Pretrigger	Sets the amount of pretrigger or queries the current setting.
Trig:Trigger:Hold Off	Sets the trigger hold off or queries the current setting.
Trig:Misc:Time Base	Sets the time base or queries the current setting.
Trig:Misc:CH Mode	Sets the channel mode or queries the current setting.
Trig:Overlapped Acquisition	Sets the overlap acquisition or queries the current setting.
Trig:Manual Trigger	Executes a manual trigger.

Acquisition Mode

Description

Sets the acquisition mode or queries the current setting.

Parameter

Triggered | Free Run | Gate(Level) | Gate(Edge)

Example (Visual Basic)

```
' Set the acquisition mode to free run.
ret = WeSetControl (hMo, "Acquisition Mode", "Free Run")
Dim value As Variant
ret = WeGetControl (hMo, "Acquisition Mode", value)
```

value → Free Run

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

0.00001 s to 10 s

Example (Visual Basic)

```
' Set the sampling interval to 10 ms.
ret = WeSetControl (hMo, "Sampling Interval", "10E-3")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 10.000E–03

Record Length

Description

Sets the record length or queries the current setting.

Parameter

The selectable range of the record length differs depending on the acquisition mode that is selected.
2 to 8388608/(number of memory partitions/N*) for the trigger mode
1 to 8388608 for the free run mode/N*
2 to 8388608 for the gate mode/N*
*N = The number of measurement channels specified by channel mode (Trig:Misc:CH Mode)

However, the record length range changes so that the measurement period does not fall below 5 ms.
Example: When the sampling interval, the number of measurement channels, and the number of memory partitions are set to 100 μs, 8 CH, and 256, respectively.
50 to 4096 for the trigger mode
50 to 4194304 for the free run mode
50 to 4194304 for the gate mode

Example (Visual Basic)

```
' Set the record length to 1000 points.
ret = WeSetControl (hMo, "Record Length", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "Record Length", value)
```

value → 1000

Memory Partition

Description

Sets the memory partition or queries the current setting.

Parameter

1 to 256

Example (Visual Basic)

```
' Set the number of memory partitions to 2.
ret = WeSetControl (hMo, "Memory Partition", "2")
Dim value As Variant
ret = WeGetControl (hMo, "Memory Partition", value)
```

value → 2.0E+00

No. of Acquisitions

Description

Sets the number of acquisitions or queries the current setting.

Parameter

0 to 32768 (0 for ∞)

Example (Visual Basic)

```
' Set the number of acquisitions to 100.
ret = WeSetControl (hMo, "No. of Acquisitions", "100")
Dim value As Variant
ret = WeGetControl (hMo, "No. of Acquisitions", value)
```

value → 100

CH<x>:On

Description

Turns On/Off the channel or gets the current setting or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable acquisition on CH1.
ret = WeSetControl (hMo, "CH1:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:On", value)
```

value → On

CH<x>:Coupling

Description

Sets the input coupling of the channel or queries the current setting.

Paramter

DC | AC

Example (Visual Basic)

```
' Set the coupling of CH1 to AC.
ret = WeSetControl (hMo, "CH1:Coupling", "AC")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Coupling", value)
```

value → AC

CH<x>:Range

Description
Sets the range or queries the current setting.

Parameter
50mV | 100mV | 200mV | 500mV | 1V | 2V | 5V | 10V | 20V | 50V

Example (Visual Basic)
```
' Set CH1's range to 20 V.
ret = WeSetControl (hMo, "CH1:Range", "20.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Range", value)
```
value → 20.0E+00

CH<x>:Filter

Description
Sets the input filter or queries the current setting.

Parameter
Off | 50Hz | 500Hz | 5kHz

Example (Visual Basic)
```
' Turn Off CH1's input filter.
ret = WeSetControl (hMo, "CH1:Filter", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Filter", value)
```
value → Off

CH<x>:Trig Type

Description
Sets the trigger type or queries the current setting.

Parameter
When the acquisition mode is set to trigger mode, free run mode, or gate mode (edge)
Rise | Fall | Both | High | Low | Off
When the acquisition mode is set to gate mode (level)
High | Low | Off

Example (Visual Basic)
```
' Set CH1's trigger type to Low.
ret = WeSetControl (hMo, "CH1:Trig Type", "Low")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Type", value)
```
value → Low

CH<x>:Trig Level

Description
Sets the trigger level or queries the current setting.

Parameter
The selectable range of the trigger level differs depending on the specified measurement range.
Measurement range 50 mV: –0.0500 to 0.0500
Measurement range 100 mV: –0.100 to 0.100
Measurement range 200 mV: –0.200 to 0.200
Measurement range 500 mV: –0.500 to 0.500
Measurement range 1 V: –1.00 to 1.00
Measurement range 2 V: –2.00 to 2.00
Measurement range 5 V: –5.00 to 5.00
Measurement range 10 V: –10.0 to 10.0
Measurement range 20 V: –20.0 to 20.0
Measurement range 50 V: –50.0 to 50.0

Example (Visual Basic)
```
' Set the trigger level to 0.5 V.
ret = WeSetControl (hMo, "CH1:Trig Level", "0.5")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Level", value)
```
value → 500E–03

Trig:Trigger:Source

Description
Sets the trigger source or queries the current setting.

Parameter
Internal | BUSTRG

Example (Visual Basic)
```
' Set the trigger source to internal.
ret = WeSetControl (hMo, "Trig:Trigger:Source", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Source", value)
```
value → Internal

Trig:Trigger:Combination

Description
Sets the trigger mode or queries the current setting.

Parameter
AND | OR

Example (Visual Basic)

```
' Set the trigger mode to AND.
ret = WeSetControl (hMo, "Trig:Trigger:Combination", "AND")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Combination", value)
```

value → AND

Trig:Trigger:Pretrigger

Description

Sets the amount of pretrigger or queries the current setting.

Parameter

0 to the specified record length –2

Example (Visual Basic)

```
' Set the amount of pretrigger to 100.
ret = WeSetControl (hMo, "Trig:Trigger:Pretrigger", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Pretrigger", value)
```

value → 100

Trig:Trigger:Hold Off

Description

Sets the trigger hold off or queries the current setting.

Parameter

- When the acquisition mode is set to trigger mode and overlapped acquisition is disabled: Specified record length to 8388608
- When the acquisition mode is set to trigger mode and overlapped acquisition is enabled: 1 to 8388608
- When the acquisition mode is set to free run mode or gate mode (edge): 1 to 8388608

Example (Visual Basic)

```
' Set the hold off time to 1000.
ret = WeSetControl (hMo, "Trig:Trigger:Hold Off", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Trigger:Hold Off", value)
```

value → 1000

Trig:Misc:Time Base

Description

Sets the time base or queries the current setting.

Parameter

Internal | BUSCLK

Example (Visual Basic)

```
' Set the time base to bus clock.
ret = WeSetControl (hMo, "Trig:Misc:Time Base", "BUSCLK")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:Time Base", value)
```

value → BUSCLK

Trig:Misc:CH Mode

Description

Sets the channel mode or queries the current setting.

Parameter

1CH | 2CH | 4CH | 8CH

Note:

The upper limit of record length that can be specified per channel varies depending on the number of channels specified by channel mode. For details, see the explanation of the Record Length command.

Example (Visual Basic)

```
' Set the channel mode to one channel.
ret = WeSetControl (hMo, "Trig:Misc:CH Mode", "1CH")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Misc:CH Mode", value)
```

value → 1CH

Trig:Overlapped Acquisition

Description

Enables or disables overlapped acquisition or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable overlapped acquisition.
ret = WeSetControl (hMo, "Trig:Overlapped Acquisition", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Overlapped Acquisition", value)
```

value → On

Trig:Manual Trigger

Description

Executes a manual trigger.

Parameter

None

Example (Visual Basic)

```
' Execute a manual trigger.
Dim value As Variant
ret = WeSetControl (hMo, "Trig:Manual Trigger", value)
```

Valid Acquisition Modes

Event	Valid?
Triggered	Yes
Free Run	Yes
Gate(Level)	Yes
Gate(Edge)	Yes

Acquisition Restrictions

Items	Possible Settings
Memory length	1048576 (when 8CH mode is selected) 2097152 (when 4CH mode is selected) 4194304 (when 2CH mode is selected) 8388608 (when 1CH mode is selected)
Data length per block	During trigger mode: 2 to memory length/(memory partitions × number of measurement channels) During free run mode: 1 to memory length During gate (level) mode: 2 to memory length * For the selectable range of the data length per block, see the description of the Record Length command.
Number of block	During trigger/gate mode: 1 to 256 During free run mode: 1 to the memory length
Number of Acquisition	Data length per block > memory length/2: 1 Data length per block ≤ memory length/2: 1 to ∞ (specify 0 for ∞)

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is raw data in 16-bit signed integer format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is voltage in double-precision real format.
WeGetAcqDataEx	No	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?
WWE_EV_MEASEND	Yes
WE_EV_BLOCKEND	Yes
WE_EV_MEASABORT	Yes
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	Yes
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

None

8.18 WE7275 2-CH, 1MS/s Isolated Digitizer

ASCII Commands

ASCII Command	Description
Acquisition Mode	Sets the acquisition mode or queries the current setting.
Sampling Interval	Sets the sampling interval or queries the current setting.
Record Length	Sets the record length or queries the current setting.
Memory Partition	Sets the memory partition or queries the current setting.
No. of Acquisitions	Sets the number of acquisitions or queries the current setting.
Time Base	Sets the time base or queries the current setting.
CH Mode	Sets the number of measurement channels or queries the current setting.

CH<x>:

ASCII Command	Description
CH<x>:On	Sets the measurement channel or queries the current setting.
CH<x>:Coupling	Sets the input coupling of the channel or queries the current setting.
CH<x>:Range	Sets the measurement range of the channel or queries the current setting.
CH<x>:Trig Type	Sets the trigger type of the channel or queries the current setting.
CH<x>:Trig Level	Sets the trigger level of the channel or queries the current setting.
CH<x>:AAF	Sets the type of input filter of the channel or queries the current setting.
CH<x>:Filter	Sets the frequency of the input filter of the channel or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>. The number of channels per module is 2 for this module. If two modules are linked, the command used to change the measurement range of channel 2 of the second module is "CH4:Range."

Trig:

ASCII Command	Description
Trig:Source	Sets the trigger source or queries the current setting.
Trig:Combination	Sets the trigger combination or queries the current setting.
Trig:Pretrigger	Sets the amount of pre-trigger or queries the current setting
Trig:Hold Off	Sets the trigger hold-off or queries the current setting.
Trig:Overlapped Acquisition	Sets the overlap acquisition or queries the current setting.

Acquisition Mode

Description

Sets the acquisition mode or queries the current setting.

Parameter

Triggered | Free Run | Gate(Level) | Gate(Edge)

Example (Visual Basic)

```
' Set the acquisition mode to free run.
ret = WeSetControl (hMo, "Acquisition Mode", "Free Run")
Dim value As Variant
ret = WeGetControl (hMo, "Acquisition Mode", value)
```

value → Free Run

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

0.000001s to 1.0 s

Example (Visual Basic)

```
' Set the sampling interval to 10 ms.
ret = WeSetControl (hMo, "Sampling Interval", "10E-3")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 10.00E–03

Record Length

Description

Sets the record length or queries the current setting.

Parameter

2 to 4194304/(number of memory partitions x number of measurement channels) for the trigger mode
2 to 4194304 for the gate mode/number of measurement channels
1 to 4194304 for the free run mode/number of measurement channels

Note:

For the minimum record length, "the record length x sampling interval" cannot be less than 5 ms. When using the internal clock as the time base, the record length is set to the minimum allowed value when a value below this minimum value is specified. However, this restriction is not enforced when the time base signal of the measuring station (CMNCLK) is used. The value is set to the specified value, but proper measurements may be impeded. The minimum allowed record length is "5 ms/sampling interval." For example, the minimum allowed record length when the sampling interval is "10 μs" is "500."

Example (Visual Basic)

```
' Set the record length to 100.
ret = WeSetControl (hMo, "Record Length", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Record Length", value)
```

value → 100

Memory Partition

Description

Sets the memory partition or queries the current setting.

Parameter

1 to 256

Example (Visual Basic)

```
' Set the number of memory partitions to 2.
ret = WeSetControl (hMo, "Memory Partition", "2")
Dim value As Variant
ret = WeGetControl (hMo, "Memory Partition", value)
```

value → 2.0E+00

No. of Acquisitions

Description
Sets the number of acquisitions or queries the current setting.

Parameter
0 to 32768

Note:
The number of acquisitions is set to infinity if "0" is specified.

Example (Visual Basic)
```
' Set the number of acquisitions to 100.
ret = WeSetControl (hMo, "No. of Acquisitions", "100")
Dim value As Variant
ret = WeGetControl (hMo, "No. of Acquisitions", value)
```
value → 100

Time Base

Description
Sets the time base or queries the current setting.

Parameter
Internal | BUSCLK(Slow) | BUSCLK(Fast) | External(Slow) | External(Fast)

Example (Visual Basic)
```
' Set the time base to Internal.
ret = WeSetControl (hMo, "Time Base", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Time Base", value)
```
value → Internal

CH Mode

Description
Sets the number of measurement channels or queries the current setting.

Parameter
1CH | 2CH

Example (Visual Basic)
```
' Set the number of measurement channels to 2.
ret = WeSetControl (hMo, "CH Mode", "2CH")
Dim value As Variant
ret = WeGetControl (hMo, "CH Mode", value)
```
value → 2CH

CH<x>:On

Description
Turns On/Off the channel or gets the current setting.

Parameter
Off | On

Example (Visual Basic)
```
' Turn ON the measurement for CH1.
ret = WeSetControl (hMo, "CH1:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:On", value)
```
value → On

CH<x>:Coupling

Description
Sets the input coupling of the channel or queries the current setting.

Parameter
DC | AC

Example (Visual Basic)
```
' Set the CH1 input coupling to DC.
ret = WeSetControl (hMo, "CH1:Coupling", "DC")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Coupling", value)
```
value → DC

CH<x>:Range

Description
Sets the measurement range of the channel or queries the current setting.

Parameter
100 mV | 200 mV | 500 mV | 1 V | 2 V | 5 V | 10 V | 20V | 50 V | 100 V | 200 V | 350V

Example (Visual Basic)
```
' Set the CH1 measurement range to 5 V.
ret = WeSetControl (hMo, "CH1:Range", "5V")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Range", value)
```
value → 5V

CH<x>:Trig Type

Description
Sets the trigger type of the channel or queries the current setting.

Parameter
Rise | Fall | Both | High | Low | Off

Example (Visual Basic)

```
' Set CH1's trigger type to Fall.
ret = WeSetControl (hMo, "CH1:Trig Type", "Fall")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Type", value)
```

value → Fall

CH<x>:Trig Level

Description

Sets the trigger level of the channel or queries the current setting.

Parameter

–0.1 to 0.1 V for 100 mV range
–0.2 to 0.2 V for 200 mV range
–0.5 to 0.5 V for 500 mV range
–1 to 1 V for 1 V range
–2 to 2 V for 2 V range
–5 to 5 V for 5 V range
–10 to 10 V for 10 V range
–20 to 20 V for 20 V range
–50 to 50 V for 50 V range
–100 to 100 V for 100 V range
–200 to 200 V for 200 V range
–350 to 350 V for 350 V range

Example (Visual Basic)

```
' Set the CH1 trigger level to 1 V.
ret = WeSetControl (hMo, "CH1:Trig Level", "1")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Trig Level", value)
```

value → 1.0E+00

CH<x>:AAF

Description

Sets the type of input filter of the channel or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Set the CH1 input filter type to low pass.
ret = WeSetControl (hMo, "CH1:AAF", "Off")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:AAF", value)
```

value → Off

CH<x>:Filter

Description

Sets the frequency of the input filter of the channel or queries the current setting.

Parameter

During low-pass filter: OFF | 400 Hz | 4 kHz | 40 kHz | 100 kHz
During anti-aliasing filter: Off | 20 Hz | 40 Hz | 80 Hz | 200 Hz| 400 Hz | 800 Hz | 2 kHz | 4 kHz | 8 kHz | 20 kHz | 40 kHz

Example (Visual Basic)

```
' Set the CH1 input filter frequency to 400 Hz.
ret = WeSetControl (hMo, "CH1:Filter", "400Hz")
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Filter", value)
```

value → 400.0E+00

Trig:Source

Description

Sets the trigger source or queries the current setting

Parameter

Internal | BUSTRG

Example (Visual Basic)

```
' Set the trigger source to Internal.
ret = WeSetControl (hMo, "Trig:Source", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Source", value)
```

value → Internal

Trig:Combination

Description

Sets the trigger combination or queries the current setting.

Parameter

AND | OR

Example (Visual Basic)

```
' Set the trigger combination to AND.
ret = WeSetControl (hMo, "Trig:Combination", "AND")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Combination", value)
```

value → AND

Trig:Pretrigger

Description

Sets the amount of pre-trigger or queries the current setting.

Parameter

1 to the specified record length – 2

Example (Visual Basic)

```
' Set the pre-trigger amount to 100.
ret = WeSetControl (hMo, "Trig:Pretrigger", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Pretrigger", value)
```

value → 100

Trig:Hold Off

Description

Sets the trigger hold-off or queries the current setting.

Parameter

- When the acquisition mode is set to trigger mode and overlapped acquisition is disabled: Specified record length to 4194304
- When the acquisition mode is set to trigger mode and overlapped acquisition is enabled: 1 to 4194304
- When the acquisition mode is set to free run mode or gate mode (edge): 1 to 4194304

Example (Visual Basic)

```
' Set the trigger hold-off period to 100 samples.
ret = WeSetControl (hMo, "Trig:Hold Off", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Hold Off", value)
```

value → 100

Trig:Overlapped Acquisition

Description

Sets the overlap acquisition or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Allow overlap acquisition.
ret = WeSetControl (hMo, "Trig:Overlapped Acquisition", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Overlapped Acquisition", value)
```

value → On

Valid Acquisition Modes

Event	Valid?
Triggered	Yes
Free Run	Yes
Gate(Level)	Yes
Gate(Edge)	Yes

Acquisition Restrictions

Items	Possible Settings
Record length	4194304/N (where N is the number of measurement channels (1 or 2).
Data length per block	During trigger mode: 2 to the memory length During free run mode: 1 to the memory length
Number of block	During trigger/gate mode: 1 to 256 During free run mode: 1 to the memory length
Number of Acquisition	Data length per block > memory length/2: 1 Data length per block ≤ memory length/2: 1 to ∞ (specify 0 for ∞)

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	Yes	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is raw data in 16-bit signed integer format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is voltage in double-precision real format.
WeGetAcqDataEx	No	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeLoadPatternDataEx	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?
WWE_EV_MEASEND	Yes
WE_EV_BLOCKEND	Yes
WE_EV_MEASABORT	Yes
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

None

8.19 WE7281 4-CH, 100 kS/s D/A

ASCII Command	Description
Operation Mode	Sets the operation mode or queries the current setting.

DC:

ASCII Command	Description
DC:Output Mode	Sets the output mode in the DC mode or queries the current setting.
DC:CH<x>:On	Sets the output channel in the DC mode or queries the current setting.
DC:CH<x>:Range	Sets the output range in the DC mode or queries the current setting.
DC:CH<x>:Output	Sets the output value in the DC mode or queries the current setting.
DC:Output	Turns ON/OFF the output in the DC mode or queries the current setting.
DC:Trig:Trigger Out:Source	Sets the trigger output source channel for the DC mode or queries the current setting.
DC:Manual Triggger	Turns ON/OFF the manual trigger in the DC mode.

AG:

ASCII Command	Description
AG:Output Mode	Sets the output mode in the AG mode or queries the current setting.
AG:CH<x>:On	Sets the output channel in the AG mode or queries the current setting.
AG:CH<x>:Range	Sets the output range in the AG mode or queries the current setting.
AG:Sampling Interval	Sets the sampling interval in the AG mode or queries the current setting.
AG:Memory Partition	Sets the number of memory partitions in the AG mode or queries the current setting.
AG:Pattern Length	Sets the memory length in the AG mode or queries the current setting.
AG:Start Block No.	Sets the output start block in the AG mode or queries the current setting.
AG:End Block No.	Sets the output end block in the AG mode or queries the current setting.
AG:Trig:Trigger Out:Point	Sets the trigger signal output timing in the AG mode or queries the current setting.
AG:Misc:Load:Auto	Turns ON/OFF the auto load function in the AG mode or queries the current setting.
AG:Misc:Load:Manual	Sets the module/channel to be loaded in the AG mode or queries the current setting.
AG:Misc:Load:Block No.	Sets the block to be loaded in the AG mode or queries the current setting.
AG:Misc:CH Mode	Sets the number of channels used in the AG mode or queries the current setting.
AG:Misc:Time Base	Sets the time base in the AG mode or queries the current setting.
AG:Misc:Load Data	Loads the arbitrary waveform data in the AG mode.
AG:Manual Trigger	Turns ON/OFF the manual trigger signal in the AG mode.
AG:Output	Turns ON/OFF the output in the AG mode or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>. The number of channels per module is 4 for the WE7281. If two modules are linked, the command used to change the range of the AG function of channel 2 of the second module is "AG:CH6:Range."

FG:

ASCII Command	Description
FG:Output Mode	Sets the output mode in the FG mode or queries the current setting.
FG:CH<x>:On	Sets the output channel in the FG mode or queries the current setting.
FG:CH<x>:Sweep	Sets the sweep type in the FG mode or queries the current setting.
FG:CH<x>:Range	Sets the output range in the FG mode or queries the current setting.
FG:CH<x>:Func	Sets the output waveform in the FG mode or queries the current setting.
FG:CH<x>:Freq Start	Sets the output start frequency in the FG mode or queries the current setting.
FG:CH<x>:Freq End	Sets the output end frequency in the FG mode or queries the current setting.
FG:CH<x>:Ampl Start	Sets the output start amplitude in the FG mode or queries the current setting.
FG:CH<x>:Ampl End	Sets the output end amplitude in the FG mode or queries the current setting.
FG:CH<x>:Offset	Sets the offset voltage in the FG mode or queries the current setting.
FG:CH<x>:Phase	Sets the phase in the FG mode or queries the current setting.
FG:CH<x>:Invert	Sets the waveform inversion in the FG mode or queries the current setting.
FG:CH<x>:Duty Start	Sets the output waveform start duty ratio in the FG mode or queries the current setting.
FG:CH<x>:Duty End	Sets the output waveform end duty ratio in the FG mode or queries the current setting.
FG:CH<x>:Load ARB	Sets the arbitrary waveform data in the FG mode.
FG:Swp:Sweep	Sets the sweep mode in the FG mode or queries the current setting.
FG:Swp:Sweep Time	Sets the sweep time in the FG mode or queries the current setting.
FG:CH<x>:Sweep Pattern:Freq	Sets the frequency sweep pattern in the FG mode or queries the current setting.
FG:CH<x>:Sweep Pattern:Ampl	Sets the amplitude sweep pattern in the FG mode or queries the current setting.
FG:CH<x>:Sweep Pattern:Duty	Sets the duty ratio sweep pattern in the FG mode or queries the current setting.
FG:Swp:Arbitrary Sweep Pattern:Load Frequency	Sets the frequency sweep data in the FG mode or queries the current setting.
FG:Swp:Arbitrary Sweep Pattern:Load Amplitude	Sets the amplitude sweep data in the FG mode or queries the current setting.
FG:Swp:Arbitrary Sweep Pattern:Load Duty	Sets the duty ratio sweep data in the FG mode or queries the current setting.
FG:Swp:Arbitrary Sweep Pattern	Sets the transfer destination channel of the sweep data in the FG mode.
FG:Trig:Trigger Out:Source	Sets the trigger output source in the FG mode or queries the current setting.
FG:Trig:Trigger Out:Phase	Sets the trigger output phase in the FG mode or queries the current setting.
FG:Manual Trigger	Turns ON/OFF the manual trigger signal in the FG mode.
FG:Output	Turns ON/OFF the output in the FG mode or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>. The number of channels per module is 4 for the WE7281. If two modules are linked, the command used to change the range of the FG function of channel 2 of the second module is "FG:CH6:Range."

Operation Mode

Description
Sets the operation mode or queries the current setting.

Parameter
DC | AG | FG

Note:
The ASCII commands (DC:***, FG:***, and AG:***) for the various operation modes (DC, AG, and FG) are issued after setting the operation mode with this command.

Example (Visual Basic)
```
' Set the operation mode to AG.
ret = WeSetControl (hMo, "Operation Mode", "AG")
Dim value As Variant
ret = WeGetControl (hMo, "Operation Mode", value)
```
value → AG

DC:Output Mode

Description
Sets the output mode in the DC mode or queries the current setting.

Parameter
Cont | Trigger | Gate

Example (Visual Basic)
```
' Set the output mode in the DC mode to Trigger.
ret = WeSetControl (hMo, "DC:Output Mode", "Trigger")
Dim value As Variant
ret = WeGetControl (hMo, "DC:Output Mode", value)
```
value → DC

DC:CH<x>:On

Description
Sets the output channel in the DC mode or queries the current setting.

Parameter
On | Off

Example (Visual Basic)
```
' Turn ON the CH2 output in the DC mode.
ret = WeSetControl (hMo, "DC:CH2:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "DC:CH2:On", value)
```
value → On

DC:CH<x>:Range

Description
Sets the output range in the DC mode or queries the current setting.

Parameter

1V | 2V | 5V | 10V

Example (Visual Basic)

```
' Set the output range of CH1 in the DC mode to 5 V.
ret = WeSetControl (hMo, "DC:CH1:Range", "5V")
Dim value As Variant
ret = WeGetControl (hMo, "DC:CH1:Range", value)
```

value → 5V

DC:CH<x>:Output

Description

Sets the output value in the DC mode or queries the current setting.

Parameter

–1V to 1V for 1V range
–2V to 2V for 2V range
–5V to 5V for 5V range
–10V to 10V for 10V range

Example (Visual Basic)

```
' Set the output value of CH2 in the DC mode to 2.6 V.
ret = WeSetControl (hMo, "DC:CH2:Output", "2.6")
Dim value As Variant
ret = WeGetControl (hMo, "DC:CH2:Output", value)
```

value → 2.600E+00

DC:Output

Description

Turns ON/OFF the output in the DC mode or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Turns ON the output in the DC mode.
ret = WeSetControl (hMo, "DC:Output", "On")
Dim value As Variant
ret = WeGetControl (hMo, "DC:Output", value)
```

value → On

DC:Trig:Trigger Out:Source

Description

Sets the trigger output source channel for the DC mode or queries the current setting.

Parameter

Slot<y>-Ch<z> (<y> is the slot number, <z> is the channel number)

Example (Visual Basic)

```
' Set the trigger output source channel for the DC mode to CH1 of
' the module in Slot2.
ret = WeSetControl (hMo, "DC:Trig:Trigger Out:Source", "Slot2-Ch1")
Dim value As Variant
ret = WeGetControl (hMo, "DC:Trig:Trigger Out:Source", value)
```

value → Slot2-Ch1

DC:Manual Trigger

Description

Turns ON/OFF the manual trigger in the DC mode.

Parameter

On | Off

Example (Visual Basic)

```
' Turn ON the manual trigger in the DC mode.
ret = WeSetControl (hMo, "DC:Manual Trigger", "On")
```

AG:Output Mode

Description

Sets the output mode in the AG mode or queries the current setting.

Parameter

One Shot | Cont | Trigger | Repeat

Example (Visual Basic)

```
' Set the output mode in the AG mode to Trigger.
ret = WeSetControl (hMo, "AG:Output Mode", "Trigger")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Output Mode", value)
```

value → Trigger

AG:CH<x>:On

Description

Sets the output channel in the AG mode or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Turn ON the CH1 output in the AG mode.
ret = WeSetControl (hMo, "AG:CH1:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "AG:CH1:On", value)
```

value → On

AG:CH<x>:Range

Description
Sets the output range in the AG mode or queries the current setting.

Parameter
1V | 2V | 5V | 10V

Example (Visual Basic)
```
' Set the output range of CH1 in the AG mode to 2 V.
ret = WeSetControl (hMo, "AG:CH1:Range", "2V")
Dim value As Variant
ret = WeGetControl (hMo, "AG:CH1:Range", value)
```
value → 2V

AG:Sampling Interval

Description
Sets the sampling interval in the AG mode or queries the current setting.

Parameter
10 μs to 10 s

Example (Visual Basic)
```
' Set the sampling interval in the AG mode to 1 ms.
ret = WeSetControl (hMo, "AG:Sampling Interval", "1E-3")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Sampling Interval", value)
```
value → 1.00E-03

AG:Memory Partition

Description
Sets the number of memory partitions in the AG mode or queries the current setting.

Parameter
1 to 256

Example (Visual Basic)
```
' Set the number of memory partitions in the AG mode to 8.
ret = WeSetControl (hMo, "AG:Memory Partition", "8")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Memory Partition", value)
```
value → 8.0E+00

AG:Pattern Length

Description
Sets the memory length in the AG mode or queries the current setting.

Parameter
10 to the maximum allowed data length

Note:

As shown below, the maximum allowed data length varies depending on the number of channels used and the number of memory partitions.

Number of channels	Maximum allowed data length*
1	4 Mword
2	2 Mword
4	1 Mword

* When the Number of Memory Partitions is 1.
Data exceeding the maximum allowed data length are discarded when loaded.

Example (Visual Basic)

```
' Set the memory length in the AG mode to 1000.
ret = WeSetControl (hMo, "AG:Pattern Length", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Pattern Length", value)
```
value → 1000

AG:Start Block No.

Description

Sets the output start block in the AG mode or queries the current setting.

Parameter

1 to 256

Example (Visual Basic)

```
' Set the output start block in the AG mode to block 2.
ret = WeSetControl (hMo, "AG:Start Block No.", "2")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Start Block No.", value)
```
value → 2

AG:End Block No.

Description

Sets the output end block in the AG mode or queries the current setting.

Parameter

1 to 256

Example (Visual Basic)

```
' Set the output end block in the AG mode to block 2.
ret = WeSetControl (hMo, "AG:End Block No.", "2")
Dim value As Variant
ret = WeGetControl (hMo, "AG:End Block No.", value)
```
value → 2

AG:Trig:Trigger Out:Point

Description

Sets the trigger signal output timing in the AG mode or queries the current setting.

Parameter

1 to the data length

Example (Visual Basic)

```
' Set the trigger signal output timing in the AG mode to the 100th
' point.
ret = WeSetControl (hMo, "AG:Trigger Out:Point", "100")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Trigger Out:Point", value)
```

value → 100

AG:Misc:Load:Auto

Description

Turns ON/OFF the auto load function in the AG mode or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Turn ON the auto load function in the AG mode.
ret = WeSetControl (hMo, "AG:Misc:Load:Auto", "On")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Misc:Load:Auto", value)
```

value → On

AG:Misc:Load:Manual

Description

Sets the module/channel to be loaded in the AG mode or queries the current setting.

Parameter

Slot<y>-Ch<z> (<y> is the slot number, <z> is the channel number)

Example (Visual Basic)

```
' Set the waveform data from CH1 of the module in Slot2 to be
' loaded in the AG mode.
ret = WeSetControl (hMo, "AG:Misc:Load:Manual", "Slot2-Ch1")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Misc:Load:Manual", value)
```

value → Slot2-Ch1

AG:Misc:Load:Block No.

Description

Sets the block to be loaded in the AG mode or queries the current setting.

Parameter

Within the number of blocks

Example (Visual Basic)

```
' Set the waveform data block to be loaded in the AG mode to block 2.
ret = WeSetControl (hMo, "AG:Misc:Load:Block No.", "2")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Misc:Load:Block No.", value)
```

value → 2

AG:Misc:CH Mode

Description

Sets the number of channels used in the AG mode or queries the current setting.

Parameter

1CH | 2CH | 4CH

Example (Visual Basic)

```
' Set the number of channels used in the AG mode to 1.
ret = WeSetControl (hMo, "AG:Misc:CH Mode", "1CH")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Misc:CH Mode", value)
```

value → 1CH

AG:Misc:Time Base

Description

Sets the time base in the AG mode or queries the current setting.

Parameter

Internal | BUSCLK

Example (Visual Basic)

```
' Set the time base in the AG mode to internal clock.
ret = WeSetControl (hMo, "AG:Misc:Time Base", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Misc:Time Base", value)
```

value → Internal

AG:Misc:Load Data

Description

Loads the arbitrary waveform data in the AG mode.

Parameter

WE_UBYTE*n

Example (Visual Basic)

```
Dim size As Long
' Retrieve the data size after the conversion.
ret = WeWvf2W7281GetSize("c:\adc16384.wvf", 1, 0, size)
ReDim w7281buf(size) As Byte
' Acquire the waveform data and convert the data in the first block
' of CH1 to the arbitrary waveform data format.
ret = WeWvf2W7281("c:\adc16384.wvf", 1, 0, w7281buf(0), size)
' Transfer the converted data to the WE7281 waveform memory.
ret = WeSetControl(hMo, "AG:Misc:Load Data", w7281buf)
```

AG:Manual Trigger

Description

Turns ON/OFF the manual trigger signal in the AG mode.

Parameter

On | Off

Example (Visual Basic)

```
' Turn ON the manual trigger signal in the AG mode.
ret = WeSetControl (hMo, "AG:Manual Trigger", "On")
```

AG:Output

Description

Turns ON/OFF the output in the AG mode or queries the current setting.

Parameter

On | Off

Example (Visual Basic)

```
' Turn ON the output in the AG mode.
ret = WeSetControl (hMo, "AG:Output", "On")
Dim value As Variant
ret = WeGetControl (hMo, "AG:Output", value)
```

value → On

FG:Output Mode

Description

Turns ON/OFF the output in the AG mode or queries the current setting.

Parameter

Cont | Trigger | Gate

Example (Visual Basic)

```
' Set the output mode in the FG mode to Trigger.
ret = WeSetControl (hMo, "FG:Output Mode", "Trigger")
Dim value As Variant
ret = WeGetControl (hMo, "FG:Output Mode", value)
```

value → Trigger

FG:CH<x>:On

Description
Sets the output channel in the FG mode or queries the current setting.

Parameter
On | Off

Example (Visual Basic)
```
' Turn OFF the CH1 output in the FG mode.
ret = WeSetControl (hMo, "FG:CH1:On", "OFF")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:On", value)
```
value → OFF

FG:CH<x>:Sweep

Description
Sets the sweep type in the FG mode or queries the current setting.

Parameter
Off | Freq | Ampl | Freq&Ampl | Duty

Example (Visual Basic)
```
' Set the sweep type of CH1 in the FG mode to Freq.
ret = WeSetControl (hMo, "FG:CH1:Sweep", "Freq")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Sweep", value)
```
value → Freq

FG:CH<x>:Range

Description
Sets the output range in the FG mode or queries the current setting.

Parameter
1V | 2V | 5V | 10V

Example (Visual Basic)
```
' Set the output range of CH1 in the FG mode to 5 V.
ret = WeSetControl (hMo, "FG:CH1:Range", "5V")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Range", value)
```
value → 5.0E+00

FG:CH<x>:Func

Description
Sets the output waveform in the FG mode or queries the current setting.

Parameter
Sine | Pulse | Ramp | Triangle | Arbitrary | DC

Example (Visual Basic)

```
' Set the output waveform of CH1 in the FG mode to Pulse.
ret = WeSetControl (hMo, "FG:CH1:Func", "Pulse")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Func", value)
```

value → Pulse

FG:CH<x>:Freq Start

Description

Sets the output start frequency in the FG mode or queries the current setting.

Parameter

1 mHz to 20 kHz

Example (Visual Basic)

```
' Set the output start frequency of CH1 in the FG mode to 5 Hz.
ret = WeSetControl (hMo, "FG:CH1:Freq Start", "5.0")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Freq Start", value)
```

value → 5.000E+00

FG:CH<x>:Freq End

Description

Sets the output end frequency in the FG mode or queries the current setting.

Parameter

1 mHz to 20 kHz

Example (Visual Basic)

```
' Set the output end frequency of CH1 in the FG mode to 5 Hz.
ret = WeSetControl (hMo, "FG:CH1:Freq End", "5.0")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Freq End", value)
```

value → 5.000E+00

FG:CH<x>:Ampl Start

Description

Sets the output start amplitude in the FG mode or queries the current setting.

Parameter

0 to 2Vp-p for 1V range
0 to 4Vp-p for 2V range
0 to 10Vp-p for 5V range
0 to 20Vp-p for 10V range

Example (Visual Basic)

```
' Set the output start amplitude of CH3 in the FG mode to 2.5 V.
ret = WeSetControl (hMo, "FG:CH3:Ampl Start", "2.5")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH3:Ampl Start", value)
```

value → 2.500E+00

FG:CH<x>:Ampl End

Description

Sets the output end amplitude in the FG mode or queries the current setting.

Parameter

0 to 2Vp-p for 1V range
0 to 4Vp-p for 2V range
0 to 10Vp-p for 5V range
0 to 20Vp-p for 10V range

Example (Visual Basic)

```
' Set the output end amplitude of CH1 in the FG mode to 2.5 V.
ret = WeSetControl (hMo, "FG:CH1:Ampl End", "2.5")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Ampl End", value)
```

value → 2.500E+00

FG:CH<x>:Offset

Description

Sets the offset voltage in the FG mode or queries the current setting.

Parameter

–1V to 1V for 1V range
–2V to 2V for 2V range
–5V to 5V for 5V range
–10V to 10V for 10V range

Example (Visual Basic)

```
' Set the offset voltage of CH2 in the FG mode to 2.5 V.
ret = WeSetControl (hMo, "FG:CH2:Offset", "2.5")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH2:Offset", value)
```

value → 2.500E+00

FG:CH<x>:Phase

Description

Sets the phase of the output waveform in the FG mode or queries the current setting.

Parameter

–360 degree to 360 degree

Example (Visual Basic)

```
' Set the phase of the output waveform of CH1 in the FG mode to 180
' degrees.
ret = WeSetControl (hMo, "FG:CH1:Phase", "180")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Phase", value)
```

value → 180.0E+00

FG:CH<x>:Invert

Description

Sets the waveform inversion in the FG mode or queries the current setting.

Parameter

On | Off (On: invert, Off: do not invert)

Example (Visual Basic)

```
' Invert the output waveform of CH1 in the AG mode.
ret = WeSetControl (hMo, "FG:CH1:Invert", "On")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Invert", value)
```

value → On

FG:CH<x>:Duty Start

Description

Sets the output waveform start duty ratio in the FG mode or queries the current setting.

Parameter

0 to 100%

Example (Visual Basic)

```
' Set the output waveform start duty ratio of CH1 in the FG mode to
' 50%.
ret = WeSetControl (hMo, "FG:CH1:Duty Start", "50")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Duty Start", value)
```

value → 50.0E+00

FG:CH<x>:Duty End

Description

Sets the output waveform end duty ratio in the FG mode or queries the current setting.

Parameter

0 to 100%

Example (Visual Basic)

```
' Set the output waveform end duty ratio of CH1 in the FG mode to
' 50%.
ret = WeSetControl (hMo, "FG:CH1:Duty End", "50")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Duty End", value)
```

value → 50.0E+00

FG:CH<x>:Load ARB

Description

Sets the arbitrary waveform data in the FG mode.

Parameter

WE_SWORD*65536

Example (Visual Basic)

```
Dim s16buf(65536-1) As Integer
Dim size As Long
Dim ret As Integer
size = 65536*2
' Acquire the waveform data and convert the data in the first block
' of CH1 to the arbitrary waveform data format in FG format.
ret = WeWvf2S16("c:\adc65536.wvf", 1, 0, s16buf(0), size)
' Transfer the converted data to the WE7281 CH1 waveform memory.
ret = WeSetControl(hMo, "FG:CH1:Load ARB", s16buf)
```

FG:Swp:Sweep

Description

Sets the sweep mode in the FG mode or queries the current setting.

Parameter

Off | Repeat | Single | Single&&Hold

Example (Visual Basic)

```
' Set the sweep mode in the FG mode to Repeat.
ret = WeSetControl (hMo, "FG:Swp:Sweep", "Repeat")
Dim value As Variant
ret = WeGetControl (hMo, "FG:Swp:Sweep", value)
```

value → Repeat

FG:Swp:Sweep Time

Description

Sets the sweep time in the FG mode or queries the current setting.

Parameter

1 to 1000 s

Example (Visual Basic)

```
' Set the sweep time in the FG mode to 2 s.
ret = WeSetControl (hMo, "FG:Swp:Sweep Time", "2")
Dim value As Variant
ret = WeGetControl (hMo, "FG:Swp:Sweep Time", value)
```

value → 2.000E+00

FG:CH<x>:Sweep Pattern:Freq

Description

Sets the frequency sweep pattern in the FG mode or queries the current setting.

Parameter

Linear | Log | Arbitrary

Example (Visual Basic)

```
' Set the frequency sweep pattern of CH1 in the FG mode to Linear.
ret = WeSetControl (hMo, "FG:CH1:Sweep Pattern:Freq", "Linear")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Sweep Pattern:Freq", value)
```

value → Linear

FG:CH<x>:Sweep Pattern:Ampl

Description

Sets the amplitude sweep pattern in the FG mode or queries the current setting.

Parameter

Linear | Log | Arbitrary

Example (Visual Basic)

```
' Set the amplitude sweep pattern of CH1 in the FG mode to Linear.
ret = WeSetControl (hMo, "FG:CH1:Sweep Pattern:Ampl", "Linear")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Sweep Pattern:Ampl", value)
```

value → Linear

FG:CH<x>:Sweep Pattern:Duty

Description

Sets the amplitude sweep pattern in the FG mode or queries the current setting.

Parameter

Linear | Log | Arbitrary

Example (Visual Basic)

```
' Set the amplitude sweep pattern of CH1 in the FG mode to Linear.
ret = WeSetControl (hMo, "FG:CH1:Sweep Pattern:Duty", "Linear")
Dim value As Variant
ret = WeGetControl (hMo, "FG:CH1:Sweep Pattern:Duty", value)
```

value → Linear

FG:Swp:Arbitrary Sweep Pattern

Description

Sets the transfer destination channel of the sweep pattern in the FG mode.

Parameter

Slot<y>-Ch<z> (<y> is the slot number, <z> is the channel number)

Example (Visual Basic)

```
' Set the transfer destination of the sweep pattern in the FG mode
' to CH1 of the module in Slot 1.
ret = WeSetControl (hMo, "FG:Swp:Arbitrary Sweep Pattern", "Slot1-
Ch1")
Dim value As Variant
ret = WeGetControl (hMo, "FG:Swp:Arbitrary Sweep Pattern", value)
```

value → Slot1-Ch1

FG:Swp:Arbitrary Sweep Pattern:Load Frequency

Description

Sets the frequency sweep data in the FG mode or queries the current setting.

Parameter

WE_SLONG*65536

Example (Visual Basic)

```
Dim w32buf(65536-1) As Long
Dim size As Long
size = 65536*4
' Acquire the waveform data and convert the data in the first block
' of CH1 to the arbitrary waveform sweep data in FG format.
ret = WeWvf2W32("c:\adc65536.wvf", 1, 0, w32buf(0), size)
' Transfer the converted data to the WE7281 CH1.
ret = WeSetControl (hMo, "FG:Swp:Arbitrary Sweep Pattern", "Slot1-
Ch1")
ret = WeSetControl(hMo, "FG:Swp:Arbitrary Sweep Pattern:Load
Frequency", w32buf)
```

FG:Swp:Arbitrary Sweep Pattern:Load Amplitude

Description

Sets the amplitude sweep data in the FG mode or queries the current setting.

Parameter

WE_SLONG*65536

Example (Visual Basic)

```
Dim w32buf(65536-1) As Long
Dim size As Long
size = 65536*4
' Acquire the waveform data and convert the data in the first block
' of CH1 to the arbitrary waveform sweep data in FG format.
ret = WeWvf2W32("c:\adc65536.wvf", 1, 0, w32buf(0), size)
' Transfer the converted data to the WE7281 CH1.
ret = WeSetControl (hMo, "FG:Swp:Arbitrary Sweep Pattern", "Slot1-
Ch1")
ret = WeSetControl(hMo, " FG:Swp:Arbitrary Sweep Pattern:Load
Amplitude", w32buf)
```

FG:Swp:Arbitrary Sweep Pattern:Load Duty

Description

Sets the duty ratio sweep data in the FG mode or queries the current setting.

Parameter

WE_SLONG*65536

Example (Visual Basic)

```
Dim w32buf(65536-1) As Long
Dim size As Long
size = 65536*4
' Acquire the waveform data and convert the data in the first block
' of CH1 to the arbitrary waveform sweep data in FG format.
ret = WeWvf2W32("c:\adc65536.wvf", 1, 0, w32buf(0), size)
' Transfer the converted data to the WE7281 CH1.
ret = WeSetControl (hMo, "FG:Swp:Arbitrary Sweep Pattern", "Slot1-
Ch1")
ret = WeSetControl(hMo, "FG:Swp:Arbitrary Sweep Pattern:Load Duty",
w32buf)
```

FG:Trig:Trigger Out:Source

Description

Sets the trigger output source in the FG mode or queries the current setting.

Parameter

Slot<y>-Ch<z> (<y> is the slot number, <z> is the channel number)

Example (Visual Basic)

```
' Set the trigger output source in the FG mode to CH1 of the
' module in Slot 1.
ret = WeSetControl (hMo, "FG:Trig:Trigger Out:Source", "Slot1-Ch1")
Dim value As Variant
ret = WeGetControl (hMo, "FG:Trig:Trigger Out:Source", value)
```

value → Slot1-Ch1

FG:Trig:Trigger Out:Phase

Description
Sets the trigger output source in the FG mode or queries the current setting.

Parameter
0 to 360 degree

Example (Visual Basic)
```
' Set the trigger output phase in the FG mode to 180 degrees.
ret = WeSetControl (hMo, "FG:Trig:Trigger Out:Phase", "180")
Dim value As Variant
ret = WeGetControl (hMo, "FG:Trig:Trigger Out:Phase", value)
```
value → 180.0E+00

FG:Manual Trigger

Description
Turns ON/OFF the manual trigger signal in the FG mode.

Parameter
On | Off

Example (Visual Basic)
```
' Turn ON the manual trigger signal in the FG mode.
ret = WeSetControl (hMo, "FG:Manual Trigger", "On")
Dim value As Variant
ret = WeGetControl (hMo, "FG:Manual Trigger", value)
```
value → On

FG:Output

Description
Turns ON/OFF the output in the FG mode or queries the current setting.

Parameter
On | Off

Example (Visual Basic)
```
' Turn ON the output in the FG mode.
ret = WeSetControl (hMo, "FG:Output", "On")
Dim value As Variant
ret = WeGetControl (hMo, "FG:Output", value)
```
value → On

Valid Acquisition Modes

This module is not a data acquisition module.

Acquisition Restrictions

This module is not a data acquisition module.

Valid Common Measurement Control API

API	Valid?
WeStart	No
WeStop	No
WeStartSingle	No
WeStartWithEvent	No
WeIsRun	No
WeLatchData	No
WeGetAcqDataInfo	No
WeGetAcqData	No
WeGetScaleData	No
WeGetMeasureParam	No
WeSaveAcqData	No
WeSaveScaleData	No
WeSaveAsciiData	No
WeGetCurrentData	No
WeGetAcqDataEx	No
WeGetAcqDataSize	No
WeSaveAcqHeader	No
WeSavePatternData	No
WeLoadPatternData	No
WeLoadPatternDataEx	Yes
WeStartEx	No
WeStopEx	No

Valid Common Events

Event	Event Description
WWE_EV_MEASEND	No
WE_EV_BLOCKEND	No
WE_EV_MEASABORT	No
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

None

8.20 WE7311 1 GS/s Digital Oscilloscope

ASCII Commands

Acquisition controls

ASCII Command	Description	OSC	DGTZR
Trigger Mode	Sets the trigger mode or queries the current setting.	Yes	Yes
Sampling Interval	Sets the sampling interval or queries the current setting.	No	Yes
Time/div	Sets the Time/div setting or queries the current setting.	Yes	No
Acquisition Counter	Sets the number of times to execute the acquisition or queries the current setting.	Yes	Yes

OSC: Can be used when the operation mode is set to oscilloscope, DGTZR: Can be used when the operation mode is set to digitizer.

Trace controls (CH<x>:)

ASCII Command	Description	OSC	DGTZR
CH<x>:Range	Sets the range or queries the current setting.	No	Yes
CH<x>:Offset	Sets offset voltage or queries the current setting.	Yes	Yes
CH<x>:Coupling	Sets the input coupling or queries the current setting.	Yes	Yes
CH<x>:Probe	Sets probe attenuation setting or queries the current setting.	Yes	Yes
CH<x>:V/div	Sets the V/div setting or queries the current setting.	Yes	No

OSC: Can be used when the operation mode is set to oscilloscope, DGTZR: Can be used when the operation mode is set to digitizer.

Note

Enter the channel number in the <x> of CH<x>. There is one channel on each WE7311 module. If two modules are linked and you wish to change the range of the second module, specify "CH2:Range" in the command line.

Trigger controls

ASCII Command	Description	OSC	DGTZR
Trig:Source	Sets the trigger source or queries the current setting.	Yes	Yes
Trig:Coupling	Sets the trigger coupling or queries the current setting.	Yes	Yes
Trig:Slope	Sets the trigger slope or queries the current setting.	Yes	Yes
Trig:Level	Sets the trigger level or queries the current setting.	Yes	Yes
Trig:Pretrigger	Sets the amount of pretrigger or queries the current setting.	No	Yes
Trig:Delay	Sets the amount of delay or queries the current setting.	No	Yes
Trig:Position	Sets the trigger position or queries the current setting.	Yes	No
Trig:DelayTime	Sets the delay time or queries the current setting.	Yes	No

OSC: Can be used when the operation mode is set to oscilloscope, DGTZR: Can be used when the operation mode is set to digitizer.

Clock controls

ASCII Command	Description	OSC	DGTZR
CLK:Sampling CLK Source	Sets the sampling clock source or queries the current setting.	Yes	Yes
CLK:Reference CLK Source	Sets reference clock or queries the current setting.	Yes	Yes
CLK:External Threshold Level	Sets the threshold level of the external clock or queries the current setting.	Yes	Yes

OSC: Can be used when the operation mode is set to oscilloscope, DGTZR: Can be used when the operation mode is set to digitizer.

Memory controls

ASCII Command	Description	OSC	DGTZR
Mem:Memory Partition	Sets the number of memory partitions or queries the current setting.	Yes	Yes
Mem:Record Length	Sets the record length or queries the current setting.	Yes	Yes
Mem:No. of Acquisitions	Sets the number of acquisitions or queries the current setting.	Yes	Yes

OSC: Can be used when the operation mode is set to oscilloscope, DGTZR: Can be used when the operation mode is set to digitizer.

Other controls

ASCII Command	Description	OSC	DGTZR
Misc:Auto Setup:Exec	Executes auto setup.	Yes	Yes
Misc:Calibration:Auto Cal	Sets the automatic calibration setting or queries the current setting.	Yes	Yes
Misc:Calibration:Cal Exec	Executes calibration.	Yes	Yes
Misc:Calibration:Cal Query	Queries the calibration condition.	Yes	Yes
Misc:Operation Mode	Sets the operation mode or queries the current setting.	Yes	Yes

OSC: Can be used when the operation mode is set to oscilloscope, DGTZR: Can be used when the operation mode is set to digitizer.

Trigger Mode

Description

Sets the trigger mode or queries the current setting.

Parameter

Auto | Normal

Example (Visual Basic)

```
' Set the trigger mode to Auto.
ret = WeSetControl (hMo, "Trigger Mode", "Auto")
Dim value As Variant
ret = WeGetControl (hMo, "Trigger Mode", value)
```

value → Auto

Sampling Interval

Description

Sets the sampling interval or queries the current setting.

Parameter

1 ns to 10 ms (1-2-2.5-4-5 steps)

Example (Visual Basic)

```
' Set the sampling interval to 1 ns.
ret = WeSetControl (hMo, "Sampling Interval", "1E-9")
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```

value → 1.0E-09

Time/div

Description
Sets the Time/div setting or queries the current setting.

Parameter
10 ns to 50 s

Example (Visual Basic)
```
' Set the Time/Div to 10 s.
ret = WeSetControl (hMo, "Time/div", "10")
Dim value As Variant
ret = WeGetControl (hMo, "Time/div", value)
```
value → 10.0E+00

Acquisition Counter

Description
Sets the number of times to execute the acquisition or queries the current setting.

Parameter
0 to 2047

Example (Visual Basic)
```
Dim value As Variant
ret = WeGetControl (hMo, "Acquisition Counter", value)
```
value → 100

CH<x>:Range

Description
Sets the range or queries the current setting.

Parameter
25 mV | 50 mV | 100 mV | 250 mV | 500 mV | 1 V | 2.5 V

Example (Visual Basic)
```
' Set the range of CH2 to 1 V.
ret = WeSetControl (hMo, "CH2:Range", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CH2:Range", value)
```
value → 1.0E+00

CH<x>:Offset

Description
Sets offset voltage or queries the current setting.

Parameter

When probe attenuation is set to 1:1
Oscilloscope mode:
When V/div = 5 mV, 10 mV, 20 mV, 50 mV: offset = ±2.0000 V
When V/div = 100 mV, 200 mV, 500 mV: offset = ±20.000 V
Digitizer mode:
When range = 25 mV, 50 mV, 100 mV, 250 mV: offset = ±2.0000 V
When range = 500 mV, 1V, 2.5V: offset = ±20.000 V

Example (Visual Basic)

```
' Set the offset voltage of CH3 to 1.25 V.
ret = WeSetControl (hMo, "CH3:Offset", "1.25")
Dim value As Variant
ret = WeGetControl (hMo, "CH3:Offset", value)
```

value → 1.25E+00

CH<x>:Coupling

Description

Sets input coupling or queries the current setting.

Parameter

AC | DC | GND | AC 50 | DC 50

Example (Visual Basic)

```
' Set the input coupling of CH4 to AC 50.
ret = WeSetControl (hMo, "CH4:Coupling", "AC 50")
Dim value As Variant
ret = WeGetControl (hMo, "CH4:Coupling", value)
```

value → AC 50

CH<x>:Probe

Description

Sets probe attenuation setting or queries the current setting.

Parameter

1:1 | 10:1 | 100:1 | 1000:1

Example (Visual Basic)

```
' Set the probe attenuation of CH5 to 100:1.
ret = WeSetControl (hMo, "CH5:Probe", "100:1")
Dim value As Variant
ret = WeGetControl (hMo, "CH5:Probe", value)
```

value → 100:1

CH<x>:V/div

Description

Sets the V/div setting or queries the current setting.

Parameter

5 mV | 10 mV | 20 mV | 50 mV | 100 mV | 200 mV | 500 mV

Example (Visual Basic)

```
' Set the Volt/div setting of CH6 to 50 mV.
ret = WeSetControl (hMo, "CH6:V/div", "50E-3")
Dim value As Variant
ret = WeGetControl (hMo, "CH6:V/div", value)
```

value → 50.0E-3

Trig:Source

Description

Sets the trigger source or queries the current setting.

Parameter

BUSTRG | External-1M | External-50 | CH1 | CH2 | CH3 | CH4 | CH5 | CH6 | CH7 | CH8

Example (Visual Basic)

```
' Set the trigger source to External 1M.
ret = WeSetControl (hMo, "Trig:Source", "External-1M")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Source", value)
```

value → External-1M

Trig:Coupling

Description

Sets the trigger coupling or queries the current setting.

Parameter

DC | LF Reject

Example (Visual Basic)

```
' Set the trigger coupling to DC.
ret = WeSetControl (hMo, "Trig:Coupling", "DC")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Coupling", value)
```

value → DC

Trig:Slope

Description

Sets the trigger slope or queries the current setting.

Parameter

Rise | Fall

Example (Visual Basic)

```
' Set the trigger slope to Rise.
ret = WeSetControl (hMo, "Trig:Slope", "Rise")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Slope", value)
```

value → Rise

Trig:Level

Description

Sets the trigger level or queries the current setting.

Parameter

During the oscilloscope mode: Value equivalent to 5.0 times the V/div setting.
During the digitizer mode: Value corresponding to the range.

Example (Visual Basic)

```
' Set the trigger level to 1.0 V.
ret = WeSetControl (hMo, "Trig:Level", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Level", value)
```

value → 1.00000E+00

Trig:Pretrigger

Description

Sets the amount of pretrigger (number of sampling points) or queries the current setting.

Parameter

0 to specified record length

Example (Visual Basic)

```
' Set the amount of pretrigger to 100.
ret = WeSetControl (hMo, "Trig:Pretrigger", "100")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Pretrigger", value)
```

value → 100

Trig:Delay

Description

Sets the amount of delay (number of sampling points) or queries the current setting.

Parameter

0 to 200000000 (However, the maximum value is equal to the value corresponding to 300 s.)

Example (Visual Basic)

```
' Set the amount of delay to 1000.
ret = WeSetControl (hMo, "Trig:Delay", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Delay", value)
```

value → 1000

Trig:Position

Description

Sets the trigger position or queries the current setting.

Parameter

±5 div

Example (Visual Basic)

```
' Set the trigger position to +2.5 div.
ret = WeSetControl (hMo, "Trig:Position", "2.5")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Position", value)
```
value → 2.5E+00

Trig:DelayTime

Description

Sets the delay time or queries the current setting.

Parameter

1ns to 300s

Example (Visual Basic)

```
' Set the delay time to 100 ms.
ret = WeSetControl (hMo, "Trig:DelayTime", "100E-3")
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Delay Time", value)
```
value → 100.00000E-03

CLK:Sampling Source

Description

Sets the sampling source or queries the current setting.

Parameter

Internal | External-1M | External-50

Example (Visual Basic)

```
' Set the sampling source to Internal.
ret = WeSetControl (hMo, "CLK:Sampling Source", "Internal")
Dim value As Variant
ret = WeGetControl (hMo, "CLK:Sampling Source", value)
```
value → Internal

CLK:Reference Source

Description

Sets the reference source or queries the current setting.

Parameter

Internal | External-1M | External-50 | BUSCLK

Example (Visual Basic)

```
' Set the reference source to BUSCLK.
ret = WeSetControl (hMo, "CLK:Reference Source", "BUSCLK")
Dim value As Variant
ret = WeGetControl (hMo, "CLK: Reference Source", value)
```
value → BUSCLK

CLK:External Threshold Level

Description

Sets the threshold level of the external clock or queries the current setting.

Parameter

±2V

Example (Visual Basic)

```
' Set the threshold level of the external clock to 1.0 V.
ret = WeSetControl (hMo, "CLK:External Threshold Level", "1.0")
Dim value As Variant
ret = WeGetControl (hMo, "CLK: External Threshold Level", value)
```

value → 1.0E+00

Mem:Memory Partition

Description

Sets the number of memory partitions or queries the current setting.

Parameter

1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096

Example (Visual Basic)

```
' Set the number of memory partitions to 128.
ret = WeSetControl (hMo, "Mem:Memory Partition", "128")
Dim value As Variant
ret = WeGetControl (hMo, "Mem:Memory Partition", value)
```

value → 128.0E+00

Mem:Record Length

Description

Sets the record length or queries the current setting.

Parameter

When the number of memory partitions = 1:	100 to 2000000
When the number of memory partitions = 2:	100 to 1000000
When the number of memory partitions = 4:	100 to 520000
When the number of memory partitions = 8:	100 to 240000
When the number of memory partitions = 16:	100 to 120000
When the number of memory partitions = 32:	100 to 64000
When the number of memory partitions = 64:	100 to 32000
When the number of memory partitions = 128:	100 to 16000
When the number of memory partitions = 256:	100 to 7200
When the number of memory partitions = 512:	100 to 3600
When the number of memory partitions = 1024:	100 to 1600
When the number of memory partitions = 2048:	100 to 600
When the number of memory partitions = 4096:	100

Example (Visual Basic)

```
' Set the record length to 2000.
ret = WeSetControl (hMo, "Mem:Record Length", "2000")
Dim value As Variant
ret = WeGetControl (hMo, "Mem:Record Length", value)
value → 2000
```

Mem:No. of Acquisitions

Description

Sets the number of acquisitions or queries the current setting.

Parameter

1 to the number of memory partitions

Example (Visual Basic)

```
' Set the number of acquisitions to 5.
ret = WeSetControl (hMo, "Mem:No. Of Acquisition", "5")
Dim value As Variant
ret = WeGetControl (hMo, "Mem:No. Of Acquisition", value)
value → 5
```

Misc:Auto Setup:Exec

Description

Executes auto setup.

Parameter

None

Example (Visual Basic)

```
' The Variant parameter is not initialized, because there are no
parameters.
Dim value As Variant
' Execute auto setup.
ret = WeSetControl (hMo, "Misc:Auto Setup:Exec", value)
```

Misc:Calibration:Auto Cal

Description

Sets the automatic calibration setting or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable automatic calibration.
ret = WeSetControl (hMo, "Misc:Calibration:Auto Cal", "On")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Calibration:Auto Cal", value)
value → On
```

Misc:Calibration:Cal Exec

Description
Executes calibration.

Parameter
None

Example (Visual Basic)
```
' The Variant parameter is not initialized, because there are no
parameters.
Dim value As Variant
' Execute calibration.
ret = WeSetControl (hMo, "Misc:Calibration:Cal Exec", value)
```

Misc:Calibration:Cal Query

Description
Queries the calibration condition.

Parameter
0 (completed) | 1 (in progress)

Note:

Other commands cannot be executed while the calibration is in progress. Execute them after the calibration has completed.

Example (Visual Basic)
```
Dim value As Variant
' Queries the calibration condition.
ret = WeGetControl (hMo, "Misc:Calibration:Cal Query", value)
```
value → 1

Misc:Operation Mode

Description
Sets the operation mode or queries the current setting.

Parameter
Oscilloscope | Digitizer

Example (Visual Basic)
```
' Set the operation mode to oscilloscope mode.
ret = WeSetControl (hMo, "Misc:Operation Mode", "Oscilloscope")
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Operation Mode", value)
```
value → Oscilloscope

Valid Acquisition Modes

Event	Valid?	Note
Triggered		Yes
Free Run		No
Gate(Level)		No
Gate(Edge)		No

Acquisition Restrictions

Items	
Record length	Up to 2000000
Data length per block	100 to the memory length
Number of blocks	1 to 4096
Number of acquisitions	1

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	
WeStartWithEvent	Yes	
WeIsRun	Yes	
WeLatchData	No	
WeGetAcqDataInfo	Yes	
WeGetAcqData	Yes	Data is raw data (A/D data) in 8-bit signed integer format.
WeGetScaleData	Yes	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	No	
WeGetAcqDataEx	Yes	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?	Event Description
WE_EV_MEASEND	Yes	
WE_EV_BLOCKEND	No	No events occur at the end of individual block measurements. It is issued at the same time as WE_EV_MEASEND when measurements of all blocks have been completed.
WE_EV_MEASABORT	Yes	
WE_EV_TRIG_HIGH	No	
WE_EV_TRIG_LOW	No	
WE_EV_TRIG_PULSE	No	
WE_EV_ALARM	No	
WE_EV_CLOSE_MODULE_GUI	Yes	

Module-specific Events

Event	Event Description
0x01000000	[INPUT/EXT IN] input signal of the module at CH1 overloaded.
0x02000000	[INPUT/EXT IN] input signal of the module at CH2 overloaded.
0x04000000	[INPUT/EXT IN] input signal of the module at CH3 overloaded.
0x08000000	[INPUT/EXT IN] input signal of the module at CH4 overloaded.
0x10000000	[INPUT/EXT IN] input signal of the module at CH5 overloaded.
0x20000000	[INPUT/EXT IN] input signal of the module at CH6 overloaded.
0x40000000	[INPUT/EXT IN] input signal of the module at CH7 overloaded.
0x80000000	[INPUT/EXT IN] input signal of the module at CH8 overloaded.

Module-specific Error Codes

None

8.21 WE7521 4-CH Timing Measurement

ASCII Commands

• For Counter Mode

ASCII Command	Description
Operation Mode	Sets the operation mode or queries the current setting.
Acquisition Mode	Sets the acquisition mode or queries the current setting.
Sampling Interval	Sets the sampling interval or queries the current setting.
Record Length	Sets the record length or queries the current setting.
Memory Partition	Sets the memory partitions or queries the current setting.
No. of Acquisitions	Sets the number of acquisitions or queries the current setting.
Counter Reset:Type	Sets the counter reset type or queries the current setting.
Counter Reset:Reset All	Executes the counter reset on all channels.

IN<x>:

ASCII Command	Description
IN<x>:Coupling	Sets the input coupling or queries the current setting.
IN<x>:Level	Sets the input threshold level or queries the current setting.
IN<x>:Filter	Sets the input filter or queries the current setting.
IN<x>:Hys	Sets the hysteresis width or queries the current setting.

CH<x>:

ASCII Command	Description
CH<x>:Function	Sets the measurement function or queries the current setting.
CH<x>:Source-A	Sets the input to be measured/slope or queries the current setting.
CH<x>:Source-B	Sets the input to be measured/slope or queries the current setting.
CH<x>:Limit	Turns ON/OFF the cycle stop determination function of the channel or queries the current setting.
CH<x>:Reset	Executes the counter reset on the channel.

Trig:

ASCII Command	Description
Trig:Source	Sets the trigger source or queries the current setting.
Trig:Input Source	Sets the trigger source target or queries the current setting.
Trig:Slope	Sets the trigger slope or queries the current setting.
Trig:Measure Source	Sets the trigger source target or queries the current setting.
Trig:Type	Sets the trigger type or queries the current setting.
Trig:Condition	Sets the trigger condition or queries the current setting.
Trig:Threshold	Sets the threshold level or queries the current setting.
Trig:PreTrigger	Sets the pretrigger or queries the current setting.
Trig:HoldOff	Sets the trigger hold off or queries the current setting.

Misc:

ASCII Command	Description
Misc:TimeBase:Source	Sets the time base or queries the current setting.
Misc:TimeBase:Input Source	Sets the time base input source or queries the current setting.
Misc:TimeBase:Slope	Sets the clock detection slope or queries the current setting.
Misc:DataHold	Sets the data hold or queries the current setting.
Misc:HysType	Sets the hysteresis direction or queries the current setting.
Misc:Limit Of Period	Sets the timeout time of the cycle stop determination or queries the current setting.

• **For Time Stamp Mode**

ASCII Command	Description
Operation Mode	Sets the operation mode or queries the current setting.
HysType	Sets the hysteresis direction or queries the current setting.

IN<x>:

ASCII Command	Description
IN<x>:On	Turns On/Off the measurement or queries the current setting.
IN<x>:Coupling	Sets the input coupling or queries the current setting.
IN<x>:Slope	Sets the slope or queries the current setting.
IN<x>:Level	Sets the input threshold level or queries the current setting.
IN<x>:Filter	Sets the input filter or queries the current setting.
IN<x>:Hys	Sets the hysteresis width or queries the current setting.

Note

Enter the channel number in the <x> of CH<x>. If the modules are linked specify a serial number from the parent module.
There are four inputs per module for the WE7521 module.
If two modules are linked and you wish to change the range of channel 2 of the second module, specify "CH6:Range" in the command line.

Operation Mode (common to counter and time stamp modes)

Description

Sets the operation mode or queries the current setting.

Parameter

Counter | Time Stamp

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Operation Mode", "Counter")
' Set the operation mode to Counter.
Dim value As Variant
ret = WeGetControl (hMo, "Operation Mode", value)
```

value → Counter

Acquisition Mode (for counter mode only)

Description

Sets the acquisition mode or queries the current setting.

Parameter

Triggered | Free Run | Gate(Level) | Gate(Edge)

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Acquisition Mode", "Free Run")
' Set the acquisition mode to Free Run.
Dim value As Variant
ret = WeGetControl (hMo, "Acquisition Mode", value)
```

value → Free Run

Sampling Interval (for counter mode only)

Description
Sets the sampling interval or queries the current setting.

Parameter
0.000002 to 10.0

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Sampling Interval", "10E-3")
' Set the sampling interval to 10 ms.
Dim value As Variant
ret = WeGetControl (hMo, "Sampling Interval", value)
```
value → 10.000E-03

Record Length (for counter mode only)

Description
Sets the record length or queries the current setting.

Parameter
During trigger mode: 2 to 1048576/the number of memory partitions
During gate mode: 2 to 1048576
During free run mode: 1 to 1048576
However, the record length range changes so that the measurement period does not fall below 5 ms.
Example: When the sampling interval is 10 ms
500 to 1048576 regardless of the acquisition mode.

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Record Length", "1000")
' Set the record length to 1000.
Dim value As Variant
ret = WeGetControl (hMo, "Record Length", value)
```
value → 1000

Memory Partition (for counter mode only)

Description
Sets the number of memory partitions or queries the current setting.

Parameter
1 to 256 (2n step values)

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Memory Partition", "2")
' Set the number of memory partitions to 2.
Dim value As Variant
ret = WeGetControl (hMo, "Memory Partition", value)
```
value → 2.0E+00

No. of Acquisitions (for counter mode only)

Description
Sets the number of acquisitions or queries the current setting.

Parameter
0 to 65535

Example (Visual Basic)
```
ret = WeSetControl (hMo, "No. of Acquisitions", "100")
' Set the number of acquisitions to 100.
Dim value As Variant
ret = WeGetControl (hMo, "No. of Acquisitions", value)
```
value → 100

Counter Reset:Type (for counter mode only)

Description
Sets the counter reset type or queries the current setting.

Parameters
Auto | Manual

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Counter Reset:Type", "Manual")
' Set the counter reset type to Manual.
Dim value As Variant
ret = WeGetControl (hMo, "Counter Reset:Type", value)
```
value → Manual

Counter Reset:Reset All (for counter mode only)

Description
Executes counter reset. Counter reset is available on all channels whose measurement function is set to Totalize/UpDown1/ UpDown2/UpDown4.

Parameters
None

Example (Visual Basic)
```
Dim value As Variant
' Execute counter reset.
ret = WeGetControl (hMo, "Counter Reset:Reset All", value)
```

IN<x>:On (for time stamp mode only)

Description
Turns On/Off the measurement or queries the current setting.

Parameter
Off | On

Example (Visual Basic)

```
ret = WeSetControl (hMo, "IN1:On", "On")
' Make measurement of IN1.
Dim value As Variant
ret = WeGetControl (hMo, "IN1:On", value)
```

value → On

IN<x>:Coupling (common to counter and time stamp modes)

Description

Sets input coupling or queries the current setting.

Parameter

AC | DC

Example (Visual Basic)

```
ret = WeSetControl (hMo, "IN1:Coupling", "DC")
' Set the input coupling of IN1 to DC.
Dim value As Variant
ret = WeGetControl (hMo, "IN1:Coupling", value)
```

value → DC

IN<x>:Slope (for time stamp mode only)

Description

Sets the slope or queries the current setting.

Parameter

Rise | Fall | Both

Example (Visual Basic)

```
ret = WeSetControl (hMo, "IN1:Slope", "Rise")
' Set the signal detection slope of IN1 to Rise.
Dim value As Variant
ret = WeGetControl (hMo, "IN1:Slope", value)
```

value → Rise

IN<x>:Level (common to counter and time stamp modes)

Description

Sets the input threshold level or queries the current setting.

Parameter

–20.0 to 20.0 (0.1 V steps)

Example (Visual Basic)

```
ret = WeSetControl (hMo, "IN1:Level", "5.0")
' Set the input threshold level of IN1 to 5.0 V.
Dim value As Variant
ret = WeGetControl (hMo, "IN1:Level", value)
```

value → 5.0E+00

IN<x>:Filter (common to counter and time stamp modes)

Description
Sets the input filter or queries the current setting.

Parameter
Off | 1kHz | 10kHz | 100kHz

Example (Visual Basic)
```
ret = WeSetControl (hMo, "IN1:Filter", "1kHz")
' Set the input filter of IN1 to 1 kHz.
Dim value As Variant
ret = WeGetControl (hMo, "IN1:Filter", value)
```
value → 1.0000E+03

IN<x>:Hys (common to counter and time stamp modes)

Description
Sets the hysteresis width or queries the current setting.

Parameter
Off (NORMAL) | On (WIDE)

Example (Visual Basic)
```
ret = WeSetControl (hMo, "IN1:Hys", "On")
' Set the hysteresis width of IN1 to On (WIDE).
Dim value As Variant
ret = WeGetControl (hMo, "IN1:Hys", value)
```
value → On

CH<x>:Function (for counter mode only)

Description
Sets the measurement function or queries the current setting.

Parameter
Off | Period | TI | Totalize | Totalize(Gate) | UpDown1 | UpDown2 | UpDown4 | Ratio x1 | Ratio x16 | Ratio x128 | Ratio x1024

Example (Visual Basic)
```
ret = WeSetControl (hMo, "CH1:Function", "Period")
' Set the measurement function of CH1 to Period.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Function", value)
```
value → Period

CH<x>:Source-A (for counter mode only)

Description
Set the combination of the input to be measured (IN) and the slope of the input signal or queries the current setting.

Parameter

- When the measurement function is Period/TI/Totalize/Totalize(Gate)/Ratio x1/Ratio x16/Ratio x128/Ratio x1024
  IN1_Rise | IN1_Fall | IN2_Rise | IN2_Fall | IN3_Rise | IN3_Fall | IN4_Rise | IN4_Fall
- When the measurement function is UpDown1/UpDown2/UpDown4
  IN1-IN2 | IN3- IN4

Example (Visual Basic)

```
ret = WeSetControl (hMo, "CH1:Source-A", "IN1_Rise")
' Set the input to be measured/slope of CH1 to IN1_Rise.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Source-A", value)
```

value → IN1_Rise

CH<x>:Source-B (for counter mode only)

Description

Set the combination of the input to be measured (IN) and the slope of the input signal or queries the current setting.

Parameter

- When the measurement function is Period/Totalize
  Invalid
- When the measurement function is TI/Ratio x1/Ratio x16/Ratio x128/Ratio x1024
  IN1_Rise | IN1_Fall | IN2_Rise | IN2_Fall | IN3_Rise | IN3_Fall | IN4_Rise | IN4_Fall
- When the measurement function is Totalize(Gate)
  IN1 | IN2 | IN3 | IN4
- When the measurement function is UpDown1/UpDown2/UpDown4
  When Source-A is set to IN1-IN2
  Off | IN3 | IN4
  When Source-A is set to IN3-IN4
  Off | IN1 | IN2

Example (Visual Basic)

```
ret = WeSetControl (hMo, "CH1:ChSetS", "IN1_Rise")
' Set the input to be measured/slope of CH1 to IN1_Rise.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:ChSetS", value)
```

value → IN1_Rise

CH<x>:Limit (for counter mode only)

Description

Turns ON/OFF the cycle stop determination function of the channel or queries the current setting.

Parameters

Off | On

Example (Visual Basic)

```
ret = WeSetControl (hMo, "CH1:Limit", "On")
' Enable the cycle stop determination function of CH1.
Dim value As Variant
ret = WeGetControl (hMo, "CH1:Limit", value)
```

value → On

CH<x>:Reset (for counter mode only)

Description

Executes the counter reset of the channel. This command is valid when the measurement function is set to Totalize/UpDown1/ UpDown2/UpDown4.

Parameters

None

Example (Visual Basic)

```
Dim value As Variant
' Execute counter reset of CH1.
ret = WeGetControl (hMo, "CH1:Reset", value)
```

Trig:Source (for counter mode only)

Description

Sets the trigger source or queries the current setting.

Parameter

Input | Measure | BUSTRG

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Source", "Input")
' Set the input signal to be the trigger source.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Source", value)
```

value → Input

Trig:Input Source (for counter mode only)

Description

Sets the input to be used as the trigger source (when input signal is selected for the trigger source) or queries the current setting.

Parameter

IN1 to IN32

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Input Source", "IN1")
' Set the trigger source to Slot1-IN1.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Input Source", value)
```

value → IN1

Trig:Slope (for counter mode only)

Description

Sets the trigger slope when input signal is selected for the trigger source or queries the current setting.

Parameter

- When the acquisition mode is Triggered/Gate(Edge)
  Rise | Fall | Both | High | Low
- When the acquisition mode is Gate(Level)
  High | Low

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Slope", "Rise")
' Set the trigger slope to Rise.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Slope", value)
```

value → Rise

Trig:Measure Source (for counter mode only)

Description

Sets the measured value to be used as the trigger source (when measured value is selected for the trigger source) or queries the current setting.

Parameter

CH1 to CH32

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Measure Source", "CH1")
' Set the trigger source to Slot1-CH1.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Measure Source", value)
```

value → CH1

Trig:Type (for counter mode only)

Description

Sets the trigger type when measured value is selected for the trigger source or queries the current setting.

Parameter

The selectable parameters vary depending on the measurement function.

- When set to Totalize/Totalize(Gate)
  >= | ==
- When set to UpDown1/UpDown2/UpDown4
  >= | == | <=
- When set to a function other than above
  >= | <=

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Type", ">=")
' Set the trigger type to ">=."
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Type", value)
```

value → >=

Trig:Condition (for counter mode only)

Description

Sets the trigger condition when the bus trigger is selected for the trigger source or queries the current setting.

Parameter

Enter | Exit | Both | True | False

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Condition", "Enter")
' Sets the trigger condition to "Enter."
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Condition", value)
```

value → Enter

Trig:Threshold (for counter mode only)

Description

Sets the threshold level when measured value is selected for the trigger source or queries the current setting.

Parameter

The selectable range varies depending on the measurement function.

- When set to Period/TI
  100ns to 20s
  However, when the period stop deptermination function of the channel selected as trigger source is ON, the timeout time set up by Misc:Limit of Period to 20s.
- When set to Totalize/Totalize(Gate)
  0 to 536870911
- When set to UpDown1/UpDown2/UpDown4
  + | 268435456 to 268435455
- When set to Ratio x1
  0 to 536870911
- When set to Ratio x16
  0 to 33554431.9
- When set to Ratio x128
  0 to 4194303.99
- When set to Ratio x1024
  0 to 524287.999
- When set to Frequency
  0.05 Hz to 10 MHz
  However, when the period stop deptermination function of the channel selected as trigger source is ON, 0.05 Hz to the timeout time set up by Misc:Limit of Period.

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Threshold", "100E-9")
' Set the threshold level to 100 ns.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Threshold", value)
```

value → 100.0E-09

Trig:Pretrigger (for counter mode only)

Description

Sets the pretrigger or queries the current setting.

Parameter

0 to record length – 2

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Pretrigger", "1000")
' Set the pretrigger to 1000.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Pretrigger", value)
```

value → 1000

Trig:Hold Off (for counter mode only)

Description

Sets the trigger hold off or queries the current setting.

Parameter

During trigger mode: Record length to 1,048,576 (1M)
During freerun mode/gate (edge) mode: 2 to 1,048,576 (1M)

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Trig:Hold Off", "1000")
' Set the trigger hold off to 1000.
Dim value As Variant
ret = WeGetControl (hMo, "Trig:Hold Off", value)
```

value → 1000

Misc:TimeBase:Source (for counter mode only)

Description

Sets the time base or queries the current setting.

Parameter

Internal | Input | BUSCLK

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Misc:TimeBase:Source", "Internal")
' Set the time base to internal clock.
Dim value As Variant
ret = WeGetControl (hMo, "Misc:TimeBase:Source", value)
```

value → Internal

Misc:TimeBase:Input Source (for counter mode only)

Description
Select the input to be used as the time base (when time base is selected for the trigger source) or queries the current setting.

Parameter
IN1 to IN32

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Misc:TimeBase:Input Source", "IN1")
' Set the clock source to Slot1-IN1.
Dim value As Variant
ret = WeGetControl (hMo, "Misc:TimeBase:Input Source", value)
```
value → IN1

Misc:TimeBase:Slope (for counter mode only)

Description
Sets the slope of the clock or queries the current setting.

Parameter
Rise | Fall

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Misc:TimeBase:Slope", "Rise")
' Set the slope of the clock to Rise.
Dim value As Variant
ret = WeGetControl (hMo, "Misc:TimeBase:Slope", value)
```
value → Rise

Misc:DataHold (for counter mode only)

Description
Sets the data hold or queries the current setting.

Parameter
Off | On

Example (Visual Basic)
```
ret = WeSetControl (hMo, "Misc:DataHold", "On")
' Turn ON data hold.
Dim value As Variant
ret = WeGetControl (hMo, "Misc:DataHold", value)
```
value → On

Misc:HysType (for counter mode only)

Description
Sets the hysteresis direction or queries the current setting.

Parameter

Upper | Center | Lower

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Misc:HysType", "Upper")
' Set the hysteresis direction to Upper.
Dim value As Variant
ret = WeGetControl (hMo, "Misc:HysType", value)
```

value → Upper

Misc:Limit Of Period (for counter mode only)

Description

Sets the timeout time that is used by the cycle stop determination function or queries the current setting.

Parameters

Varies depending on the specified measurement function as follows.

- When the measurement function is not set to frequency measurement
  0.01 s to 20.00 s (10 ms resolution)
- When the measurement function of any channel is set to frequency measurement
  0.01 s, 0.02 s, 0.04 s, 0.08 s, 0.17 s, 0.34 s, 0.67 s, 1.34 s, 2.68 s, 5.37 s, 10.74 s

Example (Visual Basic)

```
ret = WeSetControl (hMo, "Misc:Limit Of Period", "0.01")
' Set the timeout time to 0.01 s.
Dim value As Variant
ret = WeGetControl (hMo, "Misc:Limit Of Period", value)
```

value → 10E-03

HysType (for time stamp mode only)

Description

Sets the hysteresis direction or queries the current setting.

Parameter

Upper | Center | Lower

Example (Visual Basic)

```
ret = WeSetControl (hMo, "HysType", "Upper")
' Set the hysteresis direction to Upper.
Dim value As Variant
ret = WeGetControl (hMo, "HysType", value)
```

value → Upper

Valid Acquisition Modes

Acquisition Type	Valid?
Triggered	Yes
Free Run	Yes
Gate(Level)	Yes
Gate(Edge)	Yes

Acquisition Restrictions

	Description
Memory length	1048576 (during counter mode) 4194304 (during time stamp mode)
Data length per block	During trigger/gate mode: 2 to memory length During free run mode: 1 to memory length * For the record length range, see the description of the Record Length command.
Number of blocks	During trigger/gate mode: 1 to 256 During free run mode: 1 to memory length
Number of Acquisitions	Data length per block > memory length/2: 1 Data length per block ≤ memory length/2: 1 to ∞ (set ∞ with a 0)

Valid Common Measurement Control API

API	Valid?	Note
WeStart	Yes	
WeStop	Yes	
WeStartSingle	Yes	(Counter mode)
WeStartWithEvent	Yes	When measurement function is UpDown1/UpDown2/
WeIsRun	Yes	UpDown4
WeLatchData	Yes	32-bit signed integer
WeGetAcqDataInfo	Yes	Other measurement functions
WeGetAcqData	Yes	32-bit unsigned integer
WeGetScaleData	Yes	(Time stamp mode)
WeGetMeasureParam	No	Upper 24 bits: Time data
WeSaveAcqData	Yes	Lower 8 bits: Channel change information
WeSaveScaleData	Yes	
WeSaveAsciiData	Yes	
WeGetCurrentData	Yes	Data is physical value in double-precision real format.
WeGetAcqDataEx	No	
WeGetAcqDataSize	Yes	
WeSaveAcqHeader	Yes	
WeSavePatternData	No	
WeLoadPatternData	No	
WeStartEx	Yes	
WeStopEx	Yes	

Valid Common Events

Event	Valid?
WE_EV_MEASEND	Yes
WE_EV_BLOCKEND	Yes
WE_EV_MEASABORT	Yes
WE_EV_TRIG_HIGH	No
WE_EV_TRIG_LOW	No
WE_EV_TRIG_PULSE	No
WE_EV_ALARM	No
WE_EV_CLOSE_MODULE_GUI	Yes

Module-specific Events

None

Module-specific Error Codes

Error Codes	Description
0x1801	The frequency of the signal input to slot 1 is too high.
0x1802	The frequency of the signal input to slot 2 is too high.
0x1803	The frequency of the signal input to slot 3 is too high.
0x1804	The frequency of the signal input to slot 4 is too high.
0x1805	The frequency of the signal input to slot 5 is too high.
0x1806	The frequency of the signal input to slot 6 is too high.
0x1807	The frequency of the signal input to slot 7 is too high.
0x1808	The frequency of the signal input to slot 8 is too high.
0x1810	Counter overflow occurred in slot 1 CH1.
0x1811	Counter overflow occurred in slot 1 CH2.
0x1812	Counter overflow occurred in slot 1 CH3.
0x1813	Counter overflow occurred in slot 1 CH4.
0x1814	Counter overflow occurred in slot 2 CH1.
0x1815	Counter overflow occurred in slot 2 CH2.
0x1816	Counter overflow occurred in slot 2 CH3.
0x1817	Counter overflow occurred in slot 2 CH4.
0x1818	Counter overflow occurred in slot 3 CH1.
0x1819	Counter overflow occurred in slot 3 CH2.
0x181a	Counter overflow occurred in slot 3 CH3.
0x181b	Counter overflow occurred in slot 3 CH4.
0x181c	Counter overflow occurred in slot 4 CH1.
0x181d	Counter overflow occurred in slot 4 CH2.
0x181e	Counter overflow occurred in slot 4 CH3.
0x181f	Counter overflow occurred in slot 4 CH4.
0x1820	Counter overflow occurred in slot 5 CH1.
0x1821	Counter overflow occurred in slot 5 CH2.
0x1822	Counter overflow occurred in slot 5 CH3.
0x1823	Counter overflow occurred in slot 5 CH4.
0x1824	Counter overflow occurred in slot 6 CH1.
0x1825	Counter overflow occurred in slot 6 CH2.
0x1826	Counter overflow occurred in slot 6 CH3.
0x1827	Counter overflow occurred in slot 6 CH4.
0x1828	Counter overflow occurred in slot 7 CH1.
0x1829	Counter overflow occurred in slot 7 CH2.
0x182a	Counter overflow occurred in slot 7 CH3.
0x182b	Counter overflow occurred in slot 7 CH4.
0x182c	Counter overflow occurred in slot 8 CH1.
0x182d	Counter overflow occurred in slot 8 CH2.
0x182e	Counter overflow occurred in slot 8 CH3.
0x182f	Counter overflow occurred in slot 8 CH4.

8.22 WE7562 Multichannel Analyzer Module

ASCII Commands

ASCII Command	Description
Start Mode	Sets the start mode or queries the current setting.
Page Up	Executes page up.
Cal Exec	Executes manual calibration.
Cal Exec:Status	Queries the execution status of the manual calibration.
Cal Exec:Result	Queries the execution result of the manual calibration.
Wave Monitor	Sets the trigger mode for input waveform monitoring or queries the current setting.

INPUT<X>:

ASCII Command	Description
INPUT<X>:On	Turns ON/OFF the target input or queries the current setting.
INPUT<X>:Operation Mode	Sets the operation mode of the target input or queries the current setting.
INPUT<X>:Measure Mode	Sets the measurement mode of the target input or queries the current setting.
INPUT<X>:Gain	Sets the total number of AD channels of the target input or queries the current setting.
INPUT<X>:Peak Detection:Pulse Width	Sets the window time for peak detection of the target input or queries the current setting.
INPUT<X>:Peak Detection:Error Check	Sets the error check for peak detection of the target input or queries the current setting.
INPUT<X>:Peak Detection:ROI Enabled	Sets the ROI judgment for peak detection of the target input or queries the current setting.
INPUT<X>:Peak Detection:LLD	Sets the lower level discriminator for peak detection of the target input or queries the current setting.
INPUT<X>:Peak Detection:ULD	Sets the upper limit discriminator for peak detection of the target input or queries the current setting.
INPUT<X>:Peak Detection:LLD:Scale	Sets the lower level discriminator (scaled value) for peak detection of the target input or queries the current setting.
INPUT<X>:Peak Detection:ULD:Scale	Sets the upper limit discriminator (scaled value) for peak detection of the target input or queries the current setting.
INPUT<X>:Measure Stop:Source	Sets the measurement stop source for the target input or queries the current setting.
INPUT<X>:Measure Stop:Time	Sets the measurement stop time for the target input or queries the current setting.
INPUT<X>:Measure Stop:Events	Sets the number of measurement stop events for the target input or queries the current setting.
INPUT<X>:Trigger:Source	Sets the trigger source of the target input or queries the current setting.
INPUT<X>:Trigger:Gate Function	Sets the gate function of the target input or queries the current setting.
INPUT<X>:Trigger:Output	Sets the trigger signal output destination of the target input or queries the current setting.
INPUT<X>:Trigger:Signal Type	Sets the output trigger signal type of the target input or queries the current setting.
INPUT<X>:Trigger:Synchronize	Sets the synchronized output of the trigger signal of the target input or queries the current setting.
INPUT<X>:Memory Size	Sets the channel size of the target input or queries the current setting (for PHA mode only).
INPUT<X>:No. Of Pages	Sets the number of measurement pages of the target input or queries the current setting (for PHA mode only).
INPUT<X>:Page Up	Executes page up of the target input (for PHA mode only).

ASCII Command	Description
INPUT<X>:GetCurrentPage	Queries the page number currently being measured (for PHA mode only).
INPUT<X>:Dwell Time	Sets the dwell time of the target input or queries the current setting (for MCS mode only).
INPUT<X>:No. Of Channels	Sets the number of measurement channels of the target input or queries the current setting (for MCS mode only).
INPUT<X>:Time Stamp	Sets the time stamp of the target input or queries the current setting (for LIST mode only).
INPUT<X>:Time Resolution	Sets the measurement time resolution of the target input or queries the current setting (for LIST mode only).
INPUT<X>:IsRun	Queries the measurement status of the target input.
INPUT<X>:GetCurrentEvents	Queries the current number of accumulated measurement events of the target input.
INPUT<X>:GetCurrentErrorEvents	Queries the current number of accumulated error events of the target input.
INPUT<X>:GetCurrentMeasureTime	Queries the current elapsed time of measurement of the target input.
INPUT<X>:GetTotalTriggerEvents	Queries the current number of accumulated trigger events (updated only during the measurement) of the target input.
INPUT<X>:GetTotalMeasureTime	Queries the current elapsed time of measurement since measurement start (updated only during the measurement) of the target input.
INPUT<X>:GetCurrentWaveData	Queries the current waveform data of the target input.

Note

Enter the input number in <x>. If the modules are linked, specify a serial number from the parent module. There are 2 inputs on each module. If two modules are linked and you wish to change the gain of input 2 of the second module, specify “INPUT4:Gain” in the command line.

Start Mode

Description

Sets the measurement start mode or queries the current setting.

Parameter

Normal | BUSTRG

Example (Visual Basic)

```
' Set the start mode to BUSTRG.
ret = WeSetControl (hMo, "Start Mode", "BUSTRG")
Dim value As Variant
ret = WeGetControl (hMo, "Start Mode", value)
```

value → BUSTRG

Page Up

Description
Executes page up.

Parameter
None

Note:
The target input for executing page up are those measured in PHA mode among all inputs of the linked modules.

Example (Visual Basic)
```
' Because there are no parameters, do not set the value.
Dim value As Variant
' Execute page up.
ret = WeSetControl (hMo, "Page Up", value)
```

Cal Exec

Description
Executes manual calibration.

Parameter
None

Note:
Do not carry out other operations while the manual calibration is in progress. This can lead to erroneous operation.
You can check the execution status of the manual calibration using “Cal Exec:Status.”

Example (Visual Basic)
```
' Because there are no parameters, do not set the value.
Dim value As Variant
' Execute manual calibration.
ret = WeSetControl (hMo, "Cal Exec", value)
```

Cal Exec:Status

Description
Queries the execution status of the manual calibration.

Parameter
0 (calibration completed) | 1 (manual calibration in progress)

Example (Visual Basic)
```
Dim value As Variant
' Query the execution status of the manual calibration.
ret = WeGetControl (hMo, "Cal Exec:Status", value)
```
value → 0

Cal Exec:Result

Description

Queries the execution result of the manual calibration.

Parameter (return value)

Returns a 32-bit unsigned integer.
Returns 0 if successful.
If an error occurs, the value returned will have a bit set that corresponds to the channel in which the error occurred.
The mapping of the input number to the bit is as follows:
bit0:INPUT1, bit1:INPUT2, bit2:INPUT3, •••• bit17:INPUT18

Example (Visual Basic)

```
Dim value As Variant
' Query the execution result of the manual calibration.
ret = WeGetControl (hMo, "Cal Exec:Results", value)
```

value → 0

Wave Monitor

Description

Sets the trigger mode for input waveform monitoring or queries the current setting.

Parameter

Auto | Normal

Example (Visual Basic)

```
' Set the trigger mode for input waveform monitoring to Normal.
ret = WeSetControl (hMo, "Wave Monitor", "Normal")
Dim value As Variant
ret = WeGetControl (hMo, "Wave Monitor", value)
```

value → Normal

INPUT<X>:On

Description

Turns ON/OFF the target input or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Enable the input signal of INPUT1.
ret = WeSetControl (hMo, "INPUT1:On", "On")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:On", value)
```

value → On

INPUT<X>:Operation Mode

Description
Sets the operation mode of the target input or queries the current setting.

Parameter
PHA | MCS | LIST

Note:
The operation mode can be set or queried independently for each input.

Example (Visual Basic)
```
' Set the operation mode of INPUT1 to PHA.
ret = WeSetControl (hMo, "INPUT1:Operation Mode", "PHA")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Operation Mode", value)
```
value → PHA

INPUT<X>:Measure Mode

Description
Sets the measurement mode of the target input or queries the current setting.

Parameter
The settings vary depending on the operation mode as follows:
For PHA mode: Cannot be set (will result in error).
For MCS mode: Peak | Counter
For LIST mode: Peak | Instantaneous

Example (Visual Basic)
```
' Set the measurement mode of INPUT1 to Peak.
ret = WeSetControl (hMo, "INPUT1:Measure Mode", "Peak")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Measure Mode", value)
```
value → Peak

INPUT<X>:Gain

Description
Sets the total number of AD channels of the target input or queries the current setting.

Parameter
512 | 1024 | 2048 | 4096 | 8192 | 16384

Example (Visual Basic)
```
' Set the total number of AD channels of INPUT1 to 1024.
ret = WeSetControl (hMo, "INPUT1:Gain", "1024")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Gain", value)
```
value → 1.0240E+03

INPUT<X>:Peak Detection:Pulse Width

Description

Sets the window time for peak detection of the target input or queries the current setting.

Parameter

1 µs to 10 ms (0.1-µs steps)

Example (Visual Basic)

```
' Set the window time for peak detection of INPUT1 to 10us.
ret = WeSetControl (hMo, "INPUT1:Peak Detection:Pulse Width", "10e-
6")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Peak Detection:Pulse Width",
value)
```

value → 10E-06

INPUT<X>:Peak Detection:Error Check

Description

Sets the error check for peak detection of the target input or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Turn ON the error check for peak detection of INPUT1.
ret = WeSetControl (hMo, "INPUT1:Peak Detection:Error Check", "On")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Peak Detection:Error Check",
value)
```

value → On

INPUT<X>:Peak Detection:ROI Enabled

Description

Sets the ROI judgment for peak detection of the target input or queries the current setting.

Parameter

Off | On

Example (Visual Basic)

```
' Turn ON the ROI judgment for peak detection of INPUT1.
ret = WeSetControl (hMo, "INPUT1:Peak Detection:ROI Enabled", "On")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Peak Detection:ROI Enabled",
value)
```

value → On

INPUT<X>:Peak Detection:LLD

Description

Sets the lower level discriminator for peak detection of the target input or queries the current setting.

Parameter

1 to the total number of AD channels selected using INPUT<X>:Gain – 7

Example (Visual Basic)

```
' Set the LLD for peak detection of INPUT1 to 5000.
ret = WeSetControl (hMo, "INPUT1:Peak Detection:LLD", "5000")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Peak Detection:LLD", value)
```

value → 5.000E+03

INPUT<X>:Peak Detection:ULD

Description

Sets the upper limit discriminator for peak detection of the target input or queries the current setting.

Parameter

1 to the total number of AD channels selected using INPUT<X>:Gain – 7

Example (Visual Basic)

```
' Set the ULD for peak detection of INPUT1 to 10000.
ret = WeSetControl (hMo, "INPUT1:Peak Detection:ULD", "10000")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Peak Detection:ULD", value)
```

value → 10.000E+03

INPUT<X>:Peak Detection:LLD:Scale

Description

Sets the lower level discriminator for peak detection of the target input using a scale value or queries the current setting.

Parameter

1 × scaled value to (the total number of AD channels selected using INPUT<X>:Gain – 7) × scaled value

Example (Visual Basic)

```
' Set the scaled LLD value for peak detection of INPUT1 to 5000.
ret = WeSetControl (hMo, "INPUT1:Peak Detection:LLD:Scale", "5000")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Peak Detection:LLD:Scale", value)
```

value → 5.000E+03

INPUT<X>:Peak Detection:ULD:Scale

Description

Sets the upper limit discriminator for peak detection of the target input using a scale value or queries the current setting.

Parameter

1 × scaled value to (the total number of AD channels selected using INPUT<X>:Gain – 7) × scaled value

Example (Visual Basic)

```
' Set the scaled ULD value for peak detection of INPUT1 to 10000.
ret = WeSetControl (hMo, "INPUT1:Peak Detection:ULD:Scale",
"10000")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Peak Detection:ULD:Scale", value)
```

value → 10.000E+03

INPUT<X>:Measure Stop:Source

Description

Sets the measurement stop source or page up source for the target input or queries the current setting.

Parameter

For PHA mode

None | Time | Events | Gate | OverFlow

For MCS mode

None only. Cannot be set (will result in error).

For LIST mode

None | Time | Events | Gate

Note:

If the operation mode is PHA,

- If the number of measurement pages (see INPUT<X>:No. Of Pages) is 1, this command sets the measurement stop source.
- If the number of measurement pages is 0 or 2, this command sets the page up source.

Example (Visual Basic)

```
' Set the measurement stop source of INPUT1 to Time.
ret = WeSetControl (hMo, "INPUT1:Measure Stop:Source", "Time")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Measure Stop:Source", value)
```

value → Time

INPUT<X>:Measure Stop:Time

Description

Sets the measurement stop time or page up time for the target input or queries the current setting.

Parameter

1 ms to 4294967295 ms (1-ms steps)

Note:

The time can be set regardless of the operation mode, but the setting takes effect only if the operation mode is PHA or LIST and the measurement stop (or page up) source is set to Time.

Example (Visual Basic)

```
' Set the measurement stop time of INPUT1 to 10 ms.
ret = WeSetControl (hMo, "INPUT1:Measure Stop:Time", "0.01")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Measure Stop:Time", value)
```

value → 10E-03

INPUT<X>:Measure Stop:Events

Description

Sets the number of measurement stop events or the number of page up events for the target input or queries the current setting.

Parameter

1 to 1099511627775

Note:

The time can be set regardless of the operation mode, but the setting takes effect only if the operation mode is PHA or LIST and the measurement stop (or page up) source is set to Events.

Example (Visual Basic)

```
' Set the number of measurement stop events of INPUT1 to 10000.
ret = WeSetControl (hMo, "INPUT1:Measure Stop:Events", "10000")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Measure Stop:Events", value)
```

value → 10.000E+03

INPUT<X>:Trigger:Source

Description

Sets the trigger source of the target input or queries the current setting.

Parameter

For INPUT1

Input1 | Input2 | Gate | Trigger2 | SCA2 | BUSTRG | Linked Module

For INPUT2

Input1 | Input2 | Gate | Trigger1 | SCA1 | BUSTRG | Linked Module

Example (Visual Basic)

```
' Set the trigger source of INPUT1 to BUSTRG.
ret = WeSetControl (hMo, "INPUT1:Trigger:Source", "BUSTRG")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Trigger:Source", value)
```

value → BUSTRG

INPUT<X>:Trigger:Gate Function

Description

Sets the gate function of the target input or queries the current setting.

Parameter

None | Enable | Disable

Example (Visual Basic)

```
' Enable the gate function of INPUT1.
ret = WeSetControl (hMo, "INPUT1:Trigger:Gate Function", "Enable")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Trigger:Gate Function", value)
```

value → Enable

INPUT<X>:Trigger:Output

Description

Sets the trigger signal output destination of the target input or queries the current setting.

Parameter

None | BUSTRG | Linked Module

Example (Visual Basic)

```
' Set the trigger signal output destination of INPUT1 to Linked
Module.
ret = WeSetControl (hMo, "INPUT1:Trigger:Output", "Linked Module")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Trigger:Output", value)
```

value → Linked Module

INPUT<X>:Trigger:Signal Type

Description

Sets the output trigger signal type of the target input or queries the current setting.

Parameter

Trigger | SCA

Example (Visual Basic)

```
' Set the output trigger signal type of INPUT1 to SCA.
ret = WeSetControl (hMo, "INPUT1:Trigger:Signal Type", "SCA")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Trigger:Signal Type", value)
```

value → SCA

INPUT<X>:Trigger:Synchronize

Description

Sets the synchronized output of the trigger signal of the target input or queries the current setting.

Parameter

Off (output for each trigger detection regardless of the measurement status) | On (output only during the measurement)

Example (Visual Basic)

```
' Set the trigger signal of INPUT1 to be output only during the
measurement.
ret = WeSetControl (hMo, "INPUT1:Trigger:Synchronize", "On")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Trigger:Synchronize", value)
```

value → On

INPUT<X>:Memory Size

Description

Sets the channel size of the target input or queries the current setting.

Parameter

16bit/CH | 32bit/CH

Note:

You can set this command regardless of the operation mode, but the setting takes effect only if the operation mode is PHA.
In addition, the selectable range of the number of measurement pages varies depending on the specified channel size.
For details, see INPUT<X>:No. Of Pages.

Example (Visual Basic)

```
' Set the channel size of INPUT1 to 16 bits per channel.
ret = WeSetControl (hMo, "INPUT1:Memory Size", "16bit/CH")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Memory Size", value)
```

value → 16bit/CH

INPUT<X>:No. Of Pages

Description

Sets the number of measurement pages of the target input or queries the current setting (for PHA mode only).

Parameter

The selectable range varies depending on the total number of AD channels and channel size that you selected as follows:

Total Number of AD Channels	Selectable Range of the Number of Pages	
	32bit/CH	16bit/CH
16384	1 to 64	1 to 128
8192	1 to 128	1 to 256
4096	1 to 256	1 to 512
2048	1 to 512	1 to 1024
1024	1 to 1024	1 to 2048
512	1 to 2048	1 to 4096

If the number of measurement pages is set to 0, continuous measurement (infinite number of measurements) is performed.
In this case, the page up source is only time. The selectable range is a value greater than equal to 100 ms.

Note:

You can set this command regardless of the operation mode, but the setting takes effect only if the operation mode is PHA.

Example (Visual Basic)

```
' Set the total number of measurement pages of INPUT1 to 10.
ret = WeSetControl (hMo, "INPUT1:No. Of Pages", "10")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:No. Of Pages", value)
```

value → 10

INPUT<X>:Page Up

Description

Executes page up of the target input.

Parameter

None

Note:

This command is valid only in PHA mode. An error occurs if you execute this command in another mode.

Example (Visual Basic)

```
' Because there are no parameters, do not set the value.
Dim value As Variant
' Execute page up of INPUT1.
ret = WeSetControl (hMo, "INPUT1:Page Up", value)
```

INPUT<X>:GetCurrentPage

Description
Queries the page number currently being measured of the target input.

Parameter (return value)
Returns a 32-bit signed integer.

Note:
This command is valid only in PHA mode. An error occurs if you execute this command in another mode.

Example (Visual Basic)
```
Dim value As Variant
' Query the page number of INPUT1.
ret = WeGetControl (hMo, "INPUT1:GetCurrentPage", value)
```
value → 1

INPUT<X>:Dwell Time

Description
Sets the dwell time of the target input or queries the current setting.

Parameter
The selectable range varies depending on the measurement mode as follows:

Peak mode	10 µs to 4294.967295 s (1-µs steps)
Counter mode	1 µs to 4294.967295s (1-µs steps)

Note:
You can set this command regardless of the operation mode, but the setting takes effect only if the operation mode is MCS.

Example (Visual Basic)
```
' Set the dwell time of INPUT1 to 1 ms.
ret = WeSetControl (hMo, "INPUT1:Dwell Time", "0.001")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Dwell Time", value)
```
value → 1.000E-03

INPUT<X>:No. Of Channels

Description
Sets the number of measurement channels of the target input or queries the current setting.

Parameter
1 to 524288 (1 steps)

Note:
You can set this command regardless of the operation mode, but the setting takes effect only if the operation mode is MCS.

Example (Visual Basic)

```
' Set the total number of measurement channels of INPUT1 to 1000.
ret = WeSetControl (hMo, "INPUT1:No. Of Channels", "1000")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:No. Of Channels", value)
```

value → 1000

INPUT<X>:Time Stamp

Description

Sets the time stamp of the target input or queries the current setting.

Parameter

Peak | Rise Edge

Note:

You can set this command regardless of the operation mode, but the setting takes effect only if the operation mode is LIST.

Example (Visual Basic)

```
' Set the time stamp of INPUT1 to Peak.
ret = WeSetControl (hMo, "INPUT1:Time Stamp", "Peak")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Time Stamp", value)
```

value → Peak

INPUT<X>:Time Resolution

Description

Sets the measurement time resolution of the target input or queries the current setting.

Parameter

100ns | 200ns | 500ns | 1µs | 5µs | 10µs | 100µs | 1ms

Note:

You can set this command regardless of the operation mode, but the setting takes effect only if the operation mode is LIST.

Example (Visual Basic)

```
' Set the measurement time resolution of INPUT1 to 1us.
ret = WeSetControl (hMo, "INPUT1:Time Resolution", "1us")
Dim value As Variant
ret = WeGetControl (hMo, "INPUT1:Time Resolution", value)
```

value → 1.0E-6

INPUT<X>:IsRun

Description
Queries the measurement status of the target input.

Parameter (return value)
The result of the measurement status is returned using an 8-bit unsigned integer.
The meaning of the returned value is as follows:
0 (measurement completed) | 1 (measurement in progress)

Example (Visual Basic)
```
Dim value As Variant
' Query the measurement status of INPUT1.
ret = WeGetControl (hMo, "INPUT1:IsRun", value)
```
value → 0

INPUT<X>:GetCurrentEvents

Description
Queries the current number of accumulated measurement events of the target input.

Parameter (return value)
The current number of measurement events is returned using a double-precision real number.

Note:
The counter value is reset to zero each time the measurement is started. The number of events after the measurement is started is counted.

Example (Visual Basic)
```
Dim value As Variant
' Query the number of measurement events of INPUT1.
ret = WeGetControl (hMo, "INPUT1:GetCurrentEvents", value)
```
value → 10.000E+03

INPUT<X>:GetCurrentErrorEvents

Description
Queries the current number of accumulated error events of the target input.

Parameter (return value)
The current number of error events is returned using a double-precision real number.

Note:
The counter value is reset to zero each time the measurement is started. The number of error events after the measurement is started is counted.

Example (Visual Basic)
```
Dim value As Variant
' Query the number of error events of INPUT1.
ret = WeGetControl (hMo, "INPUT1:GetCurrentErrorEvents", value)
```
value → 10.000E+03

INPUT<X>:GetCurrentMeasureTime

Description

Queries the current elapsed time of measurement of the target input.

Parameter (return value)

The current elapsed time of measurement is returned using a double-precision real number. The unit is ms.

Note:

The counter value is reset to zero each time the measurement is started. The elapsed time after the measurement is started is counted.

Example (Visual Basic)

```
Dim value As Variant
' Query the elapsed time of measurement of INPUT1.
ret = WeGetControl (hMo, "INPUT1:GetCurrentMeasureTime", value)
```

value → 1.000E+03

INPUT<X>:GetTotalTriggerEvents

Description

Queries the current number of accumulated trigger events (updated only during the measurement) of the target input.

Parameter (return value)

The current number of accumulated trigger events is returned using a double-precision real number.

Note:

The counter value is reset to zero each time the measurement is started. The number of trigger events from the start of the measurement is counted. However, counting stops when the measurement is stopped, and the count value is held.

Example (Visual Basic)

```
Dim value As Variant
' Query the number of accumulated trigger events of INPUT1.
ret = WeGetControl (hMo, "INPUT1:GetTotalTriggerEvents", value)
```

value → 10.000E+03

INPUT<X>:GetTotalMeasureTime

Description

Queries the current elapsed time of measurement (updated only during the measurement) of the target input.

Parameter (return value)

The current elapsed time of measurement is returned using a double-precision real number. The unit is ms.

Note:

The counter value is reset to zero each time the measurement is started. The elapsed time from the start of the measurement is counted. However, counting stops when the measurement is stopped, and the count value is held.

Example (Visual Basic)

```
Dim value As Variant
' Query the elapsed time of measurement of INPUT1.
ret = WeGetControl (hMo, "INPUT1:GetTotalMeasureTime", value)
```

value → 1.000E+03

INPUT<X>:GetCurrentWaveData

Description

Queries the current input waveform data of the target input.

Parameter (return value)

The current input waveform data is returned using 2048 points of 16-bit unsigned integers

Note:

From the data that is loaded, the data value corresponding to the peak value within the peak detection period will have its highest bit set.

Example (Visual Basic)

```
Dim buf(2047) As Integer
Dim wave(2047) As Long
Dim i As Integer
' Query the waveform data applied to INPUT1.
ret = WeGetControlEx (hMo, "INPUT1:GetCurrentWaveData", WE_UWORD,
2048, buf(0))
For i = 0 To 2047
    If buf( i ) < 0 Then
        wave( i ) = buf( i ) + 65536
    Else
        wave( i ) = buf( i )
    End If
    wave( i ) = wave( i ) Mod 16384
Next i
```

Acquisition Restrictions

Item	Description
Memory length	4 MB per input
Number of channels per page	PHA mode: 512 to 16384(2n steps) MCS mode: 1 to 524288 LIST mode: 1 to 524288 * There is no concept of pages in MCS and LIST modes. The number of measurement channels or the number of data values that can be stored in the memory is indicated.
Number of Pages	PHA mode: 1 to 4M/(total number of AD channels × channel size) MCS or LIST mode: 1
Number of measurement pages	PHA mode: 1 to the number of pages. Specify zero for continuous measurement. MCS or LIST mode: Invalid

Valid Common Measurement Control API

API	Valid?	Description
WeStart	Yes	
WeStop	Yes	
WeStartEx	No	
WeStopEx	No	
WeStartSingle	Yes	The operation of this API does not stop until the measurement of all inputs is complete. If the measurement takes a long time even for a single channel, the program executing this API is forced to wait.
WeStartWithEvent	Yes	The WE_EV_MEASEND event is issued when the measurement of all inputs is complete.
WeIsRun	Yes	The meaning of the returned value of the parameter changes as follows: Status: 0 All inputs are stopped 1 Measurement status (when any of the inputs is in the measurement status)
WeLatchData	Yes	Usable only in MCS or LIST mode. MCS and LIST modes support the latch measurement of free run mode. For details on latch measurements, see section 4.3, “Relationship between the Acquisition Mode and the Data Collection Method.”
WeGetAcqData	Yes	The meaning of the following parameters changes as follows: ch: Input number (counted from 1). The value –1 (all inputs) cannot be specified. blockNo: The read source page number for PHA mode. If you set a value of –1, the page currently being measured is read. For MCS or LIST mode, set this value to -1. The data of the area selected by the previous latch operation is read. To read all the measured channels or data, specify 0x80000000.
WeGetAcqDataInfo	Yes	The meaning of parameters ch and blockNo is the same as with WeGetAcqData.
WeGetAcqDataSize	Yes	The meaning of parameters ch and blockNo is the same as with WeGetAcqData.
WeGetAcqDataEx	Yes	The meaning of the following parameters changes as follows: The meaning of parameters ch and blockNo is the same as with WeGetAcqData. startPoint: The start position of the channel to be read The value –1 specifies the first channel. stopPoint: The end position of the channel to be read The value –1 specifies the end channel. Parameters ppNum and Interpolation are not supported.
WeGetCurrentData	Yes	An error occurs for PHA and MCS modes. Returns the latest instantaneous value only during LIST mode (WE_ULLONG: upper 16 bits: pulse height value data, lower 48 bits: time stamp data).
WeGetScaleData	No	
WeGetScaleDataEx	No	
WeGetMeasureParam	No	
WeSaveAcqData	Yes	The meaning of parameters ch and blockNo is the same as with WeGetAcqData.
WeSaveScaleData	No	
WeSaveScaleDataEx	No	
WeSaveAsciiData	Yes	The meaning of parameters ch and blockNo is the same as with WeGetAcqData.
WeSaveScaleAsciiData	No	
WeSaveScaleAsciiDataEx	No	
WeSaveAcqHeader	Yes	The meaning of parameters ch and blockNo is the same as with WeGetAcqData.
WeSavePatternData	No	
WeLoadPatternData	No	

Valid Common Events

Event	Valid?	Description
WE_EV_MEASEND	Yes	Issued when the measurement of all inputs is complete.
WE_EV_BLOCKEND	No	Issues a page up event for each input. See the module-specific events.
WE_EV_MEASABORT	Yes	
WE_EV_TRIG_HIGH	No	
WE_EV_TRIG_LOW	No	
WE_EV_TRIG_PULSE	No	
WE_EV_CLOSE_MODULE_GUI	Yes	

Module-specific Events

Event	Event Description
0x00010000	Measurement of one of the inputs being measured is complete
0x00020000	Page up occurred in one of the inputs being measured
0x00040000	Overrun occurred in one of the inputs being measured

The input number in which measurement is completed or a page up occurred is stored in bits 20 to 24 in the first parameter of WeEvent (input number is counted from 1).

(Example)

Event	Event Description
0x00110000	Measurement is complete in INPUT1
0x00a10000	Measurement is complete in INPUT10
0x00220000	Page up occurred in INPUT2
0x01020000	Page up occurred in INPUT16
0x00340000	Overrun occurred in INPUT3
0x01240000	Overrun occurred in INPUT18

Module-specific Error Codes

None

9.1 Basic Model

For the write operation of measured data, functions are provided for the following two assumed models: one in which data blocks are added in order to a single file (Single File Model) and another in which a file is created for each data block (Sequential File Model).
For the read operation of measured data, functions are provided for the following two assumed models: one in which data is read by specifying a block from a file in which data is stored in multiple blocks (Single File Model) and another in which data is read by specifying the number of samples from multiple files (Sequential File Model).

9.2 File Format

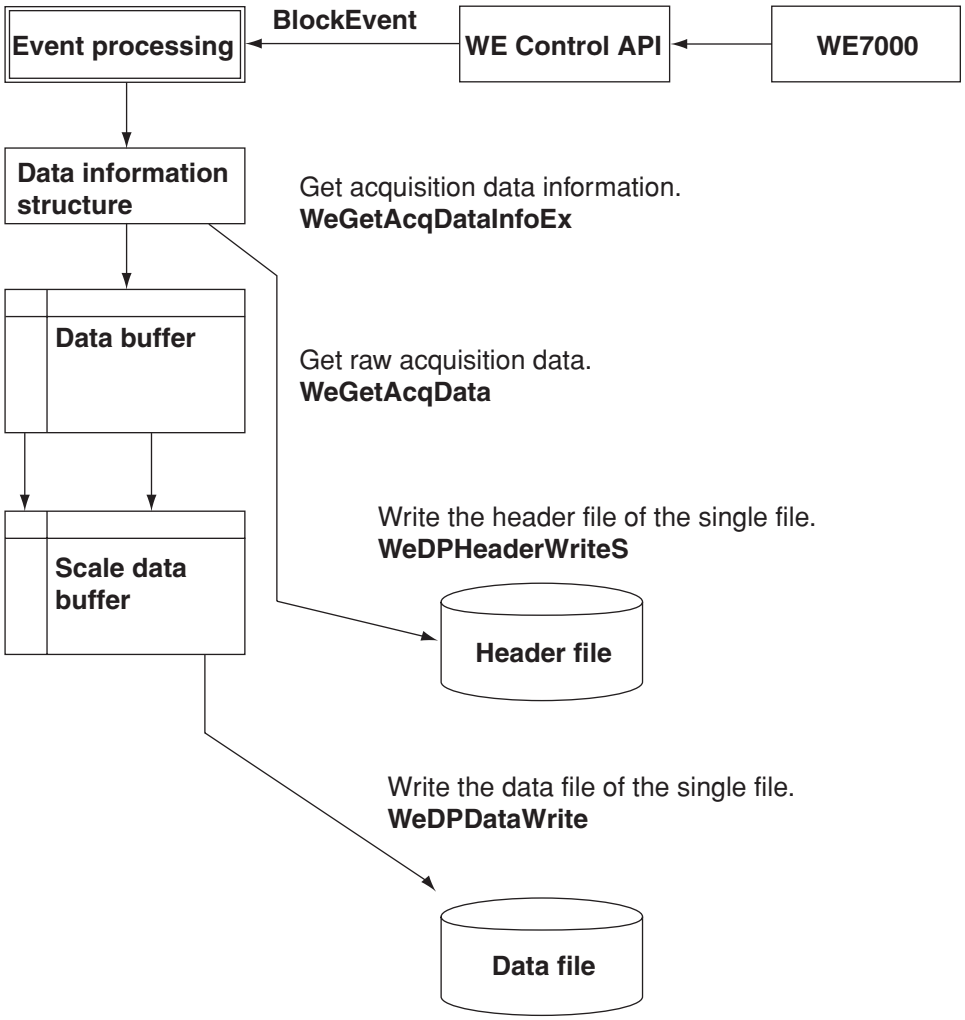
The files are in YOKOGAWA's proprietary format (the same format as the measured data that is saved in binary format using the WE7000 Control Software (.wvf file)).
There are three types of storage formats of .wvf files: block type, trace type, and scan type. This API can be used to read the measured data without being aware of the storage format of .wvf files. However, when reading a Scan type file, the entire data is assumed to be a single block. This API stores the data using the block type of the .wvf format.

Note

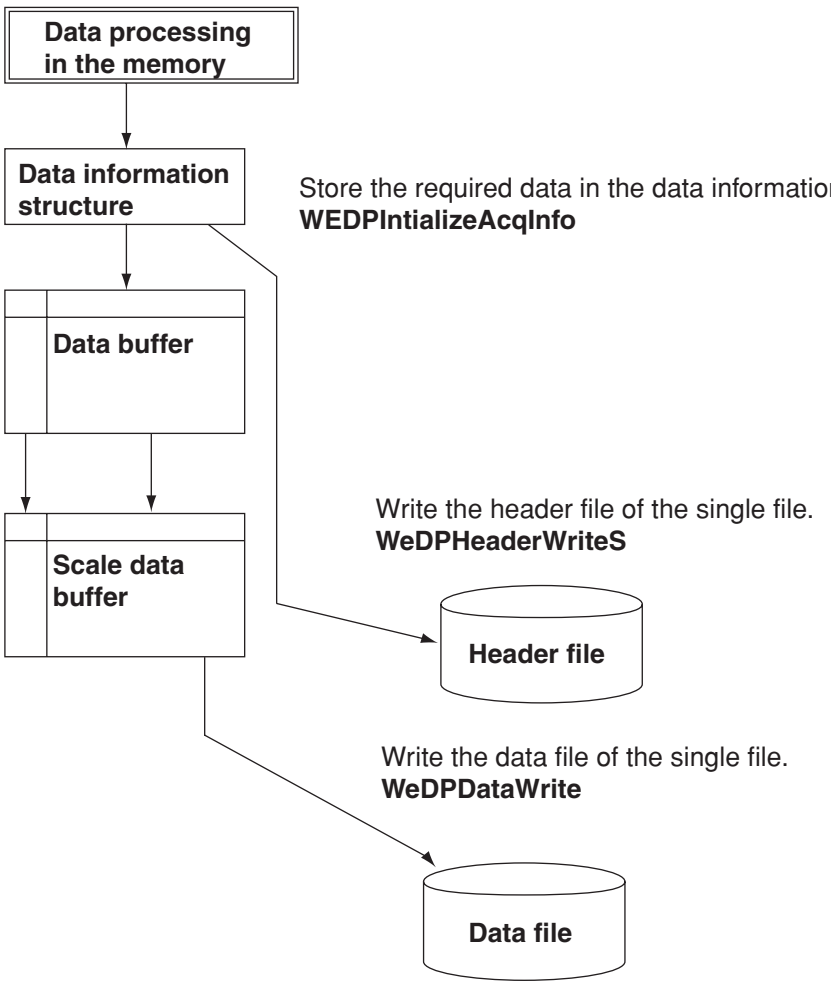
For details on .wvf files, see technical information (TI7000-21E) issued by YOKOGAWA.

9.3 Model When Performing Write Operation of Measured Data

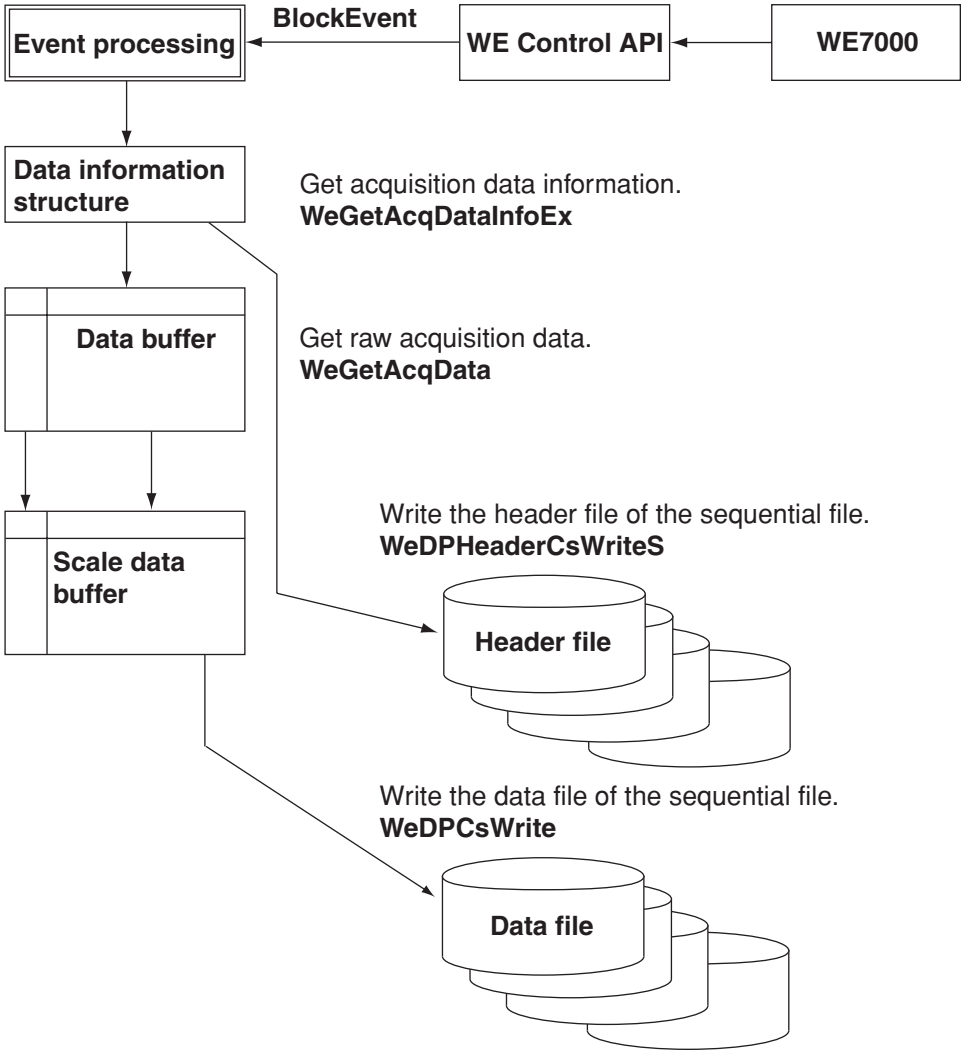
Single File Model for WE7000 Measurement Data (Multiple blocks are stored to the same file)



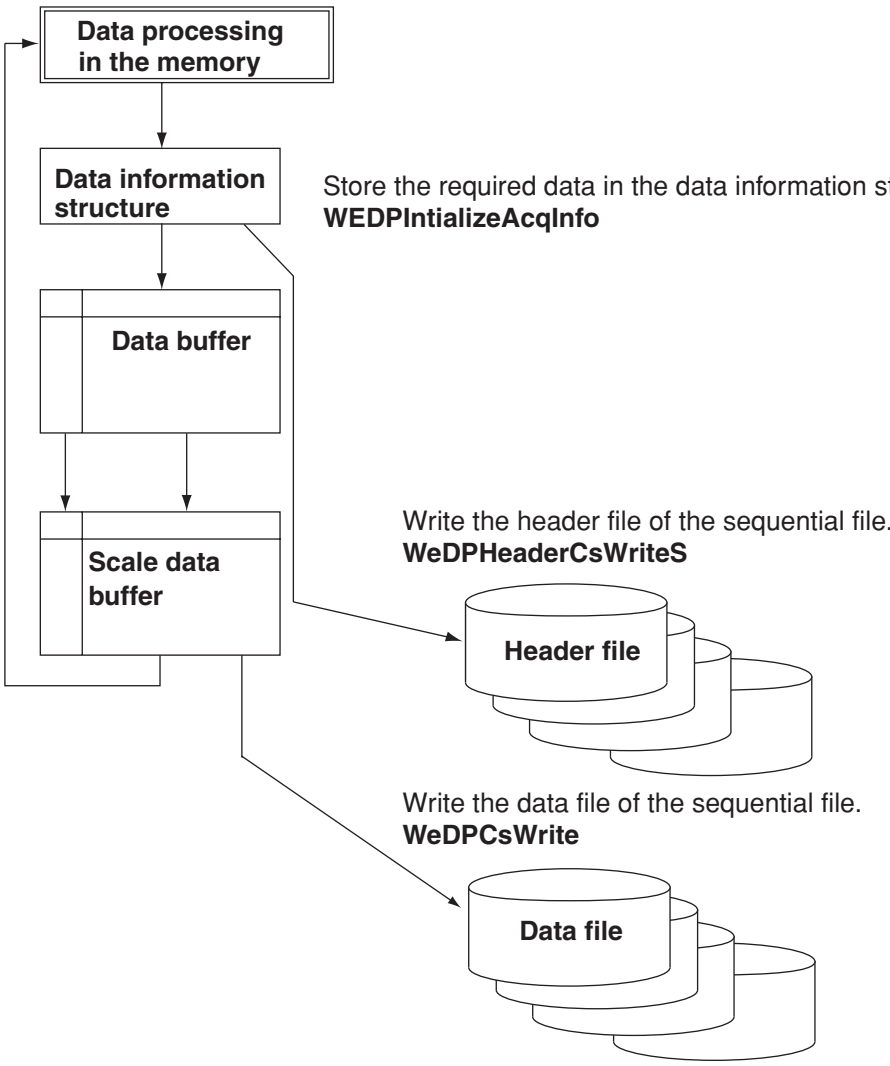
Single File Model for Data Created Using an Application Program (Multiple blocks are stored to the same file)



Sequential Data Model for WE7000 Measurement Data (Stored to separate files for each block)



Creating Sequential Files (Store data that has been created using a program to a single file for each block)



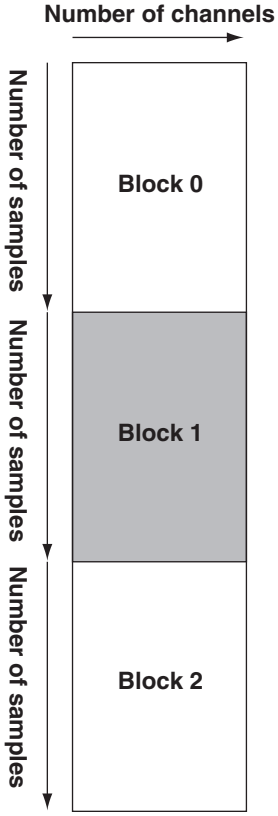
9.4 Model When Performing Read Operation of Measured Data

Single File Model (Read data by specifying blocks)

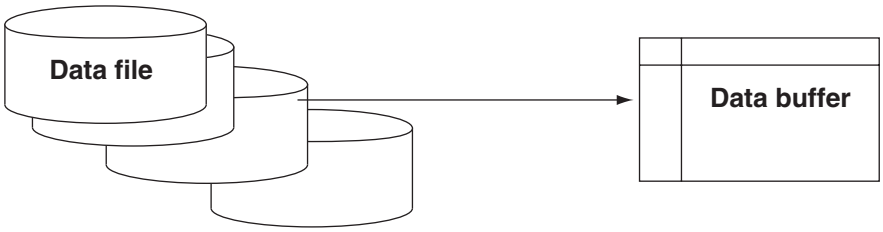
WeDPDataRead(FileName,BlockNo,ChNo,DataForm,DataBuff)

FileName: File name (excluding the sequential number section).
BlockNo: Block number (0 to n).
ChNo: Channel number. Specify -1 to read all channels.
DataForm: Data type.
DataBuf: Pointer to the buffer in which the data is to be stored.

If you specify WeDPDataRead("FileName",1,-1,DATA_SINGLE,DataBuff), the data in the shaded section in the following figure is stored to DataBuff.



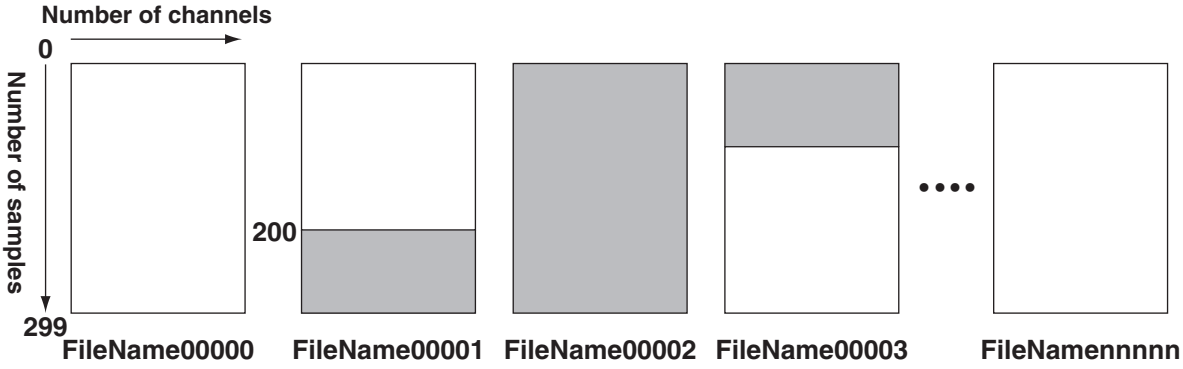
Sequential File Model (Read data by specifying the number of samples)



WeDpCsRead(FileName,SeriesNo,Start,Length,ChNo,DataForm,DataBuff)

FileName: File name (excluding the sequential number section).
SeriesNo: First sequence number.
Start: Start sample number.
Length: Number of data points per channel to be read.
ChNo: Channel number. Specify -1 to read all channels.
DataForm: DataBuff type.
DataBuff: Pointer to the buffer in which the data is to be stored.

If you specify WeDpCsRead("FileName",1,200,500,-1,DATA_SINGLE,DataBuff), the data in the shaded section in the following figure is stored to DataBuff.



This file has “4 channels × 300 samples” of data stored.
Data is stored to the memory after scaling with VResolution and VOffset in the .hdr file.

Search Method of Sequential Files

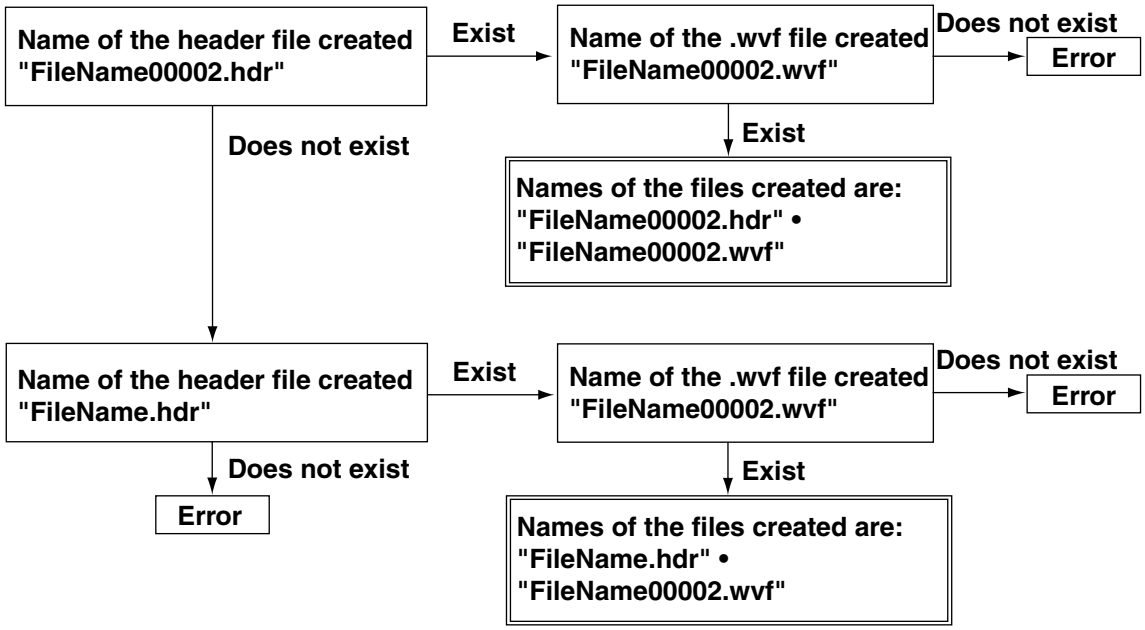
- When the sequence number is set to -1

WeDPCsRead("FileName",-1,200,500,-1,WE_FLOAT,DataBuff)

Names of the files created are:
"FileName.hdr" • "FileName.wvf"

- When the sequence number is set to a positive integer

WeDPCsRead("FileName",2,200,500,-1,WE_FLOAT,DataBuff)



Reading Header Files

- Name of the header file for single files

  WeDPHeaderReadS(FileName,BlockNo,ComBuff(),ChBuff())

  WeDPHeaderReadS("FileName", 2,ComBuff(),ChBuff())
  FileName: "FileName"
  BlockNo: 2
  ComBuff(): ComBuff()
  ChBuff(): When ChBuff()

  Set to FileName.hdr.

- Name of the header file for sequential files

  WeDPHeaderCsReadS(FileName,SeriesNo,ComBuff(),ChBuff())

  WeDPHeaderCsReadS("FileName",2,ComBuff(),ChBuff())
  FileName: "FileName"
  SeriesNo: 2
  ComBuff(): ComBuff()
  ChBuff(): When ChBuff()

  Set to FileName00002.hdr.

  WeDPHeaderCsReadS("FileName",-1,ComBuff(),ChBuff())

  FileName: "FileName"
  SeriesNo: -1
  ComBuff(): ComBuff()
  ChBuff(): When ChBuff()

  Set to FileName.hdr.

9.5 Common Information Structure and Channel Information Structure

User-Defined Type

When the X-axis data is the same on all channels, the data structure is as follows:

- Common information

```
Type CommonInf1
Comment As String*256  // Comment (set up to 255 bytes)
SampleNum As Long      // Number of samples
ChNum As Long // Number of channels
SamplingInterval As Double        // Sampling interval
PreTrigger As Long        // Amount of pretrigger
XUnit As String*16        // X-axis unit
Date As String*16         // Measurement date (set up to 15 bytes)
Time As String*16         // Measurement time (set up to 15 bytes)
End Type
```

- Channel information

```
Type ChanelInf1
ChName AS String*16    // Channel name (set up to 15 bytes)
ScaleA As Double       // Scaling coefficient A
ScaleB As Double       // Scaling coefficient B
Unit As String*16      // Unit (set up to 15 bytes)
End Type
```

Note

The current version does not allow handling of files in which the X-axis data is not the same for all channels.

Correspondence between User-Defined Type and Header File Items

User-Defined Type	Header File Item
Comment	Character string in the comment line
SampleNum	Sum of BlockSize for the number of blocks (information of the first channel)
ChNum	TraceTotalNumber value
SamplingInterval	HResolution value (information of the first channel)
PreTrigger	HOffset value (information of the first channel)
XUnit	HUnit value (information of the first channel)
Date	Date value (information of the first channel)
Time	Time value (information of the first channel)
ChName	TraceName value (for each channel)
ScaleA	Linear VResolution value (for each channel)
ScaleB	Linear VOffset value (for each channel)
Unit	Linear Unit value (for each channel)

Correspondence between Acquisition Data Information and Header File Items

The correspondence between acquisition items (acquisition data information that can be read using WeGetAcqDataInfo (page 6-66)) and the header file items is as follows:

Acquisition Data Information Item	Header File Item
Comment	Character string in the comment line
Channel	–
DataType	VdataType
BlockNum	BlockNumber
StartBit	–
EffectiveBit	–
TrigActive	–
Record	BlockSize
Recordlen	–
TrigPosition	HOffset
Interval	HResolution
VResolution	VResolution
VOffset	VOffset
TrigLevel	–
TrigWidth	–
PusOverData	VPlusOverData
MinusOverData	VMinusOverData
NonData	VIllegalData
DispMaxData	VMaxData
DispMinData	VMinData

The settings of the items of the header file that is created when writing measured data using this API are indicated in the table below.

Item Name	Setting
FormatVersion	1.01
Model	WE7000
Endian	Ltl
DataFormat	Block
GroupNumber	Automatically created to match the stored data when writing measured data
TraceTotalNumber	Automatically created to match the stored data when writing measured data
DataOffset	0
TraceNumber	Automatically created to match the stored data when writing measured data
BlockNumber	Automatically created to match the stored data when writing measured data
TraceName	Set to Ch1 to ChN to match the stored data when writing measured data
BlockSize	Record of acquisition data information
VResolution	VResolution of acquisition data information
VOffset	VOffset of acquisition data information
VdataType	Set from parameters when writing measured data
VUnit	–
VplusOverData	PlusOverData of acquisition data information
VminusOVerData	MinusOverData of acquisition data information
VillegalData	NonData of acquisition data information
VMaxData	DispMaxData of acquisition data information
VMinData	DispMinData of acquisition data information
HResolution	SamplingInterval of CommonInf1
HOffset	PreTrigger of CommonInf1
HUnit	XUnit of CommonInf1
Date	Date of CommonInf1
Time	Time of CommonInf1
PLinearSlope	ScaleA of CommonInf1
PLinearIntercept	ScaleB of CommonInf1
PLinearUnit	Unit of CommonInf1
PTraceName	ChannelName of CommonInf1
PLinearMode	"OFF"

9.6 The List of Functions

Single File Access

Function Name	Description	Page
WeDPHeaderReadS	Read the header file of the single file.	9-13
WeDPDataRead	Read the data file of the single file.	9-14
WeDPHeaderWriteS	Write the header file of the single file.	9-15
WeDPDataWrite	Write the data file of the single file.	9-16

Sequential File Access

Function Name	Description	Page
WeDPHeaderCsReadS	Read the header file of the sequential file.	9-17
WeDPCsRead	Read the data file of the sequential file.	9-18
WeDPHeaderCsWriteS	Write the header file of the sequential file.	9-19
WeDPCsWrite	Write the data file of the sequential file.	9-20

Access the Specified Item of the Header File

Function Name	Description	Page
WeDPHeaderItemRead	Read the data of the specified item.	9-21
WeDPHeaderItemWrite	Write the data of the specified item.	9-22

Data Operation

Function Name	Description	Page
WeDPGetSampleChNum	Get the number of samples and number of channels.	9-23
WeDPGetBlockNum	Get the number of blocks.	9-24
WeDPInitializeAcqInfo	Store the required data in the data information structure.	9-25

Read Acquisition Data Information

Function Name	Description	Page
WeGetAcqDataInfoEx	Get acquisition data information.	9-26

9.7 Single File Access

WeDPHeaderReadS

Description

Reads the data from the header file by specifying the block number.

Syntax

WeDPHeaderReadS(ByVal *FileName* As String,ByVal *BlockNo* As Long,ByRef *ComBuff()* As CommonInf1,ByRef *ChBuff()* As ChanelInf1)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Header file name.
BlockNo (IN)	Block number. Specify the block number you wish to read.
ComBuff() (OUT)	Common information structure. Define using an array.
ChBuff() (OUT)	Channel information structure. Define using an array.

Note:

Collectively retrieves the header file information for data of which the number of X-axis data points is the same. Prepare a single-element array for the common information structure buffer. Prepare channel information structure buffer for the amount equal to the number of channels.

Example (Visual Basic)

```
Dim ret As Long
Dim FileName As String
Dim BlockNo As Long
Dim ComBuff(0) As CommonInf1
Dim ChBuff(5) As ChanelInf1
FileName = "TestData1"
BlockNo = 2
ret = WeDPHeaderReadS(FileName,BlockNo,ComBuff(),ChBuff())
If 0 <> ret Then
   MsgBox "File read error"
End If
```

WeDPDataRead

Description

Reads the data from the data file by specifying the block number.

Syntax

WeDPDataRead(ByVal *FileName* As String,ByVal *BlockNo* As Long,ByVal *ChNo* As Long,ByVal *DataForm* As Long,ByRef *DataBuff* As Any)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Data file name. Specify the file name excluding the extension.
BlockNo (IN)	Block number.
ChNo (IN)	Channel number. -1 represents all channels. Counted from 1. If you specify -1, pass a buffer defined using a two-dimensional array for DataBuff.
DataForm (IN)	DataBuff type. Specify BYTE (char), Integer (short), Long (long), Single (float), or Double (double). No other data types can be passed. The type inside the parentheses indicates the type for the C Language.
DataBuff (OUT)	Data storage buffer. Pass the buffer in the type specified by DataForm.

Note:

Reads the data from the data file in units of blocks. The files cannot be handled as sequential files. Specify DataForm from the following defined data.

BYTE:	WE_UBYTE
Integer:	WE_SWORD
Long:	WE_SLONG
Single:	WE_FLOAT
Double:	WE_DOUBLE

Example (Visual Basic)

```
Dim FileName As String
Dim BlockNo As Long
Dim ChNo As Long
Dim DataForm As Long
Dim DataBuff(999,3) As Single
FileName = "TestData1"
BlockNo = 2
ChNo = -1
DataForm = WE_FLOAT
ret = WeDPDataRead(FileName,BlockNo,ChNo,DataForm,DataBuff(0,0))
If 0 <> ret Then
   MsgBox "File read error"
End If
```

WeDPHeaderWriteS

Description

Writes the header information at once to the header file by specifying the block.

Syntax

WeDPHeaderWriteS(ByVal *FileName* As String,ByVal *BlockNo* As Long,ByRef *ComBuff()* As CommonInf1,ByRef *ChBuff()* As ChanelInf1,ByRef *AcqInfo* As AcqDataInfEx)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Header file name.
BlockNo (IN)	Block number. Specify the block number to be written.
ComBuff() (IN)	Common information structure. Define using an array.
ChBuff() (IN)	Channel information structure. Define using an array.
AcqInfo (IN)	Data information structure (AcqDataInfoEx).

Note:

Collectively writes the header file information for data of which the number of X-axis data points is the same. Prepare a single-element array for the common information structure buffer. Prepare channel information structure buffer for the amount equal to the number of channels. Pass the data obtained by WeGetAcqDataInfoEx() or WeAcqInfoInitialize() to AcqInfo. Data is written to the specified block.

Example (Visual Basic)

```
Dim ret As Long
Dim FileName As String
Dim BlockNo As Long
Dim CommonBuff(0) As CommonInf1
Dim ChBuff(5) As ChanelInf1
Dim AcqInfo(5) As AcqDataInfoEx
Dim InfoNum As Long
BlockNo=2
Ret = WeGetAcqDataInfoEx(hMo,-1,BlockNo,AcqInfo(),infoNum)
If 0 <> ret Then
   MsgBox "Data information read error"
EndIf
FileName = "TestData1"
ret=WeDPHeaderWriteS(FileName,BlockNo,ComBuff(),ChBuff(),AcqInfo())
If 0 <> ret Then
   MsgBox "File write error"
End If
```

WeDPDataWrite

Description

Writes the data to the data file in units of blocks.

Syntax

WeDPDataWrite(ByVal *FileName* As String,ByVal *BlockNo* As Long,ByVal *SampleNum* As Long,ByRef *AcqInfo* As AcqDataInfoEx,ByVal *DataForm* As Long,ByRef *DataBuff* As Any)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Data file name. Specify the file name excluding the extension. A .wvf extension is added to the created file.
BlockNo (IN)	Block number.
SampleNum (IN)	Number of samples.
AcqInfo (IN)	Data information structure. Pass the data that has been returned by WeGetAcqDataInfoEx() or WeDPInitializeAcqInfo().
DataForm (IN)	DataBuff type. Specify BYTE (char), Integer (short), Long (long), Single (float), or Double (double). No other data types can be passed. The type inside the parentheses indicates the type for the C Language.
DataBuff (OUT)	Data storage buffer. Pass the buffer in the type specified by DataForm. Pass a buffer defined using a two-dimensional array.

Note:

Write data to a data file. The files cannot be handled as sequential files. Specify DataForm from the following defined data.

BYTE:	WE_UBYTE
Integer:	WE_SWORD
Long:	WE_SLONG
Single:	WE_FLOAT
Double:	WE_DOUBLE

Example (Visual Basic)

```
Dim FileName As String
Dim BlockNo As Long
Dim SampleNum As Long
Dim AcqInfo(1) As AcqDataInfoEx
Dim DataForm As Long
Dim DataBuff(999,1) As Single
Dim InfoNum As Integer
BlockNo = 2
SampleNum = 1000
WeGetAcqDataInfoEx(hMo,-1,BlockNo,AcqInfo(),InfoNum)
FileName = "TestData1"
DataForm = WE_FLOAT
ret = WeDPDataWrite(FileName,BlockNo,SampleNum,AcqInfo(),DataForm,
DataBuff(0,0))
If 0 <> ret Then
   MsgBox "File write error"
End If
```

9.8 Sequential File Access

WeDPHeaderCsReadS

Description

Collectively reads the header information from a header file.

Syntax

WeDPHeaderCsReadS(ByVal *FileName* As String,ByVal *SeriesNo* As Long, ByRef *ComBuff()* As CommonInf1,ByRef *ChBuff()* As ChanelInf1)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Header file name.
SeriesNo (IN)	Sequence number when creating sequence files. If you specify -1, FileName becomes the file name as-is.
ComBuff() (OUT)	Common information structure. Define using an array.
ChBuff() (OUT)	Channel information structure. Define using an array.

Note:

Collectively retrieves the header file information for data of which the number of X-axis data points is the same. Prepare a single-element array for the common information structure buffer.

Prepare channel information structure buffer for the amount equal to the number of channels.

Example (Visual Basic)

```
Dim ret As Long
Dim FileName As String
Dim SeriesNo As Long
Dim ComBuff(0) As CommonInf1
Dim ChBuff(5) As ChanelInf1
FileName = "TestData1"
BlockNo = 2
ret = WeDPHeaderCsReadS(FileName,SeriesNo,ComBuff(),ChBuff())
If 0 <> ret Then
   MsgBox "File read error"
End If
```

WeDPCsRead

Description

Reads the data from the data files (sequential files) by specifying the number of samples.

Syntax

WeDPCsRead(ByVal *FileName* As String,ByVal *SeriesNo* As Long,ByVal *Start* As Long,ByVal *Length* As Long,ByVal *ChNo* As Long,ByVal *DataForm* As Long,BYRef *DataBuff* As Any)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Data file name. Specify the file name excluding the extension.
SeriesNo (IN)	Sequence number when reading sequential files. If you specify -1, only the file with the name specified by FileName is read.
Start (IN)	Start sample number. Specify the first sample number to be retrieved using a value greater than or equal to 0.
Length (IN)	Number of samples to be retrieved. -1 specifies all data after the sample specified by Start. An error occurs if the specified value is greater than the number of samples that is stored.
ChNo (IN)	Channel number. -1 specifies all channels. Counted from 1.
DataForm (IN)	DataBuff type. Specify BYTE (char), Integer (short), Long (long), Single (float), or Double (double). No other data types can be passed. The type inside the parentheses indicates the type for the C Language.
DataBuff (OUT)	Buffer for storing data. Pass the buffer in the type specified by DataForm.

Note:

Reads the data from the data file by specifying the number of samples.
Retrieves data of Length from sample Start in the file specified by FileName and SeriesNo.
Specify DataForm from the following defined data.

BYTE:	WE_UBYTE	Integer:	WE_SWORD
Long:	WE_SLONG	Single:	WE_FLOAT
Double:	WE_DOUBLE		

Example (Visual Basic)

```
Dim FileName As String
Dim SeriesNo As Long
Dim Start As Long
Dim Length As Long
Dim ChNo As Long
Dim DataForm As Long
Dim DataBuff(1193,3) As Single
FileName = "TestData1"
ret = WeDPCsRead(FileName,2,100,1200,-1,WE_FLOAT,DataBuff(0,0))
If 0 <> ret Then
   MsgBox "File read error"
End If
```

WeDPHeaderCsWriteS

Description

Collectively writes the header information to the header file.

Syntax

WeDPHeaderCsWriteS(ByVal *FileName* As String,ByVal *SeriesNo* As Long,ByRef *ComBuff()* As CommInf1,ByRef *ChBuff()* As ChanelInf1,ByRef *AcqInfo* As AcqDataInfoEx)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Header file name.
SeriesNo (IN)	Sequence number when creating sequence files. If you specify -1, FileName becomes the file name as-is.
CommBuff() (IN)	Common information structure. Define using an array.
ChBuff() (IN)	Channel information structure. Define using an array.
AcqInfo (IN)	Data information structure.

Note:

Collectively writes the header file information for data of which the number of X-axis data points is the same. Prepare a single-element array for the common information structure buffer.
Prepare channel information structure buffer for the amount equal to the number of channels.
Pass the data obtained by WeGetAcqDataInfoEx() or WeAcqInfoInitialize() to AcqInfo.
If the file already exists, it is overwritten.

Example (Visual Basic)

```
Dim ret As Long
Dim FileName As String
Dim SeriesNo As Long
Dim CommonBuff(0) As CommonInf1
Dim ChBuff(5) As ChanelInf1
Dim AcqInfo(5) As AcqDataInfoEx
Dim InfoNum As Long
Ret = WeGetAcqDataInfoEx(hMo,-1,5,AcqInfo(),infoNum)
If 0 <> ret Then
   MsgBox "Data information read error"
End If
FileName = "TestData1"
SeriesNo = 2
ret = WeDPHeaderWriteS(FileName,SeriesNo,CommonBuff(),ChBuff(),
AcqInfo())
If 0 <> ret Then
   MsgBox "File write error"
End If
```

WeDPCsWrite

Description

Write data to a sequence file.

Syntax

WeDPCsWrite(ByVal *FileName* As String,ByVal *SeriesNo* As Long,ByVal *SampleNum* As Long,ByRef *AcqInfo* As Acq DataInfoEx,ByVal *DataForm* As Long,ByRef *DataBuff()* As Any)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Data file name. Specify the file name excluding the extension. A .wvf extension is added to the created file.
SeriesNo (IN)	Sequence number when creating sequence files. If you specify -1, FileName becomes the file name as-is.
SampleNum (IN)	Number of samples.
AcqInfo (IN)	Data information structure. Pass the data that has been returned by WeGetAcqDataInfoEx() or WeDPInitializeAcqInfo().
DataForm (IN)	DataBuff type. Specify BYTE (char), Integer (short), Long (long), Single (float), or Double (double). No other data types can be passed. The type inside the parentheses indicates the type for the C Language.
DataBuff (OUT)	Data storage buffer. Pass the buffer in the type specified by DataForm. Pass a buffer defined using a two-dimensional array.

Note:

Write data to a data file.
The files are handled as sequential files. SeriesNo is automatically added to the file names.
Specify DataForm from the following defined data.

BYTE:	WE_UBYTE	Integer:	WE_SWORD
Long:	WE_SLONG	Single:	WE_FLOAT
Double:	WE_DOUBLE		

Example (Visual Basic)

```
Dim FileName As String
Dim SeriesNo As Long
Dim AcqInfo(2) As AcqDataInfoEx
Dim DataForm As Long
Dim DataBuff(2,1000) As Single
Dim InfoNum As Integer
WeGetAcqDataInfoEx(hMo,-1,2,AcqInfo(),InfoNum)
FileName = "TestData1"
SeriesNo = 2
DataForm = WE_FLOAT
ret = WeDPCsWrite(FileName,SeriesNo,AcqInfo(),DataForm,
DataBuff(0,0))
If 0 <> ret Then
   MsgBox "File write error"
End If
```

9.9 Access the Specified Item of the Header File

WeDPHeaderItemRead

Description

Reads the information of the specified item name and specified channel from the header information of the header file.

Syntax

WeDPHeaderItemRead(ByVal *FileName* As String,ByVal *ItemName* As String,ByVal *ChNo* As Long,ByVal *BlockNo* As Long,ByRef *DataBuff* As String)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Data file name. Specify the file name excluding the extension.
ItemName (IN)	Item name.
ChNo (IN)	Channel number. Ignored if an item unrelated to the channel number is specified.
BlockNo (IN)	Block number.
DataBuff (OUT)	Buffer for storing data.

Note:

Retrieves the information of the specified item and specified channel from the header file information.
You must have an understanding of the structure of the .hdr file when using this function.
A block number is added to the item number for data containing multiple blocks. To read such item, specify the item name with the block number.
Items "GroupNumber" and "TraceNumber" cannot be retrieved.

Example (Visual Basic)

```
Dim FileName As String
Dim ItemName As String
Dim ChNo As Long
Dim DataBuff As String*256
Dim BlockNo As Long
FileName = "TestData"
ItemName = "VResolution"
ChNo = 1
BlockNo = 0
Ret = WeDPHeaderItemRead(FileName,ItemName,ChNo,BlockNo,DataBuff)
If 0 <> ret Then
   MsgBox "Error in reading a header file item"
End If
```

WeDPHeaderItemWrite

Description

Writes data to the specified item name and specified channel in the header information of the header file.

Syntax

WeDPHeaderItemWrite(ByVal *FileName* As String,ByVal *ItemName* As String,ByVal *ChNo* As Long,ByVal *BlockNo* As Long,ByRef *DataBuff* As String)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Data file name. Specify the file name excluding the extension.
ItemName (IN)	Item name.
ChNo (IN)	Channel number.
BlockNo (IN)	Block number.
DataBuff (OUT)	Buffer for storing data.

Note:

Writes data to the specified item and specified channel in the header information file. Items "FormatVersion," "Model," "Endian," "DataFormat," "GroupNumber," "TraceTotalNumber," "TraceName," "BlockNumber," and "VDataType" cannot be specified. When you set PLinearMode, set to "0" as OFF or "1" as ON.

Example (Visual Basic)

```
Dim FileName As String
Dim ItemName As String
Dim ChNo As Long
Dim BlockNo As Long
Dim DataBuff As String
FileName = "TestData"
ItemName = "VResolution"
ChNo = 1
BlockNo = 0
DataBuff = CStr(5.42)
Ret = WeDPHeaderItemWrite(FileName,ItemName,ChNo,BlockNo,DataBuff)
If 0 <> ret Then
   MsgBox "Error in reading a header file item"
End If
```

9.10 Data Operation

WeDPGetSampleChNum

Description

Gets the number of samples and number of channels of the specified file.

Syntax

WeDPGetSampleChNum(ByVal *FileName* As String,ByVal *BlockNo* As Long,ByRef *SampleNum* As Long,ByRef *ChNum* As Long)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Data file name. Specify the file name excluding the extension.
BlockNo (IN)	Block number. -1 specifies all blocks.
SampleNum (OUT)	Number of samples.
ChNum (OUT)	Number of channels.

Note:

The number of samples and number of channels of the specified file are returned.
For Scan type files, the total number of samples is returned regardless of the BlockNo setting.

Example (Visual Basic)

```
Dim FileName As String
Dim BlockNo As Long
Dim SampleNum As Long
Dim ChNum As Long
FileName = "TestData"
BlockNo = 0
ret = WeDPGetSampleChNum(FileName,SampleNum,ChNum)
If 0 <> ret Then
   MsgBox "Error in reading the number of samples and number of channels"
End If
```

WeDPGetBlockNum

Description

Gets the number of blocks of the specified file.

Syntax

WeDPGetBlockNum(ByVal *FileName* As String,ByRef *BlockNum* As Long)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN)	Data file name. Specify the file name excluding the extension.
BlockNum (OUT)	Number of blocks.

Note:

The number of blocks of the specified file is returned.

Example (Visual Basic)

```
Dim FileName As String
Dim BlockNum As Long
FileName = "TestData"
ret = WeDPGetBlockNum(FileName,BlockNum)
If 0 <> ret Then
   MsgBox "Error in reading the number of blocks"
End If
```

WeDPInitializeAcqInfo

Description

Stores the required data in the data information structure.

Syntax

WeDPInitializeAcqInfo(ByVal *VMaxData* AsDouble,ByVal *VMinData* As Double,ByVal *SampleNum* As Long,ByVal *SampInterval* As Double,ByRef *AcqInfo()* As AcqDataInfoEx)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

VMaxData (IN)	Scale Max.
VMinData (IN)	Scale Min.
SampleNum (IN)	Number of samples.
SampInterval (IN)	Sampling interval (s).
AcqInfo() (OUT)	Data information structure.

Note:

Sets and returns the required data in the data information structure.
VMaxData and VMinData are the scale values of the Y-axis when the measured data in the stored file is displayed using the Waveform Monitor of the WE7000 Control Software.
The data set in AcqDataInfo by this function is as follows:

Chanel	1 through n is set in the order of the array.
dataType	Data buffer type is set using the parameter of WeDPDataWrite or WeDPCsWrite.
blockNum	1
startBit	0
effectiveBit	0
trigActive	0
record	SampleNum.
recordLen	SampleNum.
time	0
trigPosition	0.0
Interval	Set to the SamplingInterval parameter.
VResolution	Set to 1.0.
VOffset	Set to 0.0.
TrigLevel	Set to 0.0.
TrigWidth	Set to 0.0.
PlusOverData	Set to the VMaxData parameter.
MinusOverData	Set to the VMinData parameter.
NonData	Set to the lost data.
DispMaxData	Set to the VMaxData parameter.
DispMinData	Set to the VMinData parameter.

Example (Visual Basic)

```
Dim VMaxData As Double
Dim VMinData As Double
Dim AcqInfo(3) As AcqDataInfoEx
VMaxData = 2.0
VMinData = –2.0
ret = WeDPInitializeAcqInfo(VMaxData,VMinData,1000,0.001,AcqInfo())
If 0 <> ret Then
   MsgBox "Error in setting data to data information structure"
End If
```

9.11 Read Acquisition Data Information

WeGetAcqDataInfoEx

Description

Reads the acquisition data information of the measurement module.

Syntax

WeGetAcqDataInfoEx (ByVal *hMo* As Long, ByVal *ch* As Integer, ByVal *blockNo* As Integer, ByRef *info* As AcqDataInfoEx, ByRef *infoNum* As Integer) As Integer

Return value

Returns 0 if successful. Returns an error code if unsuccessful.
An error occurs if you specify a block number that exceeds the valid number of data blocks.

Parameters

hMo (IN) Module handle.

ch (IN) Channel number.
Counted from 1. If the modules are linked, it is the link number.
-1 represents all channels.

blockNo (IN) Block number
For details, see section 4.2, "**Specifying the Data Block that You Wish to Retrieve**."

info (OUT) Data information pointer
Type AcqDataInfoEx

```
channel As Integer          ' Channel number (1 or greater)
dataType As Integer         ' Acquisition data type
                            ' WE_NULL      No parameter
                            ' WE_UBYTE     Unsigned 8-bit integer
                            ' WE_SBYTE     Signed 8-bit integer
                            ' WE_UWORD     Unsigned 16-bit integer
                            ' WE_SWORD     Signed 16-bit integer
                            ' WE_ULONG     Unsigned 32-bit integer
                            ' WE_SLONG     Signed 32-bit integer
                            ' WE_FLOAT     32-bit real number
                            ' WE_DOUBLE    64-bit real number
blockNum As Integer         ' Number of valid blocks
startBit As Integer         ' Valid bit start position for integers (bit0
                              orgreater)
effectiveBit As Integer     ' Valid bit length for integers (0 specifies
                              up to the highest bit)
trigActive As Integer       ' Trigger active (0/1)
record As Long              ' Record length (number of samples)
recordLen As Long           ' Display record length
trigPosition As Long        ' Trigger position (record position of the
                              0 base point)
time As Long                ' Date and time of collection
interval As Double          ' Sampling interval (s)
vResolution As Double       ' Scaling factor for converting physical values
vOffset As Double           ' Offset for converting physical values
trigLevel As Double         ' Trigger level (the order of the physical value)
trigWidth As Double         ' Trigger width (the order of the physical value)
```

```
        plusOverData As Double  ' Positive overrange value
                                ' (the order of the acquisition data)
        minusOverData As Double ' Negative overrange value
                                ' (the order of the acquisition data)
        nonData As Double       ' Illegal value (the order of the acquisition data)
        dispMaxData As Double   ' Maximum display value
                                ' (the order of the acquisition data)
        dispMinData As Double   ' Minimum display value
                                ' (the order of the acquisition data)
    End Type
```

infoNum (OUT) Number of AcqDataInfo that is actually read

Note:

See the note for GetAcqDataInfo for information on the AcqDataInfoEx of each module.

9.12 The List of C Language Interface

Single File Access

Function Name	Syntax
WeDPHeaderReadS	int WeDPHeaderReadS(char* lpName, UINT blockno, LPSAFEARRAY* ComInfo, LPSAFEARRAY* ChBuff)
WeDPDataRead	int WeDPDataRead(char* lpName,int chno, UINT blockno, UINT dataForm, LPVOID dataBuff)
WeDPHeaderWriteS	int WeDPHeaderWriteS(char* lpName, UINT blockno, LPSAFEARRAY* ComBuff, LPSAFEARRAY* ChBuff, LPSAFEARRAY* AcqInfo)
WeDPDataWrite	int WeDPDataWrite(char* lpName, UINT chNum, UINT blockno, UINT SampleNum, UINT dataForm, LPVOID buf, LPSAFEARRAY* AcqData), LPVOID DataBuff)

Sequential File Access

Function Name	Syntax
WeDPHeaderCsReadS	int WeDPHeaderCsReadS(char* lpName, int seriesNo, LPSAFEARRAY* ComBuff, LPSAFEARRAY* ChBuff)
WeDPCsRead	int WeDPCsRead(char* lpName, int seriesNo, UINT start, int length, int chno, UINT dataForm, LPVOID dataBuff)
WeDPHeaderCsWriteS	int WeDPHeaderCsWriteS(char* lpName, int seriesNo, LPSAFEARRAY* ComBuff, LPSAFEARRAY* ChBuff, LPSAFEARRAY* AcqInfo)
WeDPCsWrite	int WeDPCsWrite(char* lpName, int seriesNo, LPSAFEARRAY* AcqData, UINT dataForm, LPVOID dataBuff)

Access the Specified Item of the Header File

Function Name	Syntax
WeDPHeaderItemRead	int WeDPHeaderItemRead(char* lpName, char* ItemName, UINT ChNo, UINT blockno,char* data)
WeDPHeaderItemWrite	int WeDPHeaderItemWrite(char* lpName, char* ItemName, UINT ChNo, UINT blockno,char* data)

Data Operation

Function Name	Syntax
WeDPGetSampleChNum	int WeDPGetSampleChNum(char* lpName, UINT* blockno, UINT* SampleNum, UINT* ChNum)
WeDPGetBlockNum	int WeDPGetBlockNum(char* lpName, UINT* blockNo)
WeDPInitializeAcqInfo	int WeDPInitializeAcqInfo(double VMaxData, double VMinData, double SamplingInterval,LPSAFEARRAY* AcqInfo)

Read Acquisition Data Information

Function Name	Syntax
WeGetAcqDataInfoEx	USHORT WeGetAcqDataInfoEx(HANDLE hMo, short ch, int blockNo,AcqDataInfoEx* inf, USHORT chNum)

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

V

W