# Agilent U2701A/U2702A USB Modular Oscilloscopes

## IVI-COM Programmer's Reference for Visual Basic .Net

Agilent Technologies

# Notices



## Manual Part Number

U2702-90035

## Edition

Third Edition, August 13, 2012

Agilent Technologies, Inc.
5301 Stevens Creek Blvd.
Santa Clara, CA 95052 USA

## Warranty

**The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.**

## Technology Licenses

The hardware and or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

## Restricted Rights Legend

U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

## Safety Notices

**CAUTION**

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

**WARNING**

**A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.**

# Contents

# 1 Introduction to IVI-COM



This chapter introduces the remote programming basics of the U2701A/U2702A USB modular oscilloscopes. The IVI-COM programming commands provide the means to control this instrument remotely via a PC.

# Getting Started

The IVI Foundation is an open consortium founded in year 1998 to promote specifications for programming test instruments.

For complete information on the IVI Foundation and for the most up-to-date versions of all IVI specifications and components, you can visit the IVI Foundation web site at www.ivifoundation.org.

# Installation Guide

1. To download the AgilentU2701A IVI Driver, go to the Agilent Technical Support web site at: http://www.home.agilent.com/agilent/techSupport.jspx?pid=1369621&cc=US&lc=eng&t=80029.k.0&guid=181735
2. Click **Drivers & Software > IVI-COM and IVI-C for U2701A/U2702A** to download the AgilentU2701A IVI Driver package.
3. Save the file to any location on your hard disk.
4. Disconnect any instrument that is connected to your PC and close all other applications on your PC.
5. Double-click the saved installation file to begin installation.
6. The latest version of the IVI Shared Components should be downloaded from www.ivifoundation.org prior to installing the AgilentU2701A IVI Driver package. It is recommended that you periodically check for newer versions of the IVI Shared Components and update as available from the IVI Foundation.
7. If the latest version of the IVI Shared Components is detected, installation of the AgilentU2701A IVI Driver will proceed as normal.

8 The **AgilentU2701A IVI Driver 1.0.0.0 Setup Wizard** dialog will appear. Click **Next** to begin.

9 Read the License Agreement and select **I accept the terms in the License Agreement** to proceed. You may click **Print** to print a hardcopy of the Agilent License Terms for your reference. Click **Next** to proceed.

10 Fill in the Customer Information Form accordingly, and click **Next**.

11 Select the **Typical** or **Full** option in the Setup Type to install the AgilentU2701A IVI Driver package. Advanced users may select the **Custom** option to customize the program features to be installed and where they will be installed. Click **Next** to proceed.

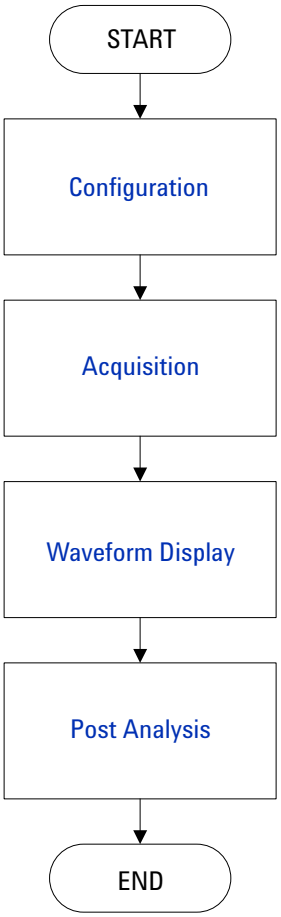
12 Click **Next** to install to the specified folder or click **Change** to install to a different folder.

13 Click **Install** to begin the installation of the AgilentU2701A IVI Driver package.

14 Click **Finish** when the installation has completed.

# General Guidelines for Oscilloscope Programming

The block diagram below illustrates the general flow of steps required to program an oscilloscope. The necessary IVI-COM driver commands are arranged to reflect this block diagram.




You can use the commands in this section to configure the oscilloscope channels for waveform acquisition, set the type of event that triggers the oscilloscope, modify the registry status, and perform various utility operations.

You can use the commands in this section to configure the acquisition type, the size of the waveform record, the length of time that corresponds to overall waveform record, and the position of the first point in the waveform record relative to the trigger event.

The statements in this section is used to acquire waveform data from the oscilloscope. You can use the commands in this section to initiate, acquire, and abort measurements.

The statements in this section is used to analyze the waveform data acquired from the oscilloscope. You can also use the commands in this section to initiate, acquire, and abort measurements.

# Side-by-Side SCPI and IVI-COM Comparison of the U2701A/U2702A Commands

**Table 1-1** SCPI, IAgilentU2701A IVI, and IIviScope command comparison

| SCPI command | IAgilentU2701A IVI command | IIviScope command |
|---|---|---|
| **IEEE 488.2 Common commands** | | |
| *CLS | IAgilentU2701AStatus.Clear | - |
| *IDN | IIviDriver.Identity | IIviDriver.Identity |
| *OPC? | IAgilentU2701ASystem.WaitForOperation Complete() | - |
| *RST | IIviDriverUtility.Reset() | IIviDriverUtility.Reset() |
| **Root level commands** | | |
| AUTO | IAgilentU2701AMeasurements.AutoSetup | IIviScopeMeasurements.AutoSetup |
| **ACQuire commands** | | |
| AVERages | IAgilentU2701AAcquisition.NumberOf Averages | IIviScopeAcquisition.NumberOfAverages |
| MODE | IAgilentU2701AAcquisition.SampleMode | IIviScopeAcquisition.SampleMode |
| SRATe? | IAgilentU2701AAcquisition.SampleRate | IIviScopeAcquisition.SampleRate |
| TYPE | IAgilentU2701AAcquisition.Type | IIviScopeAcquisition.Type |
| **CHANnel<n> commands** | | |
| BWLimit | IAgilentU2701AChannel.BandwidthLimit | - |
| COUPling | IAgilentU2701AChannel.Coupling | IIviScopeChannel.Coupling |
| DISPlay | IAgilentU2701AChannel.Enabled | IIviScopeChannel.Enabled |
| INVert | - | - |
| OFFSet | IAgilentU2701AChannel.Offset | IIviScopeChannel.Offset |
| PROBe | IAgilentU2701AChannel.ProbeAttenuation | IIviScopeChannel.ProbeAttenuation |
| SCALe | IAgilentU2701AChannel.Range | IIviScopeChannel.Range |

**Table 1-1** SCPI, IAgilentU2701A IVI, and IIviScope command comparison

| SCPI command | IAgilentU2701A IVI command | IIviScope command |
|---|---|---|
| **KEY commands** | | |
| AUTO_SCALE | IAgilentU2701AMeasurements.AutoSetup | IIviScopeMeasurements.AutoSetup |
| CH1 | IAgilentU2701AChannel.Enabled | IIviScopeChannel.Enabled |
| CH1_POS_DEC | IAgilentU2701AChannel.Offset | IIviScopeChannel.Offset |
| CH1_POS_INC | IAgilentU2701AChannel.Offset | IIviScopeChannel.Offset |
| CH1_SCALE_DEC | IAgilentU2701AChannel.Range | IIviScopeChannel.Range |
| CH1_SCALE_INC | IAgilentU2701AChannel.Range | IIviScopeChannel.Range |
| CH2 | IAgilentU2701AChannel.Enabled | IIviScopeChannel.Enabled |
| CH2_POS_DEC | IAgilentU2701AChannel.Offset | IIviScopeChannel.Offset |
| CH2_POS_INC | IAgilentU2701AChannel.Offset | IIviScopeChannel.Offset |
| CH2_SCALE_DEC | IAgilentU2701AChannel.Range | IIviScopeChannel.Range |
| CH2_SCALE_INC | IAgilentU2701AChannel.Range | IIviScopeChannel.Range |
| MAIN_DELAYED | IAgilentU2701AAcquisition.StartTime | IIviScopeAcquisition.StartTime |
| MATH | IAgilentU2701AMeasurementsMathFunction | - |
| MEASURE | IAgilentU2701AMeasurement | IIviScope.Measurement |
| MODE_COUPLING | IAgilentU2701AChannel.Coupling | IIviScopeChannel.Coupling |
| SINGLE | IAgilentU2701AMeasurements.Initiate | IIviScopeMeasurements.Initiate |
| TIME_POS_DEC | IAgilentU2701AAcquisition.StartTime | IIviScopeAcquisition.StartTime |
| TIME_POS_INC | IAgilentU2701AAcquisition.StartTime | IIviScopeAcquisition.StartTime |
| TIME_SCALE_DEC | IAgilentU2701AAcquisition.TimePerRecord | IIviScopeAcquisition.TimePerRecord |
| TIME_SCALE_INC | IAgilentU2701AAcquisition.TimePerRecord | IIviScopeAcquisition.TimePerRecord |
| TRIG_LVL_DEC | IAgilentU2701ATrigger.Level Property | IIviScopeTrigger.Level |
| TRIG_LVL_INC | IAgilentU2701ATrigger.Level Property | IIviScopeTrigger.Level |

**Table 1-1** SCPI, IAgilentU2701A IVI, and IIviScope command comparison

| SCPI command | IAgilentU2701A IVI command | IIviScope command |
|---|---|---|
| **MEASure commands** | | |
| FALLtime | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| FREQuency | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| NDUTycycle | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| NWIDth | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| OVERshoot | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| PDUTycycle | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| PERiod | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| PREShoot | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| PWIDth | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| RISetime | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| VAMPlitude | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| VAVerage | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| VBASe | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| VMAX | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| VMIN | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |

**Table 1-1** SCPI, IAgilentU2701A IVI, and IIviScope command comparison

| SCPI command | IAgilentU2701A IVI command | IIviScope command |
|---|---|---|
| **MEASure commands** | | |
| VPP | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| VRMS | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| VTOP | AgilentU2701AMeasurementEnum Enumeration | IviScopeMeasurementEnum Enumeration |
| **TIMebase commands** | | |
| DELayed | IAgilentU2701AAcquisition.StartTime | IIviScopeAcquisition.StartTime |
| HOLDoff | IAgilentU2701ATrigger.Holdoff | IIviScopeTrigger.Holdoff |
| POSition | IAgilentU2701AAcquisition.StartTime | IIviScopeAcquisition.StartTime |
| SCALe | IAgilentU2701AAcquisition.TimePerRecord | IIviScopeAcquisition.TimePerRecord |
| **TRIGger commands** | | |
| [EDGE]:COUPling | IAgilentU2701ATrigger.Coupling Property | IIviScopeChannel.Coupling |
| [EDGE]:LEVel | IAgilentU2701ATrigger.Level Property | IIviScopeTrigger.Level |
| [EDGE]:SLOPe | IAgilentU2701ATriggerEdge.Slope Property | IIviScopeTriggerEdge.Slope |
| [EDGE]:SOURe | IAgilentU2701ATrigger.Source Property | IIviScopeTrigger.Source |
| [EDGE]:SWEep | IAgilentU2701ATrigger.Modifier Property | IIviScopeTrigger.Modifier |
| MODE | IAgilentU2701ATrigger.Type Property | IIviScopeTrigger.Type |
| PULSe:MODE | AgilentU2701ATriggerTypeEnum Enumeration | IIviScopeTriggerTypeEnum Enumeration |
| PULSEe:WIDTh | AgilentU2701ATriggerTypeEnum Enumeration | IIviScopeTriggerTypeEnum Enumeration |
| STATus | IAgilentU2701ATrigger.Status | - |
| **WAVeform commands** | | |
| DATA? | IAgilentU2701AMeasurement.ReadWaveform | IIviScopeMeasurement.ReadWaveform |
| XINCrement? | IAgilentU2701AMeasurement.ReadWaveform | IIviScopeMeasurement.ReadWaveform |
| XORigin? | IAgilentU2701AMeasurement.ReadWaveform | IIviScopeMeasurement.ReadWaveform |

# 1 Introduction to IVI-COM

# 2 Configuration



This chapter describes the configuration commands used to program the U2701A/U2702A USB modular oscilloscopes over the remote interface. You can use the commands in this chapter to configure the oscilloscope channels for waveform acquisition, set the type of event that triggers the oscilloscope, modify the registry status, and perform various utility operations.

# Count

### Type

Property

### Function

Get

### Description

This command returns the number of channels available.

### Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Count
```

### Parameters

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Count Statement
    Dim Count As Int32
    Count = Driver.Channels.Count
End Sub
```

# Item

This is an interface reference pointer to the IAgilentU2701AChannel interface which is selected by the channel name.

## BandwidthLimit

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the bandwidth limit status for the selected channel. If `True`, the bandwidth limit for the selected channel is enabled. If `False`, the bandwidth limit for the selected channel is disabled.

### Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Item(Name)
          └ BandwidthLimit
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' BandwidthLimit Statement
    ' To Get Bandwidth Limit from the Instrument
    Dim BandwidthLimit As Boolean
    BandwidthLimit =
     Driver.Channels.Item("Channel1").BandwidthLimit
    ' To Set Bandwidth Limit eg. True
    Driver.Channels.Item("Channel1").BandwidthLimit = True
End Sub
```

# Configure

### Type

Method

### Function

Set

### Description

This command configures the most commonly used properties of the oscilloscope channel sub-system. Use this command to enable or disable the channel and to set the range, offset, coupling, and probe attenuation values.

### Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Item(Name)
          └ Configure(Range, Offset, Coupling, ProbeAttenuation,
            Enabled)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |
| Range | Double | Specifies the vertical range. This value sets the Vertical Range property. |
| Offset | Double | Specifies the vertical offset. This value sets the Vertical Offset property. |
| Coupling | AgilentU2701A VerticalCoupling Enum | Specifies how to couple the input signal. This value sets the Vertical Coupling property. |
| ProbeAttenuation | Double | Specifies the probe attenuation. This value sets the ProbeAttenuation property. |
| Enabled | Boolean | Specifies if the channel is enabled for acquisition. This value sets the Channels.Enabled property. |

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Configure Statement
    Dim BandwidthLimit As Boolean
    Driver.Channels.Item("Channel1").Configure(40.0, 0.0,
     AgilentU2701AVerticalCouplingEnum.AgilentU2701AVerticalCoupl
     ingDC, 1.0, True)
End Sub
```

# Coupling

## Type

Property

## Function

Get and Set

## Description

This command returns/sets how the oscilloscope couples the input signal.

## Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Item(Name)
          └ Coupling
```

## Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

## Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A VerticalCoupling Enum | Enum | See "AgilentU2701AVerticalCouplingEnum" on page 96. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Coupling Statement
    ' To Get Coupling from the Instrument
    Dim Coupling As New AgilentU2701AVerticalCouplingEnum
    Coupling = Driver.Channels.Item("Channel1").Coupling
    ' To Set Coupling eg. DC
    Driver.Channels.Item("Channel1").Coupling =
     AgilentU2701AVerticalCouplingEnum.AgilentU2701AVerticalCoupl
     ingDC
End Sub
```

# Enabled

### Type

Property

### Function

Get and Set

### Description

If this command is set to True, the oscilloscope acquires a waveform for this channel when the IAgilentU2701AMeasurement.Initiate, IAgilentU2701AMeasurement.ReadWaveform, IAgilentU2701AMeasurement.ReadWaveformMeasurement, or IIviScopeMeasurement.ReadWaveformMinMax methods are called.

### Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Item(Name)
          └ Enabled
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Enabled Statement
    ' To Get Enabled from the Instrument
    Dim Enabled As Boolean

    Enabled = Driver.Channels.Item("Channel1").Enabled
    ' To Set Enabled eg. True
    Driver.Channels.Item("Channel1").Enabled = True

End Sub
```

# Offset

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the location of the center of the range that was specified with the Range property. The units are expressed in volts, with respect to ground. For example, to acquire a sine wave spanning from 0.0 V to 10.0 V, set Offset to 5.0 V.

### Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Item(Name)
          └ Offset
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Offset Statement
    ' To Get Offset from the Instrument
    Dim Offset As Double
    Offset = Driver.Channels.Item("Channel1").Offset
    ' To Set Offset eg. 0.0
    Driver.Channels.Item("Channel1").Offset = 0.0
End Sub
```

# ProbeAttenuation

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the scaling factor by which the probe attenuates the input signal. For example, with a 10:1 probe, the value is 10.0.

### Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Item(Name)
          └ ProbeAttenuation
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ProbeAttenuation Statement
    ' To Get ProbeAttenuation from the Instrument
    Dim ProbeAttenuation As Double
    ProbeAttenuation =
     Driver.Channels.Item("Channel1").ProbeAttenuation
    ' To Set ProbeAttenuation eg. 1.0
    Driver.Channels.Item("Channel1").ProbeAttenuation = 1.0
End Sub
```

# Range

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the absolute value of the input range that the oscilloscope can acquire for the channel. The units are expressed in volts. For example, to acquire a sine wave spanning from –5.0 V to 5.0 V, set Range to 10.0 V.

### Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Item(Name)
          └ Range
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Remarks

The absolute value of the input range that the oscilloscope can acquire is equivalent to the Volt/Div value multiplied by eight at the scope setting.

| Volt/Div (Attenuation = 1x) | Voltage range |
|---|---|
| 2 mV | 16 mV |
| 5 mV | 40 mV |
| 10 mV | 80 mV |
| 20 mV | 160 mV |
| 50 mV | 400 mV |
| 100 mV | 800 mV |
| 200 mV | 1.6 V |
| 500 mV | 4 V |
| 1 V | 8 V |
| 2 V | 16 V |
| 5 V | 40 V |

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Range Statement
    ' To Get Range from the Instrument
    Dim Range As Double
    Range = Driver.Channels.Item("Channel1").Range
    ' To Set Range eg. 40
    Driver.Channels.Item("Channel1").Range = 40

End Sub
```

# Name

### Type

Property

### Function

Get

### Description

This command returns the channel name for a given index.

### Hierarchy

```
IAgilentU2701A
    └ Channels
       └ Name(Index)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Index | Long/Int32 | One based index into the collection of channels. |

### Return Format

String

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Name Statement
    Dim Name As String
    Name = Driver.Channels.Name(1)
End Sub
```

# Close

### Type

Method

### Function

Set

### Description

This command closes the I/O session to the instrument. Driver methods and properties that access the instrument are not accessible after Close is called.

### Hierarchy

```
IAgilentU2701A
    └ Close()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    'Close Statement
    Driver.Close()
End Sub
```

# Initialize

### Type

Method

### Function

Set

### Description

This command opens the I/O session to the instrument. Driver methods and properties that access the instrument are only accessible after Initialize is called. Initialize optionally performs a Reset and queries the instrument to validate the instrument model.

### Hierarchy

```
IAgilentU2701A
    └ Initialize(ResourceName, IdQuery, Reset, OptionString)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| ResourceName | String/BSTR | An IVI logical name or an instrument specific string that identifies the address of the instrument, such as a VISA resource descriptor string. |
| IdQuery | Boolean | Specifies whether to verify the ID of the instrument. |
| Reset | Boolean | Specifies whether to reset the instrument. |
| OptionString | String/BSTR | The user can use the *OptionString* parameter to specify the initial values of certain IVI inherent attributes for the session.<br>The format of an assignment in the *OptionString* parameter is "Name=Value", where *Name* is one of: RangeCheck, QueryInstrumentStatus, Cache, Simulate, RecordCoercions, InterchangeCheck, or DriverSetup.<br>Value is either True or False except for DriverSetup. If the *OptionString* parameter contains an assignment for the Driver Setup attribute, the Initialize function assumes that everything following "DriverSetup=" is part of the assignment. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Initialize Statement
    Driver.Initialize("USB0::2391::10520::MY48151002::0::INSTR",
     True, True, "")
End Sub
```

# Initialized

### Type

Property

### Function

Get

### Description

This command returns `True` between a successful call to the Initialize method and a successful call to the Close method, and `False` at all other times.

### Hierarchy

```
IAgilentU2701A
    └ Initialized
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Initialized Statement
    Dim Initialized As Boolean
    Initialized = Driver.Initialized
End Sub
```

# Configure

### Type

Method

### Function

Set

### Description

This command configures the trigger *Type* and *Holdoff*. *Holdoff* units are expressed in seconds.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Configure(Type, Holdoff)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Type | AgilentU2701A TriggerTypeEnum | Specifies the trigger type. This value sets the Trigger.Type property. See "AgilentU2701ATriggerTypeEnum" on page 93 for more information. |
| Holdoff | Double | Specifies the trigger hold-off. This value sets the Trigger. Holdoff property. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Configure Statement
    Driver.Trigger.Configure(AgilentU2701ATriggerTypeEnum.Agilent
     U2701ATriggerEdge, 5)
End Sub
```

# Coupling

### Type

Property

### Function

Get and Set

### Description

This command returns/sets how the oscilloscope couples the trigger source.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Coupling
```

### Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A TriggerCoupling Enum | Enum | See "AgilentU2701ATriggerCouplingEnum" on page 91 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Coupling Statement
    ' To Get Coupling from the Instrument
    Dim Coupling As AgilentU2701ATriggerCouplingEnum
    Coupling = Driver.Trigger.Coupling
    ' To Set Coupling eg. AC
    Driver.Trigger.Coupling =
     AgilentU2701ATriggerCouplingEnum.AgilentU2701ATriggerCouplin
     gAC
End Sub
```

# Edge

This interface configures slope of edge trigger.

## Configure

### Type

Method

### Function

Set

### Description

This command configures the conditions for edge trigger. An edge trigger occurs when the trigger source signal passes through the trigger level with the specified slope.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Edge
          └ Configure(Source, Level, Slope)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Source | String | Specifies the trigger source. This value sets the Trigger.Source property. |
| Level | Double | Specifies the trigger level. This value sets the Trigger.Level property. |
| Slope | AgilentU2701A TriggerSlopeEnum | Specifies the trigger slope. This value sets the Trigger.Slope property. See "AgilentU2701ATriggerSlopeEnum" on page 92 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Configure Statement
    Driver.Trigger.Edge.Configure("Channel1", 0,
     AgilentU2701ATriggerSlopeEnum.AgilentU2701ATriggerSlopePosit
     ive)
End Sub
```

# Slope

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the slope, a rising or a falling edge, that triggers the oscilloscope.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Edge
          └ Slope
```

### Parameters

| Item | Type | Description |
|---|---|---|
| AgilentU2701A TriggerSlopeEnum | Enum | See "AgilentU2701ATriggerSlopeEnum" on page 92 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Slope Statement
    ' To Get Slope from the Instrument
    Dim Slope As AgilentU2701ATriggerSlopeEnum
    Slope = Driver.Trigger.Edge.Slope
    ' To Set Slope eg. AC
    Driver.Trigger.Edge.Slope =
     AgilentU2701ATriggerSlopeEnum.AgilentU2701ATriggerSlopePosit
     ive
End Sub
```

# Glitch

This interface configures the condition, polarity, and width of the glitch trigger.

## Condition

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the glitch condition that determines whether the oscilloscope triggers on a pulse with a width less than or greater than the glitch width value.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Glitch
          └ Condition
```

### Parameters

| Item | Type | Description |
|---|---|---|
| AgilentU2701A GlitchCondition Enum | Enum | See "AgilentU2701AGlitchConditionEnum" on page 86 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Condition Statement
    ' To Get Condition from the Instrument
    Dim Condition As AgilentU2701AGlitchConditionEnum
    Condition = Driver.Trigger.Glitch.Condition
    ' To Set Condition eg. Greater Than
    Driver.Trigger.Glitch.Condition =
     AgilentU2701AGlitchConditionEnum.AgilentU2701AGlitchConditio
     nGreaterThan
End Sub
```

# Configure

### Type

Method

### Function

Set

### Description

This command configures the glitch trigger *Source*, *Level*, *Width*, *Polarity*, and *Condition*. A glitch trigger occurs when the edge of a pulse that matches the *Width* and *Polarity* crosses the specified *Level* (expressed in volts).

### Hierarchy

```
IAgilentU2701A
    └ Trigger
      └ Glitch
        └ Configure(Source, Level, Width, Polarity, Condition)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Source | String | Specifies the trigger source. This value sets the Trigger.Source property. |
| Level | Double | Specifies the trigger level. This value sets the Trigger.Level property. |
| Width | Double | Specifies the glitch triggering glitch width in seconds. This value sets the Glitch.Width property. |
| Polarity | AgilentU2701A GlitchPolarity Enum | Specifies the glitch polarity. This value sets the Glitch. Polarity property. |
| Condition | AgilentU2701A GlitchCondition Enum | Specifies the glitch condition. This value sets the Glitch.Condition property. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Configure Statement
    Driver.Trigger.Glitch.Configure("Channel1", 0, 0.00000003,
     AgilentU2701AGlitchPolarityEnum.AgilentU2701AGlitchPolarityP
     ositive,
     AgilentU2701AGlitchConditionEnum.AgilentU2701AGlitchConditio
     nGreaterThan)
End Sub
```

# Polarity

## Type

Property

## Function

Get and Set

## Description

This command returns/sets the polarity of the glitch that triggers the oscilloscope.

## Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Glitch
          └ Polarity
```

## Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A GlitchPolarity Enum | Enum | Specifies the glitch polarity. This value sets the Glitch.Polarity property. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Polarity Statement
    ' To Get Polarity from the Instrument
    Dim Polarity As AgilentU2701AGlitchPolarityEnum
    Polarity = Driver.Trigger.Glitch.Polarity
    ' To Set Polarity eg. Positive
    Driver.Trigger.Glitch.Polarity =
     AgilentU2701AGlitchPolarityEnum.AgilentU2701AGlitchPolarityP
     ositive
End Sub
```

# Width

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the glitch width. The units are expressed in seconds.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Glitch
          └ Width
```

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Width Statement
    ' To Get Width from the Instrument
    Dim Width As Double
    Width = Driver.Trigger.Glitch.Width
    ' To Set Width eg. 0.00000003
    Driver.Trigger.Glitch.Width = 0.00000003
End Sub
```

# Holdoff

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the length of time the oscilloscope waits after it fills the acquisition buffer until the oscilloscope enables the trigger interface to detect another trigger. The units are expressed in seconds.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Holdoff
```

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Holdoff Statement
    ' To Get Holdoff from the Instrument
    Dim Holdoff As Double
    Holdoff = Driver.Trigger.Holdoff
    ' To Set Holdoff eg. 0.00000006
    Driver.Trigger.Holdoff = 0.00000006
End Sub
```

# Modifier

### Type

Property

### Function

Get and Set

### Description

This command determines the oscilloscope behavior in the absence of a trigger.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Modifier
```

### Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A TriggerModifier Enum | Enum | See "AgilentU2701ATriggerModifierEnum" on page 91 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Modifier Statement
    ' To Get Modifier from the Instrument
    Dim Modifier As AgilentU2701ATriggerModifierEnum
    Modifier = Driver.Trigger.Modifier
    ' To Set Modifier eg. Auto
    Driver.Trigger.Modifier =
     AgilentU2701ATriggerModifierEnum.AgilentU2701ATriggerModifie
     rAuto
End Sub
```

# Level

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the voltage threshold for the trigger interface. The units are expressed in volts.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
      └ Level
```

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Level Statement
    ' To Get Level from the Instrument
    Dim Level As Double
    Level = Driver.Trigger.Level
    ' To Set Level eg. 0
    Driver.Trigger.Level = 0
End Sub
```

# Source

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the signal that the oscilloscope monitors for a trigger. It can be a channel or one of many other values.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Source
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Source Statement
    ' To Get Source from the Instrument
    Dim Source As String
    Source = Driver.Trigger.Source
    ' To Set Source eg. Channel1
    Driver.Trigger.Source = "Channel1"
End Sub
```

# Status

### Type

Property

### Function

Get

### Description

If this command is set to `True`, the waveform starts directly after the trigger condition is met. If this command is set to `False`, the instrument has to send a force trigger to get a waveform.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Status
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Status Statement
    Dim Status As Boolean
    Status = Driver.Trigger.Status
End Sub
```

# TV

This interface configures the signal format, number of lines, and events of the TV trigger.

## Configure

### Type

Method

### Function

Set

### Description

This command configures the TV trigger source, signal format, event, and polarity.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
      └ TV
        └ Configure(Source, SignalFormat, Event)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Source | String | Specifies the trigger source. This value sets the Trigger.Source property. |
| SignalFormat | AgilentU2701ATV SignalFormat Enum | Specifies the TV trigger signal format. This value sets the TV.SignalFormat property. See "AgilentU2701ATVSignalFormatEnum" on page 94 for more information. |
| Event | AgilentU2701ATV TriggerEventEnum | Specifies the TV trigger event. This value sets the TV.Event property. See "AgilentU2701ATVTriggerEventEnum" on page 95 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Configure Statement
    Driver.Trigger.TV.Configure("Channel1",
     AgilentU2701ATVSignalFormatEnum.AgilentU2701ATVSignalFormatN
     TSC,
     AgilentU2701ATVTriggerEventEnum.AgilentU2701ATVTriggerEventA
     nyField)
End Sub
```

# Event

## Type

Property

## Function

Get and Set

## Description

This command returns/sets the event on which the oscilloscope triggers.

## Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ TV
          └ Event
```

## Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701ATV TriggerEventEnum | Enum | See "AgilentU2701ATVTriggerEventEnum" on page 95 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Event Statement
    ' To Get Event from the Instrument
    Dim TVEvent As AgilentU2701ATVTriggerEventEnum
    TVEvent = Driver.Trigger.TV.Event
    ' To Set Event eg. Channel1
    Driver.Trigger.TV.Event =
     AgilentU2701ATVTriggerEventEnum.AgilentU2701ATVTriggerEventA
     nyField
End Sub
```

# LineNumber

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the line on which the oscilloscope triggers. The line number is absolute and not relative to the field of the TV signal.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ TV
          └ LineNumber
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' LineNumber Statement
    ' To Get LineNumber from the Instrument
    Dim LineNumber As Int32
    LineNumber = Driver.Trigger.TV.LineNumber
    ' To Set LineNumber eg. 1
    Driver.Trigger.TV.LineNumber = 1
End Sub
```

# SignalFormat

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the format of the TV signal on which the oscilloscope triggers.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ TV
          └ SignalFormat
```

### Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701ATV SignalFormat Enum | Enum | See "AgilentU2701ATVSignalFormatEnum" on page 94 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SignalFormat Statement
    ' To Get SignalFormat from the Instrument
    Dim SignalFormat As AgilentU2701ATVSignalFormatEnum
    SignalFormat = Driver.Trigger.TV.SignalFormat
    ' To Set SignalFormat eg. NTSC
    Driver.Trigger.TV.SignalFormat =
     AgilentU2701ATVSignalFormatEnum.AgilentU2701ATVSignalFormatN
     TSC
End Sub
```

# Type

## Type

Property

## Function

Get and Set

## Description

This command returns/sets the kind of event that triggers the oscilloscope.

## Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ TV
          └ Type
```

## Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A TriggerTypeEnum | Enum | See "AgilentU2701ATriggerTypeEnum" on page 93 for more information. |

## Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Type Statement
    ' To Get Type from the Instrument
    Dim Type As AgilentU2701ATriggerTypeEnum
    Type = Driver.Trigger.Type
    ' To Set Type eg. Edge
    Driver.Trigger.Type =
     AgilentU2701ATriggerTypeEnum.AgilentU2701ATriggerEdge
End Sub
```

# Width

This interface configures condition, polarity, and threshold level of width trigger.

## Condition

### Type

Property

### Function

Get and Set

### Description

This command returns/sets whether a pulse within or outside the high and low thresholds triggers the oscilloscope.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Width
          └ Condition
```

### Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A WidthCondition Enum | Enum | See "AgilentU2701AWidthConditionEnum" on page 97 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Condition Statement
    ' To Get Condition from the Instrument
    Dim Condition As AgilentU2701AWidthConditionEnum
    Condition = Driver.Trigger.Width.condition
    ' To Set Condition eg. Outside
    Driver.Trigger.Width.condition =
     AgilentU2701AWidthConditionEnum.AgilentU2701AWidthConditionO
     utside
End Sub
```

# Configure

## Type

Method

## Function

Set

## Description

This command configures the width trigger *Source*, *Level*, *ThresholdLow*, *ThresholdHigh*, *Polarity*, and *Condition*. A width trigger occurs when a pulse, which passes through *Level*, with a width between or outside, the width threshold is detected.

## Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Width
          └ Configure(Source, Level, ThresholdLow, ThresholdHigh,
            Polarity, Condition)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Source | String | Specifies the trigger source. This value sets the Trigger.Source property. |
| Level | Double | Specifies the trigger level. This value sets the Trigger.Level property. |
| ThresholdLow | Double | Sets the width triggering low threshold in seconds. This value sets the Width.ThresholdLow property. |
| ThresholdHigh | Double | Sets the width triggering high threshold in seconds. This value sets the Width.ThresholdHigh property. |
| Polarity | AgilentU2701A WidthPolarity Enum | Specifies the width polarity. This value sets the Width.Polarity property. See "AgilentU2701AWidthPolarityEnum" on page 97 for more information. |
| Condition | AgilentU2701A WidthCondition Enum | Specifies the width condition. This value sets the Width.Condition property. See "AgilentU2701AWidthConditionEnum" on page 97 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Configure Statement
    Driver.Trigger.Width.Configure("Channel1", 0, 0, 0.000000016,
     AgilentU2701AWidthPolarityEnum.AgilentU2701AWidthPolarityPos
     itive,
     AgilentU2701AWidthConditionEnum.AgilentU2701AWidthConditionO
     utside)
End Sub
```

# Polarity

## Type

Property

## Function

Get and Set

## Description

This command returns/sets the polarity of the pulse that triggers the oscilloscope.

## Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Width
          └ Polarity
```

## Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A WidthPolarity Enum | Enum | See "AgilentU2701AWidthPolarityEnum" on page 97 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Polarity Statement
    ' To Get Polarity from the Instrument
    Dim Polarity As AgilentU2701AWidthPolarityEnum
    Polarity = Driver.Trigger.Width.polarity
    ' To Set Polarity eg. Positive
    Driver.Trigger.Width.polarity =
     AgilentU2701AWidthPolarityEnum.AgilentU2701AWidthPolarityPos
     itive
End Sub
```

# ThresholdHigh

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the high width threshold time, expressed in seconds.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Width
          └ ThresholdHigh
```

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ThresholdHigh Statement
    ' To Get ThresholdHigh from the Instrument
    Dim ThresholdHigh As Double
    ThresholdHigh = Driver.Trigger.Width.ThresholdHigh
    ' To Set ThresholdHigh eg. 0.000000016
    Driver.Trigger.Width.ThresholdHigh = 0.000000016
End Sub
```

# ThresholdLow

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the low width threshold time, expressed in seconds.

### Hierarchy

```
IAgilentU2701A
    └ Trigger
       └ Width
          └ ThresholdLow
```

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ThresholdLow Statement
    ' To Get ThresholdLow from the Instrument
    Dim ThresholdLow As Double
    ThresholdLow = Driver.Trigger.Width.ThresholdLow
    ' To Set ThresholdLow eg. 0
    Driver.Trigger.Width.ThresholdLow = 0
End Sub
```

# Clear

### Type

Method

### Function

Set

### Description

This command clears all event registers and error queues. The enable registers are unaffected.

### Hierarchy

```
IAgilentU2701A
    └ Status
       └ Clear()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Clear Statement
    Driver.Status.Clear()
End Sub
```

# ConfigureServiceRequest

### Type

Method

### Function

Set

### Description

This command clears all the enable registers. It then sets the appropriate transition filters and enable registers so when the specified event(s) occur(s) the instrument requests service. All other events are disabled from generating a service request. To detect a service request the client application must poll the status byte using the SerialPoll method or Register property and test the request service bit.

### Hierarchy

```
IAgilentU2701A
    └ Status
      └ ConfigureServiceRequest(Reason)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Reason | AgilentU2701A SRQReasonEnum | The defined values are expressed in the powers of two: 1, 2, 4, and so on. You can OR several reasons together so multiple events can generate a service request. See "AgilentU2701ASRQReasonEnum" on page 87 for more information. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ConfigureServiceRequest Statement
    Driver.Status.ConfigureServiceRequest(AgilentU2701ASRQReasonE
     num.AgilentU2701ASRQReasonEsrCommandError)
End Sub
```

# Preset

### Type

Method

### Function

Set

### Description

This command sets the SCPI defined enable registers and transition filters.

### Hierarchy

```
IAgilentU2701A
    └ Status
       └ Preset()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Preset Statement
    Driver.Status.Preset()
End Sub
```

# Register

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the instrument status registers.

### Hierarchy

```
IAgilentU2701A
    └ Status
       └ Register(Register, SubRegister, val)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Register | AgilentU2701A StatusRegister Enum | The status register to access. See "AgilentU2701AStatusRegisterEnum" on page 89 for available registers. |
| SubRegister | AgilentU2701A StatusSubRegister Enum | The status sub register to access. See "AgilentU2701AStatusSubRegisterEnum" on page 90 for available sub registers. |
| val | Long/Int32 | Instrument Status Register value |

### Return Format

| Item | Type | Description |
|---|---|---|
| val | Long/Int32 | Instrument Status Register value |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Register Statement
    ' To Get Register from Instrument
    Dim val As Int32
    val =
     Driver.Status.Register(AgilentU2701AStatusRegisterEnum.Agile
     ntU2701AStatusRegisterArmEvent,
     AgilentU2701AStatusSubRegisterEnum.AgilentU2701AStatusSubReg
     isterCondition)
    ' To Set Register eg. 0
    Driver.Status.Register(AgilentU2701AStatusRegisterEnum.Agilen
     tU2701AStatusRegisterArmEvent,
     AgilentU2701AStatusSubRegisterEnum.AgilentU2701AStatusSubReg
     isterCondition) = 0
End Sub
```

# SerialPoll

### Type

Property

### Function

Get

### Description

This command returns the serial poll of the instrument status byte.

### Hierarchy

```
IAgilentU2701A
    └ Status
       └ SerialPoll
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SerialPoll Statement
    Dim SerialPoll As Int32
    SerialPoll = Driver.Status.SerialPoll
End Sub
```

# TimeoutMilliseconds

### Type

Property

### Function

Set

### Description

This command sets the value, in milliseconds, of the default timeout used by I/O operations. This property provides access to the driver Visa Session Timeout. Only in rare, unusual circumstances should you set this property. Driver methods and properties with operations which take a significant time to perform are responsible for adjusting the I/O timeout to an appropriate value. Some methods provide a *MaxTimeMilliseconds* parameter which gives you direct control over the timeout value for that method. Sometimes, however, increasing the timeout value can work around an obscure driver defect.

### Hierarchy

```
IAgilentU2701A
    └ System
       └ TimeoutMilliseconds
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' TimeoutMilliseconds Statement
    ' To Get TimeoutMilliseconds from the Instrument
    Dim TimeoutMilliseconds As Int32
    TimeoutMilliseconds = Driver.System.TimeoutMilliseconds
    ' To Set TimeoutMilliseconds eg. 5000
    Driver.System.TimeoutMilliseconds = 5000
End Sub
```

# WaitForOperationComplete

### Type

Method

### Function

Set

### Description

This command sets the instrument not to return until previously started operations are completed or the *MaxTimeMilliseconds* time have expired.

### Hierarchy

```
IAgilentU2701A
    └ System
       └ WaitForOperationComplete (MaxTimeMilliseconds)
```

### Return Format

String/BTSR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' WaitForOperationComplete Statement
    Driver.System.WaitForOperationComplete(50000)
End Sub
```

# Disable

### Type

Method

### Function

Set

### Description

This command quickly places the instrument in a state where it has no, or minimal, effect on the external system to which it is connected. This state is not necessarily a known state.

### Hierarchy

```
IAgilentU2701A
    └ Utility
       └ Disable()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Disable Statement
    Driver.Utility.Disable()
End Sub
```

# ErrorQuery

### Type

Method

### Function

Get

### Description

This commad queries the instrument and returns the instrument specific error information. This function can be used when QueryInstrumentStatus is `True` to retrieve error details when the driver detects an instrument error.

### Hierarchy

```
IAgilentU2701A
    └ Utility
       └ ErrorQuery(ErrorCode, ErrorMessage)
```

### Return Format

| Item | Type | Description |
|---|---|---|
| ErrorCode | Long/Int32 | Instrument error code. |
| ErrorMessage | String/BSTR | Instrument error message. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ErrorQuery Statement
    Dim ErrorCode As Int32 = 0
    Dim ErrorMessage As String = ""
    Driver.Utility.ErrorQuery(ErrorCode, ErrorMessage)
End Sub
```

# LockObject

### Type

Method

### Function

Set

### Description

Obtains a multithread lock on the driver after waiting until all other execution threads have released their locks on the instrument session.

### Hierarchy

```
IAgilentU2701A
    └ Utility
       └ LockObject()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' LockObject Statement
    Driver.Utility.LockObject()
End Sub
```

# Reset

### Type

Method

### Function

Set

### Description

This command places the instrument in a known state and configures instrument options on which the IVI specific driver depends (for example, enabling/disabling headers). For an IEEE-488.2 instrument, Reset sends the command string `*RST` to the instrument.

### Hierarchy

```
IAgilentU2701A
    └ Utility
       └ Reset()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Reset Statement
    Driver.Utility.Reset()
End Sub
```

# ResetWithDefaults

### Type

Method

### Function

Set

### Description

This command does the equivalent of Reset and then, disables class extension capability groups, sets attributes to initial values defined by class specs, and configures the driver to option string settings used when Initialize was last executed.

### Hierarchy

```
IAgilentU2701A
    └ Utility
       └ ResetWithDefaults()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ResetWithDefaults Statement
    Driver.Utility.ResetWithDefaults()
End Sub
```

# SelfTest

### Type

Method

### Function

Get

### Description

This command performs an instrument self-test, waits for the instrument to complete the test, and queries the instrument for the results. If the instrument passes the test, *TestResult* is "`0`" and *TestMessage* is "`Self test passed`".

### Hierarchy

```
IAgilentU2701A
    └ Utility
       └ SelfTest(TestResult, TestMessage)
```

### Return Format

| Item | Type | Description |
|---|---|---|
| TestResult | Long/Int32 | Numeric result from the self-test operation. 0 = no error (test passed). |
| TestMessage | String/BSTR | Self-test status message. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SelfTest Statement
    Dim TestResult As Int32 = 0
    Dim TestMessage As String = ""
    Driver.Utility.SelfTest(TestResult, TestMessage)
End Sub
```

# UnlockObject

### Type

Method

### Function

Set

### Description

This command releases a previously obtained multithread lock.

### Hierarchy

```
IAgilentU2701A
    └ Utility
       └ UnlockObject()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' UnlockObject Statement
    Driver.Utility.UnlockObject()
End Sub
```

# Enumeration Members

This section describes the members of each enumeration used in this specific IVI-COM driver.

## AgilentU2701AGlitchConditionEnum

### Description

IVI scope class-compliant values for glitch trigger condition.

### Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A GlitchCondition LessThan | 1 | The oscilloscope triggers when the pulse width is less than the value you specify with the Glitch.Width property. |
| AgilentU2701A GlitchCondition GreaterThan | 2 | The oscilloscope triggers when the pulse width is greater than the value you specify with the Glitch.Width property. |

# AgilentU2701AGlitchPolarityEnum

## Description

IVI scope class-compliant values for glitch trigger polarity.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A GlitchPolarity Positive | 1 | The oscilloscope triggers on a positive glitch. |
| AgilentU2701A GlitchPolarity Negative | 2 | The oscilloscope triggers on a negative glitch. |

# AgilentU2701ASRQReasonEnum

## Description

Enumeration for the *Reason* parameter of the ConfigureServiceRequest method.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A SRQReasonStb MAV | 1 | Message Available. Bit 4 in the status byte. |
| AgilentU2701A SRQReasonStb Msg | 2 | Error/Event Queue Message Available. Bit 2 in the status byte. The Error/Event queue may be read by the ErrorQuery method. |
| AgilentU2701A SRQReasonEsr OPC | 4 | Operation Complete. Bit 0 of the standard event status register. |

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A SRQReasonEsr QueryError | 8 | Query Error. Bit 2 of the standard event status register. |
| AgilentU2701A SRQReasonEsr DeviceError | 16 | Device Dependent Error. Bit 3 of the standard event status register. |
| AgilentU2701A SRQReasonEsr ExecutionError | 32 | Execution Error. Bit 4 of the standard event status register. |
| AgilentU2701A SRQReasonEsr CommandError | 64 | Command Error. Bit 5 of the standard event status register. |
| AgilentU2701A SRQReasonEsr UserRequest | 128 | User Request. Bit 6 of the standard event status register. |
| AgilentU2701A SRQReasonEsr PowerOn | 256 | Power On. Bit 7 of the standard event status register. |
| AgilentU2701A SRQReasonStbTrg | 512 | Trigger Event has occurred. Bit 0 in the status byte register. |
| AgilentU2701A SRQReasonStb User | 1024 | User Event has occurred. Bit 1 in the status byte register. |

# AgilentU2701AStatusRegisterEnum

## Description

Enumeration for the *Register* parameter of the Register property.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A StatusRegister StatusByte | 0 | Status Byte register. Conditions defined by IEEE 488.2. |
| AgilentU2701A StatusRegister StandardEvent | 1 | Standard Event Status register. Conditions defined by IEEE 488.2. |
| AgilentU2701A StatusRegister Operation | 2 | Operation. Conditions which are part of the instrument's normal operation. |
| AgilentU2701A StatusRegister OverloadEvent | 3 | The Overload Status Register captures overload events and probe faults that occur on the input channels. This register does not support the Conditions subregister. |
| AgilentU2701A StatusRegister TriggerEvent | 4 | The Trigger Event Register captures the trigger event. This register does not support the Conditions or Enable subregisters. |
| AgilentU2701A StatusRegister ArmEvent | 5 | The Armed Event Register captures the arm event. An arm event occurs when the oscilloscope has fulfilled all its pre-trigger requirements and is waiting for a trigger event. This register does not support the Conditions or Enable subregisters. |

# AgilentU2701AStatusSubRegisterEnum

## Description

Enumeration for the *SubRegister* parameter of the Register property in the Status interface.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A StatusSubRegister Condition | 0 | Conditions Register. Read only and not available with the Standard Event Status register. Condition bits represent the current instrument state. Reading a condition register does not change the value of the bits. |
| AgilentU2701A StatusSubRegister Event | 3 | Event Register. Read only and not available with the Status Byte register. Event bits capture changes in condition bits. Reading an event register clears it. |
| AgilentU2701A StatusSubRegister Enable | 4 | Enable Register. Can be read or written and is available for all registers. Enable bits select whether an event bit causes a register summary bit, a kind of condition bit, to become True. Thus, an event can propagated to the status byte and finally to request service. Though the bits in the status byte are not technically event bits, the status byte enable register selects which bits in the status byte cause a service request. |

# AgilentU2701ATriggerCouplingEnum

## Description

IVI Scope class-compliant values for trigger Coupling.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A TriggerCouplingAC | 1 | The oscilloscope AC couples the trigger signal. |
| AgilentU2701A TriggerCouplingDC | 2 | The oscilloscope DC couples the trigger signal. |
| AgilentU2701A TriggerCouplingLF Reject | 3 | The oscilloscope filters out the low frequencies from the trigger signal. |
| AgilentU2701A TriggerCouplingHF Reject | 4 | The oscilloscope filters out the high frequencies from the trigger signal. |

# AgilentU2701ATriggerModifierEnum

## Description

IVI Scope class-compliant values for trigger Modifier.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A TriggerModifier None | 1 | The oscilloscope waits until the trigger the end-user specifies occurs. |
| AgilentU2701A TriggerModifier Auto | 2 | The oscilloscope automatically triggers if the configured trigger does not occur within the oscilloscope's timeout period. |

# AgilentU2701ATriggerSlopeEnum

## Description

IVI Scope class-compliant values for edge trigger Slope.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A TriggerSlope Negative | 0 | Triggers will occur on the falling edge. |
| AgilentU2701A TriggerSlope Positive | 1 | Triggers will occur on the rising edge. |
| AgilentU2701A TriggerSlopeEither | 2 | Triggers will occur on either the rising or falling edge. |
| AgilentU2701A TriggerSlope Alternate | 3 | Triggers will occur alternately on the rising and falling edges. |

# AgilentU2701ATriggerTypeEnum

## Description

IVI Scope class-compliant values for trigger Type.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A TriggerEdge | 1 | Configures the oscilloscope for edge triggering. An edge trigger occurs when the trigger signal specified with the Trigger.Source property passes the voltage threshold specified with the Trigger.Level property and has the slope specified with the Trigger.Slope property. |
| AgilentU2701A TriggerGlitch | 2 | Configures the oscilloscope for glitch triggering. Use the IviScopeGlitchTrigger extension properties and methods to configure the trigger. |
| AgilentU2701A TriggerTV | 3 | Configures the oscilloscope for triggering on TV signals. Use the IviScopeTVTrigger extension properties and methods to configure the trigger. |
| AgilentU2701A TriggerWidth | 4 | Configures the oscilloscope for width triggering. Use the IviScopeWidthTrigger extension properties and methods to configure the trigger. |

# AgilentU2701ATVSignalFormatEnum

## Description

IVI Scope class-compliant values for TV trigger SignalFormat.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701ATVSignalFormatNTSC | 1 | Configures the oscilloscope to trigger on the NTSC signal format. |
| AgilentU2701ATVSignalFormatPAL | 2 | Configures the oscilloscope to trigger on the PAL signal format. |
| AgilentU2701ATVSignalFormatSECAM | 3 | Configures the oscilloscope to trigger on the SECAM signal format. |

# AgilentU2701ATVTriggerEventEnum

## Description

IVI Scope class-compliant values for TV trigger Event.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701ATV TriggerEventField1 | 1 | Sets the oscilloscope to trigger on field 1 of the video signal. |
| AgilentU2701ATV TriggerEventField2 | 2 | Sets the oscilloscope to trigger on field 2 of the video signal. |
| AgilentU2701ATV TriggerEventAny Field | 3 | Sets the oscilloscope to trigger on any field. |
| AgilentU2701ATV TriggerEventAny Line | 4 | Sets the oscilloscope to trigger on any line. |
| AgilentU2701ATV TriggerEventField1 LineNumber | 5 | Sets the oscilloscope to trigger on a specific line number in Field 1 you specify with the TV.LineNumber property. |
| AgilentU2701ATV TriggerEventField2 LineNumber | 6 | Sets the oscilloscope to trigger on a specific line number in Field 2 you specify with the TV.LineNumber property. |

# AgilentU2701AVerticalCouplingEnum

## Description

IVI Scope class-compliant values for channel Coupling.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A VerticalCoupling AC | 0 | The oscilloscope AC couples the input signal. |
| AgilentU2701A VerticalCoupling DC | 1 | The oscilloscope DC couples the input signal. |
| AgilentU2701A VerticalCoupling LFReject | 2 | The oscilloscope rejects low frequency. |
| AgilentU2701A VerticalCoupling HFReject | 3 | The oscilloscope rejects high frequency. |

# AgilentU2701AWidthConditionEnum

## Description

IAgilentU2701A instrument-specific values for the polarity of the pulse that triggers the oscilloscope.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A WidthPolarity Positive | 1 | Configures the oscilloscope to trigger on positive pulses that have a width that meets the condition the user specifies with the Width.Condition property. |
| AgilentU2701A WidthPolarity Negative | 2 | Configures the oscilloscope to trigger on negative pulses that have a width that meets the condition the user specifies with the Width.Condition property. |

# AgilentU2701AWidthPolarityEnum

## Description

IAgilentU2701A instrument-specific values for the polarity of the pulse that triggers the oscilloscope.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A WidthPolarity Positive | 1 | Configures the oscilloscope to trigger on positive pulses that have a width that meets the condition the user specifies with the Width.Condition property. |
| AgilentU2701A WidthPolarity Negative | 2 | Configures the oscilloscope to trigger on negative pulses that have a width that meets the condition the user specifies with the Width.Condition property. |

# 3 Acquisition



This chapter describes the acquisition commands used to program the U2701A/U2702A USB modular oscilloscopes over the remote interface. You can use the commands in this chapter to configure the acquisition type, the size of the waveform record, the length of time that corresponds to overall waveform record, and the position of the first point in the waveform record relative to the trigger event.

# ConfigureRecord

### Type

Method

### Function

Set

### Description

This command configures the most commonly used properties of the oscilloscope channel sub-system. Use this command to enable or disable the channel and to set the range, offset, coupling, and probe attenuation values.

### Hierarchy

```
IAgilentU2701A
    └ Acquisition
       └ ConfigureRecord(TimePerRecord, AcquisitionStartTime)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| TimePerRecord | Double | Specifies the time per record. This value sets the Horizontal TimePerRecord property. |
| AcquisitionStart Time | Double | Specifies the position of the first point in the waveform record relative to the trigger event. This value sets the Acquisition.StartTime property. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ConfigureRecord Statement
    Driver.Acquisition.ConfigureRecord(0.001, 0.0)
End Sub
```

# Interpolation

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the interpolation method the oscilloscope uses when it cannot sample a voltage for every point in the waveform record.

### Hierarchy

```
IAgilentU2701A
    └ Acquisition
      └ Interpolation
```

### Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A InterpolationEnum | Enum | See "AgilentU2701AInterpolationEnum" on page 116. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Interpolation Statement
    ' To Get Interpolation from Instrument
    Dim Interpolation As New AgilentU2701AInterpolationEnum
    Interpolation = Driver.Acquisition.Interpolation
    ' To Set Interpolation eg. SineX
    Driver.Acquisition.Interpolation =
     AgilentU2701AInterpolationEnum.AgilentU2701AInterpolationSin
     eX
End Sub
```

# NumberOfAverages

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the number of waveforms the oscilloscope acquires and averages before returning to idle state.

### Hierarchy

```
IAgilentU2701A
    └ Acquisition
       └ NumberOfAverages
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' NumberOfAverage Statement
    ' To Get Number of Average from Instrument
    Dim NumberOfAverage As Int32
    NumberOfAverage = Driver.Acquisition.NumberOfAverages
    ' To Set Number of Average eg. 30
    Driver.Acquisition.NumberOfAverages = 30
End Sub
```

# RecordLength

### Type

Property

### Function

Get

### Description

This command returns the actual number of points the oscilloscope acquires for each channel. It is equal to or greater than the minimum number of points specified with the IviScopeAcquisition.NumberOfPointsMin command.

### Hierarchy

```
IAgilentU2701A
    └ Acquisition
       └ RecordLength
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' RecordLength Statement
    Dim RecordLength As Long
    RecordLength = Driver.Acquisition.RecordLength
End Sub
```

# SampleMode

### Type

Property

### Function

Get

### Description

This command returns the sample mode that the oscilloscope is currently using.

### Hierarchy

```
IAgilentU2701A
    └ Acquisition
       └ SampleMode
```

### Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A SampleMode Enum | Enum | See "AgilentU2701ASampleModeEnum" on page 116. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SampleMode Statement
    Dim SampleMode As AgilentU2701ASampleModeEnum
    SampleMode = Driver.Acquisition.SampleMode
End Sub
```

# SampleRate

### Type

Property

### Function

Get

### Description

This command returns the effective digitizing rate using the current configuration. The units are expressed in samples per second.

### Hierarchy

```
IAgilentU2701A
    └ Acquisition
       └ SampleRate
```

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SampleRate Statement
    Dim SampleRate As Double
    SampleRate = Driver.Acquisition.SampleRate
End Sub
```

# StartTime

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the length of time from the trigger event to the first point in the waveform record. The units are expressed in seconds. If positive, the first point in the waveform occurs after the trigger. If negative, the first point in the waveform occurs before the trigger.

### Hierarchy

```
IAgilentU2701A
    └ Acquisition
       └ StartTime
```

### Return Format

Double

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' StartTime Statement
    ' To Get Start Time from the Instrument
    Dim StartTime As Double
    StartTime = Driver.Acquisition.StartTime
    ' To Set Start Time eg. 0.0 s
    Driver.Acquisition.StartTime = 0.0
End Sub
```

# TimePerRecord

### Type

Property

### Function

Get and Set

### Description

This command returns/sets the time in seconds that corresponds to the record length.

### Hierarchy

```
IAgilentU2701A
    └ Acquisition
       └ TimePerRecord
```

### Return Format

Double

### Remarks

The time per record length is equivalent to the Time/Div multiplied by 10 at the scope setting. The Maximum data point per second you can achieve is 1250, except when in interleave mode (only available with the Agilent Measurement Manager), you can achieve 2000 data points.

| Time/Div | TimePer Record | Maximum Data Point |
|---|---|---|
| 1 ns | 10 ns | 5 |
| 2 ns | 20 ns | 10 |
| 5 ns | 50 ns | 25 |
| 10 ns | 100 ns | 50 |
| 20 ns | 200 ns | 100 |
| 50 ns | 500 ns | 250 |
| 100 ns | 1 µs | 500 |
| 200 ns | 2 µs | 100 |
| 500 ns | 5 µs | 1250 |
| 1 µs | 10 µs | 1250 |
| 2 µs | 20 µs | 1250 |
| 5 µs | 50 µs | 1250 |
| 10 µs | 100 µs | 1250 |
| 20 µs | 200 µs | 1250 |
| 50 µs | 500 µs | 1250 |
| 100 µs | 1 ms | 1250 |
| 200 µs | 2 ms | 1250 |
| 500 µs | 5 ms | 1250 |

| Time/Div | TimePer Record | Maximum Data Point |
|---|---|---|
| 1 ms | 10 ms | 1250 |
| 2 ms | 20 ms | 1250 |
| 5 ms | 50 ms | 1250 |
| 10 ms | 100 ms | 1250 |
| 20 ms | 200 ms | 1250 |
| 50 ms | 500 ms | 1250 |
| 100 ms | 1 s | 1250 |
| 200 ms | 2 s | 1250 |
| 500 ms | 5 s | 1250 |
| 1 s | 10 s | 1250 |
| 2 s | 20 s | 1250 |
| 5 s | 50 s | 1250 |
| 10 s | 100 s | 1250 |
| 20 s | 200 s | 1250 |
| 50 s | 500 s | 1250 |
| | | |
| | | |
| | | |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' TimePerRecord Statement
    ' To Get Time Per Record from the Instrument
    Dim TimePerRecord As Double
    TimePerRecord = Driver.Acquisition.TimePerRecord
    ' To Set Time Per Record eg. 0.001
    Driver.Acquisition.TimePerRecord = 0.001
End Sub
```

# Type

## Type

Property

## Function

Get and Set

## Description

This commands returns/sets how the oscilloscope acquires data and fills the waveform record. When set to `Envelope` or `Peak Detect`, the oscilloscope acquires minimum and maximum waveforms.

## Hierarchy

```
IAgilentU2701A
    └ Acquisition
      └ Type
```

## Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A AcquisitionType Enum | Enum | See "AgilentU2701AAcquisitionTypeEnum" on page 115. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Type Statement
    ' To Get Type from the Instrument
    Dim Type As AgilentU2701AAcquisitionTypeEnum
    Type = Driver.Acquisition.Type
    ' To Set Type eg. Normal
    Driver.Acquisition.Type =
     AgilentU2701AAcquisitionTypeEnum.AgilentU2701AAcquisitionTyp
     eNormal
End Sub
```

# Enumeration Members

This section describes the members of each enumeration used in this specific IVI-COM driver.

## AgilentU2701AAcquisitionStatusEnum

### Description

IVI scope class-compliant values for the *Status* parameter of the acquisition status method.

### Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A AcqInProgress | 0 | The oscilloscope is still acquiring data. |
| AgilentU2701A AcqComplete | 1 | The oscilloscope has completed the acquisition. |
| AgilentU2701A AcqStatus Unknown | –1 | The oscilloscope cannot determine the status of the acquisition. |

# AgilentU2701AAcquisitionTypeEnum

## Description

IVI scope class-compliant values for acquisition type.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A AcquisitionType Normal | 0 | Configures the oscilloscope to acquire one sample for each point in the waveform record. The oscilloscope uses real-time or equivalent time sampling. |
| AgilentU2701A AcquisitionType PeakDetect | 1 | Sets the oscilloscope to the peak-detect acquisition mode. The oscilloscope oversamples the input signal and keeps the minimum and maximum values that correspond to each position in the waveform record. The oscilloscope uses only real-time sampling. |
| AgilentU2701A AcquisitionType Average | 2 | Configures the oscilloscope to acquire multiple waveforms and calculates the average value for each point in the waveform record. The end-user specifies the number of waveforms to acquire with the NumberOfAverages property. The oscilloscope uses real-time or equivalent time sampling. |

# AgilentU2701AInterpolationEnum

## Description

IVI scope class-compliant values for acquisition interpolation.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A InterpolationNone | 1 | The oscilloscope does not interpolate points in the waveform. Instead, the driver sets every element in the waveform record for which the oscilloscope cannot receive a value to an IEEE-defined NaN (Not-a-Number) value. Use the IsWaveformElementInvalid method to determine if the waveform record element is invalid. |
| AgilentU2701A InterpolationSineX | 2 | The oscilloscope uses a sin(x)/x calculation to interpolate a value when it cannot resolve a voltage in the waveform record. |

# AgilentU2701ASampleModeEnum

## Description

IVI Scope class-compliant values for acquisition SampleMode.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A SampleModeReal Time | 0 | Indicates that the oscilloscope is using real-time sampling. |

# 4 Waveform Display



This chapter describes the waveform display commands used to program the U2701A/U2702A USB modular oscilloscopes over the remote interface. The statements in this section is used to acquire waveform data from the oscilloscope. You can use the commands in this section to initiate, acquire, and abort measurements.

# Abort

### Type

Method

### Function

Set

### Description

This command aborts an acquisition and returns the oscilloscope to the Idle state.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Abort()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Abort Statement
    Driver.Measurements.Abort()
End Sub
```

# AutoSetup

### Type

Method

### Function

Set

### Description

This command automatically configures all the oscilloscopes settings based on the input signals.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ AutoSetup()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' AutoSetup Statement
    Driver.Measurements.AutoSetup()
End Sub
```

# Count

### Type

Property

### Function

Get

### Description

This command returns the number of measurements.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Count
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Count Statement
    Dim Count As Int32
    Count = Driver.Measurements.Count
End Sub
```

# Initiate

### Type

Method

### Function

Set

### Description

This command initiates a waveform acquisition. The oscilloscope leaves the Idle state and waits for a trigger. The oscilloscope acquires a waveform for each enabled channel.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Initiate()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Initiate Statement
    Driver.Measurements.Initiate()
End Sub
```

# IsWaveformElementInvalid

### Type

Method

### Function

Get

### Description

This command returns `False` if an element in a waveform array returned by the driver contains a valid voltage. This command returns `True` if an element in a waveform array returned by the driver contains a value indicating that the oscilloscope could not sample a voltage.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ IsWaveformElementInvalid(Element)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Element | Double | Pass one of the values from the waveform array returned by the read and fetch waveform methods. |

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' IsWaveformElementInvalid Statement
    Dim Element As Double = 0
    Dim IsWaveformElementInvalid As Boolean
    IsWaveformElementInvalid =
     Driver.Measurements.IsWaveformElementInvalid(Element)
End Sub
```

# Item

This is an interface reference pointer to the IAgilentU2701AMeasurement interface which is selected by the measurement name.

## FetchWaveform

### Type

Method

### Function

Get

### Description

This command returns a previously acquired waveform for this channel. The acquisition must be made prior to calling this method. Call this method separately for each channel.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Item(Name)
          └ FetchWaveform(WaveformArray, InitialX, XIncrement)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

| Item | Type | Description |
|---|---|---|
| WaveformArray | Double | The array contains the acquired waveform. Units for the individual array elements are expressed in volts. |
| InitialX | Double | The time in relation to the trigger event of the first point in the waveform in seconds. Negative values mean that the first point in the waveform array was acquired before the trigger event. |
| XIncrement | Double | The effective time between points in the acquired waveform in seconds. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Initialization required
    Driver.Measurements.Initiate()
    ' Get Record Length
    Dim RecordLength As Int32
    RecordLength = Driver.Acquisition.RecordLength
    ' FetchWaveform Statement
    Dim WaveformArray() As Double
    Dim InitialX As Double
    Dim XIncrement As Double
    ReDim WaveformArray(RecordLength - 1)
    Driver.Measurements.Item("Channel1").FetchWaveform(WaveformAr
     ray, InitialX, XIncrement)
End Sub
```

# ReadWaveform

### Type

Method

### Function

Get

### Description

This command initiates an acquisition on all enabled channels, waits (up to *MaxTime*) for the acquisition to complete, and returns the waveform for this channel. Call FetchWaveform to obtain the waveforms for other channels.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Item(Name)
          └ ReadWaveform(MaxTimeMilliseconds, WaveformArray,
            InitialX, XIncrement)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

| Item | Type | Description |
|---|---|---|
| MaxTime Milliseconds | Long/Int32 | Specifies the maximum time the end-user allows for this method to complete in milliseconds. |
| WaveformArray | Double | The array contains the acquired waveform. Units for the individual array elements are expressed in volts. |
| InitialX | Double | The time in relation to the trigger event of the first point in the waveform in seconds. Negative values mean that the first point in the waveform array was acquired before the trigger event. |
| XIncrement | Double | The effective time between points in the acquired waveform in seconds. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Get Record Length
    Dim RecordLength As Int32
    RecordLength = Driver.Acquisition.RecordLength
    ' ReadWaveform Statement
    Dim WaveformArray() As Double
    Dim InitialX As Double
    Dim XIncrement As Double
    ReDim WaveformArray(RecordLength - 1)
    Driver.Measurements.Item("Channel1").ReadWaveform(50000,
     WaveformArray, InitialX, XIncrement)
End Sub
```

# 5 Post Analysis



This chapter describes the post analysis commands used to program the U2701A/U2702A USB modular oscilloscopes over the remote interface. The statements in this section is used to analyze the waveform data acquired from the oscilloscope. You can also use the commands in this section to initiate, acquire, and abort measurements.

# Abort

### Type

Method

### Function

Set

### Description

This command aborts an acquisition and returns the oscilloscope to the Idle state.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Abort()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Abort Statement
    Driver.Measurements.Abort()
End Sub
```

# AutoSetup

### Type

Method

### Function

Set

### Description

This command automatically configures all the oscilloscopes settings based on the input signals.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ AutoSetup()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' AutoSetup Statement
    Driver.Measurements.AutoSetup()
End Sub
```

# Count

### Type

Property

### Function

Get

### Description

This command returns the number of measurements.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Count
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Count Statement
    Dim Count As Int32
    Count = Driver.Measurements.Count
End Sub
```

# Initiate

### Type

Method

### Function

Set

### Description

This command initiates a waveform acquisition. The oscilloscope leaves the Idle state and waits for a trigger. The oscilloscope acquires a waveform for each enabled channel.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Initiate()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Initiate Statement
    Driver.Measurements.Initiate()
End Sub
```

# IsWaveformElementInvalid

### Type

Method

### Function

Get

### Description

This command returns `False` if an element in a waveform array returned by the driver contains a valid voltage. This command returns `True` if an element in a waveform array returned by the driver contains a value indicating that the oscilloscope could not sample a voltage.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ IsWaveformElementInvalid(Element)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Element | Double | Pass one of the values from the waveform array returned by the read and fetch waveform methods. |

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' IsWaveformElementInvalid Statement
    Dim Element As Double = 0
    Dim IsWaveformElementInvalid As Boolean
    IsWaveformElementInvalid =
     Driver.Measurements.IsWaveformElementInvalid(Element)
End Sub
```

# Item

This is an interface reference pointer to the IAgilentU2701AMeasurement interface which is selected by the measurement name.

## FetchWaveform

### Type

Method

### Function

Get

### Description

This command returns a previously acquired waveform for this channel. The acquisition must be made prior to calling this method. Call this method separately for each channel.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Item(Name)
          └ FetchWaveform(WaveformArray, InitialX, XIncrement)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

| Item | Type | Description |
|---|---|---|
| WaveformArray | Double | The array contains the acquired waveform. Units for the individual array elements are expressed in volts. |
| InitialX | Double | The time in relation to the trigger event of the first point in the waveform in seconds. Negative values mean that the first point in the waveform array was acquired before the trigger event. |
| XIncrement | Double | The effective time between points in the acquired waveform in seconds. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Initialization required
    Driver.Measurements.Initiate()
    ' Get Record Length
    Dim RecordLength As Int32
    RecordLength = Driver.Acquisition.RecordLength
    ' FetchWaveform Statement
    Dim WaveformArray() As Double
    Dim InitialX As Double
    Dim XIncrement As Double
    ReDim WaveformArray(RecordLength - 1)
    Driver.Measurements.Item("Channel1").FetchWaveform(WaveformAr
     ray, InitialX, XIncrement)
End Sub
```

# FetchWaveformMeasurement

## Type

Method

## Function

Get

## Description

This command returns a previously acquired waveform measurement for this channel. The acquisition must be made prior to calling this method. Call this method separately for each measurement.

## Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Item(Name)
          └ FetchWaveformMeasurement (MeasFunction, Measurement)
```

## Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

| Item | Type | Description |
|---|---|---|
| MeasFunction | AgilentU2701A Measurement Enum | Characteristic of the acquired waveform to be measured. See "AgilentU2701AMeasurementEnum" on page 150 for more information. |
| Measurement | Double | The measured value. The units depend on the measurement that the user specifies with the *MeasFunction* parameter. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Initialization required
    Driver.Measurements.Initiate()
    ' FetchWaveformMeasurement Statement
    Dim Measurement As Double
    Driver.Measurements.Item("Channel1").FetchWaveformMeasurement
     (AgilentU2701AMeasurementEnum.AgilentU2701AMeasurementFreque
     ncy, Measurement)
End Sub
```

# ReadFullWaveform

## Type

Method

## Function

Get

## Description

This command pulls the data directly from the device without data manipulation (interleaving or value conversion). It will automatically detect the 16M/32M data size for each channel.

## Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Item(Name)
          └ ReadFullWaveform(ref Data, ref Length)
```

## Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

## Return Format

| Item | Type | Description |
|---|---|---|
| Data | Array | The data that contains a raw waveform data in bytes. |
| Length | Long/Int32 | The length of the data size. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ReadFullWaveform Statement
    Dim Data() As Byte
    Dim Length As Int32
    Driver.Measurements.Item("Channel1").ReadFullWaveform(Data,
     Length)
End Sub
```

# ReadWaveform

### Type

Method

### Function

Get

### Description

This command initiates an acquisition on all enabled channels, waits (up to *MaxTime*) for the acquisition to complete, and returns the waveform for this channel. Call FetchWaveform to obtain the waveforms for other channels.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Item(Name)
          └ ReadWaveform(MaxTimeMilliseconds, WaveformArray,
            InitialX, XIncrement)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

| Item | Type | Description |
|---|---|---|
| MaxTime Milliseconds | Long/Int32 | Specifies the maximum time the end-user allows for this method to complete in milliseconds. |
| WaveformArray | Double | The array contains the acquired waveform. Units for the individual array elements are expressed in volts. |
| InitialX | Double | The time in relation to the trigger event of the first point in the waveform in seconds. Negative values mean that the first point in the waveform array was acquired before the trigger event. |
| XIncrement | Double | The effective time between points in the acquired waveform in seconds. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Get Record Length
    Dim RecordLength As Int32
    RecordLength = Driver.Acquisition.RecordLength
    ' ReadWaveform Statement
    Dim WaveformArray() As Double
    Dim InitialX As Double
    Dim XIncrement As Double
    ReDim WaveformArray(RecordLength - 1)
    Driver.Measurements.Item("Channel1").ReadWaveform(50000,
     WaveformArray, InitialX, XIncrement)
End Sub
```

# ReadWaveformMeasurement

## Type

Method

## Function

Get

## Description

This command returns a previously acquired waveform for this channel. The acquisition must be made prior to calling this method. Call this method separately for each channel.

## Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Item(Name)
          └ ReadWaveformMeasurement (MeasFunction,
            MaxTimeMilliseconds, Measurement)
```

## Parameters

| Item | Type | Description |
|---|---|---|
| Name | String | The name of a measurement. This command is used to select the channel the function is to be run on. (For example: Channel 1, Channel 2, and so on.) |

### Return Format

| Item | Type | Description |
|---|---|---|
| MeasFunction | AgilentU2701A Measurement Enum | Characteristic of the acquired waveform to be measured. See "AgilentU2701AMeasurementEnum" on page 150 for more information. |
| MaxTime Milliseconds | Long/Int32 | Specifies the maximum time the end-user allows for this method to complete in milliseconds. |
| Measurement | Double | The measured value. The units depend on the measurement that the user specifies with the *MeasFunction* parameter. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Initialization required
    Driver.Measurements.Initiate()
    ' ReadWaveformMeasurement Statement
    Dim Measurement As Double
    Driver.Measurements.Item("Channel1").ReadWaveformMeasurement(
     AgilentU2701AMeasurementEnum.AgilentU2701AMeasurementFrequen
     cy, 50000, Measurement)
End Sub
```

# MathFunction

### Type

Method

### Function

Set

### Description

This command configures a mathematical operation for Channel 1 and 2. The operation and reverse will be set to "ADD" and "False" by default.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
      └ MathFunction(Operation, Reverse)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Operation | AgilentU2701A Measurement Enum | Channel 1 and Channel 2 math operation. |
| Reverse | Boolean | The operation order of Channel 1 and Channel 2. If True, Channel 2 <operation> Channel 1. If False ,Channel 1 <operation> Channel 2 |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' MathFunction Statement
    Driver.Measurements.MathFunction(AgilentU2701AMathOperationEn
     um.AgilentU2701AMathOperationAdd, True)
End Sub
```

# Name

### Type

Property

### Function

Get

### Description

This command returns the measurement name for a given index.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Name(Index)
```

### Parameters

| Item | Type | Description |
|---|---|---|
| Index | Long/Int32 | One based index into the collection of measurements. |

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Name Statement
    Dim Name As String
    Name = Driver.Measurements.Name(1)
End Sub
```

# Status

### Type

Method

### Function

Get

### Description

This command returns whether an acquisition is in progress, complete, or if the status is unknown.

### Hierarchy

```
IAgilentU2701A
    └ Measurements
       └ Status()
```

### Return Format

| Item | Type | Description |
|---|---|---|
| AgilentU2701A AcquisitionStatus Enum | Enum | See "AgilentU2701AAcquisitionStatusEnum" on page 114. |

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Status Statement
    Driver.Measurements.Status()
End Sub
```

# Enumeration Members

This section describes the members of each enumeration used in this specific IVI-COM driver.

## AgilentU2701AMathOperationEnum

### Description

IAgilentU2701A instrument-specific values for channels mathematical operations.

### Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A MathOperation Add | 0 | Add operation between two channels. |
| AgilentU2701A MathOperation Subtract | 1 | Subtract operation between two channels. |
| AgilentU2701A MathOperation Multiply | 2 | Multiply operation between two channels. |
| AgilentU2701A MathOperation Divide | 3 | Divide operation between two channels. |

# AgilentU2701AMeasurementEnum

## Description

IVI scope class-compliant values for the *MeasFunction* parameter read and fetch methods.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A MeasurementRise Time | 0 | The length of time for a rising edge of the signal to rise from the low reference level to the high reference level. The units are expressed in seconds. |
| AgilentU2701A MeasurementFall Time | 1 | The length of time for a falling edge of the signal to fall from the high reference level to the low reference level. The units are expressed in seconds. |
| AgilentU2701A Measurement Frequency | 2 | The frequency of one complete cycle in the waveform. The units are expressed in hertz. |
| AgilentU2701A Measurement Period | 3 | The length of time of one complete cycle in the waveform. The units are expressed in seconds. |
| AgilentU2701A Measurement VoltageRMS | 4 | The true Root Mean Square voltage of the entire waveform. The units are expressed in volts. |
| AgilentU2701A Measurement VoltageRMSAC | 5 | The true Root Mean Square voltage(AC) of the entire waveform. The units are expressed in volts. |
| AgilentU2701A Measurement VoltagePeakTo Peak | 6 | The absolute difference between the Voltage Max and the Voltage Min. The units are expressed in volts. |
| AgilentU2701A Measurement VoltageMax | 7 | The maximum amplitude found in the entire waveform. The units are expressed in volts. |
| AgilentU2701A Measurement VoltageMin | 8 | The minimum amplitude found in the entire waveform. The units are expressed in volts. |

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A Measurement VoltageHigh | 9 | The voltage that corresponds to 100% when using the reference levels. The oscilloscope calculates this value using either the min/max or histogram methods. The min/max method uses the maximum value found. The histogram method uses a common value found above the middle of the waveform. The units are expressed in volts. |
| AgilentU2701A Measurement VoltageLow | 10 | The voltage that corresponds to 0% when using the reference levels. The oscilloscope calculates this value using either the min/max or histogram methods. The min/max method uses the minimum value found. The histogram method uses a common value found below the middle of the waveform. The units are expressed in volts. |
| AgilentU2701A Measurement VoltageAverage | 11 | The arithmetic average in volts measured over the entire waveform. The units are expressed in volts. |
| AgilentU2701A Measurement WidthNeg | 12 | The length of time between the mid reference level points of a negative pulse in the waveform. The units are expressed in seconds. |
| AgilentU2701A Measurement WidthPos | 13 | The length of time between the mid reference level points of a positive pulse in the waveform. The units are expressed in seconds. |
| AgilentU2701A MeasurementDuty CycleNeg | 14 | The ratio of the WidthNeg to the Period of an integer number of cycles in the waveform expressed as a percentage. |
| AgilentU2701A MeasurementDuty CyclePos | 15 | The ratio of the WidthPos width to the Period of an integer number of cycles in the waveform expressed as a percentage. |
| AgilentU2701A Measurement Amplitude | 16 | The VoltageHigh less the VoltageLow expressed in volts over the entire waveform. |
| AgilentU2701A Measurement VoltageCycleRMS | 17 | The true root mean square voltage over an integer number of cycles in the waveform. The units are expressed in volts. |
| AgilentU2701A MeasurementOver Shoot | 18 | The relative waveform distortion that follows an edge transition. |

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A Measurement Preshoot | 19 | The relative waveform distortion that precedes an edge transition. |
| AgilentU2701A Measurement Phase | 20 | Phase. |
| AgilentU2701A Measurement Delay | 21 | Delay. |
| AgilentU2701A Measurement CrestFactor | 22 | Crest factor. |
| AgilentU2701A Measurement StandardDeviation | 23 | Standard deviation. |
| AgilentU2701A Measurement TimeAtMax | 24 | Time at maximum voltage. |
| AgilentU2701A Measurement TimeAtMin | 25 | Time at minimum voltage. |

# AgilentU2701ATimeOutEnum

## Description

IVI Scope class-compliant values for *maxTime* parameter of the measurement Read and Fetch methods.

## Members

| Member name | Value | Description |
|---|---|---|
| AgilentU2701A TimeOut Immediate | 0 | This method returns immediately. If no measurement value exists, an error is returned. |
| AgilentU2701A TimeOutInfinite | –1 | The method waits indefinitely for the measurement to complete. |

# 5 Post Analysis

# 6 Others



This chapter describes the other commands used to program the U2701A/U2702A USB modular oscilloscopes over the remote interface. You can use the commands in this chapter to get and set some of the oscilloscope properties. You can also calibrate the instrument and perform various driver operations.

# Calibrate

### Type

Method

### Function

Set

### Description

This command calibrates the instrument. This method might take several minutes to complete.

### Hierarchy

```
IAgilentU2701A
    └ Calibration
       └ Calibrate()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Calibrate Statement
    Driver.Calibration.Calibrate()
End Sub
```

# Date

### Type

Property

### Function

Get

### Description

This command returns the date the instrument was last calibrated.

### Hierarchy

```
IAgilentU2701A
    └ Calibration
       └ Date
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Date Statement
    Dim LastCalDate As String
    LastCalDate = Driver.Calibration.Date
End Sub
```

# Label

### Type

Property

### Function

Get

### Description

This command returns the label set during the last calibration.

### Hierarchy

```
IAgilentU2701A
    └ Calibration
       └ Label
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Label Statement
    Dim Label As String
    Label = Driver.Calibration.Label
End Sub
```

# Time

### Type

Property

### Function

Get

### Description

This command returns the time the instrument was last calibrated.

### Hierarchy

```
IAgilentU2701A
    └ Calibration
       └ Time
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Time Statement
    Dim LastCalTime As String
    LastCalTime = Driver.Calibration.Time
End Sub
```

# Cache

### Type

Property

### Function

Get and Set

### Description

Drivers may choose to always cache some instrument settings, never cache others, or optionally cache others to avoid unnecessary I/O activities to the instrument. If this command is set to `True`, the driver caches optionally cache instrument settings.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ Cache
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Cache Statement
    ' To Get Cache from the Instrument
    Dim Cache As Boolean
    Cache = Driver.DriverOperation.Cache
    ' To Set Cache eg. True
    Driver.DriverOperation.Cache = True
End Sub
```

# ClearInterchangeWarnings

### Type

Method

### Function

Set

### Description

This command clears the list of interchangeability warnings that the IVI specific driver maintains.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ ClearInterchangeWarnings
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Clear InterchangeWarnings Statement
    Driver.DriverOperation.ClearInterchangeWarnings()
End Sub
```

# DriverSetup

### Type

Property

### Function

Get

### Description

This command returns the driver setup string. It is either specified in the configuration store or passed in the *OptionString* parameter of the function. Driver setup is empty if the driver is not initialized.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ DriverSetup
```

### Return Format

String/BTSR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' DriverSetup Statement
    Dim DriverSetup As String
    Cache = Driver.DriverOperation.Cache
    ' To Set Cache eg. True
    DriverSetup = Driver.DriverOperation.DriverSetup
End Sub
```

# GetNextCoercionRecord

### Type

Method

### Function

Get

### Description

This command returns the oldest record from the coercion record list. Records are only added to the list if RecordCoercions is set to `True`.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
      └ GetNextCoercionRecord()
```

### Return Format

String/BTSR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' GetNextCoercionRecord Statement
    Dim DriverSetup As String
    GetNextCoercionRecord =
     Driver.DriverOperation.GetNextCoercionRecord()
End Sub
```

# GetNextInterchangeWarning

### Type

Method

### Function

Get

### Description

This command returns the oldest warning from the interchange warning list. Records are only added to the list if InterchangeCheck is set to `True`.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ GetNextInterchangeWarning()
```

### Return Format

String/BTSR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' GetNextInterchangeWarning Statement
    Dim GetNextInterchangeWarning As String
    GetNextInterchangeWarning =
     Driver.DriverOperation.GetNextInterchangeWarning()
End Sub
```

# InterchangeCheck

### Type

Property

### Function

Get and Set

### Description

If this command is set to `True`, the driver maintains a record of interchangeability warnings. If the driver does not support interchangeability checking, attempts to set InterchangeCheck to `True` returns an error.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ InterchangeCheck
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' InterchangeCheck Statement
    ' To Get InterchangeCheck from the Instrument
    Dim InterchangeCheck As Boolean
    InterchangeCheck = Driver.DriverOperation.InterchangeCheck
    ' To Set InterchangeCheck eg. True
    Driver.DriverOperation.InterchangeCheck = True
End Sub
```

# InvalidateAllAttributes

### Type

Method

### Function

Set

### Description

This command invalidates all of the driver cached values.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ InvalidateAllAttributes()
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' InvalidateAllAttributes Statement
    Driver.DriverOperation.InvalidateAllAttributes()
End Sub
```

# IoResourceDescriptor

### Type

Property

### Function

Get

### Description

This command returns the instrument resource descriptor. The resource descriptor specifies the connection to a physical device. It is either specified in the configuration store or passed in the *ResourceName* parameter of the Initialize function. It is empty if the driver is not initialized.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
      └ IoResourceDescriptor
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' IoResourceDescriptor Statement
    Dim IoResourceDescriptor As String
    IoResourceDescriptor =
     Driver.DriverOperation.IoResourceDescriptor
End Sub
```

# LogicalName

### Type

Property

### Function

Get

### Description

This command returns the instrument logical name. The logical name identifies a driver session in the configuration store. If logical name is not empty, the driver was initialized from the information in the driver session. If it is empty, the driver was initialized without using the configuration store.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
      └ LogicalName
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' IoResourceDescriptor Statement
    Dim LogicalName As String
    LogicalName = Driver.DriverOperation.LogicalName
End Sub
```

# QueryInstrumentStatus

### Type

Property

### Function

Get and Set

### Description

If this command is set to `True`, the driver queries the instrument status at the end of each method or property that performs I/O to the instrument. If an error is reported, use ErrorQuery to retrieve error messages one at a time from the instrument.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ QueryInstrumentStatus
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' QueryInstrumentStatus Statement
    ' To Get QueryInstrumentStatus from the Instrument
    Dim QueryInstrumentStatus As Boolean
    QueryInstrumentStatus =
     Driver.DriverOperation.QueryInstrumentStatus
    ' To Set QueryInstrumentStatus eg. True
    Driver.DriverOperation.QueryInstrumentStatus = True
End Sub
```

# RangeCheck

### Type

Property

### Function

Get and Set

### Description

Drivers may choose to always validate some property or parameter values, never validate others, and optionally validate others, to avoid sending invalid commands to the instrument. If this command is set to `True`, the driver performs optional validations.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ RangeCheck
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' RangeCheck Statement
    ' To Get RangeCheck from the Instrument
    Dim RangeCheck As Boolean
    RangeCheck = Driver.DriverOperation.RangeCheck
    ' To Set RangeCheck eg. True
    Driver.DriverOperation.RangeCheck = True
End Sub
```

# RecordCoercions

### Type

Property

### Function

Get and Set

### Description

If this command is set to `True`, the driver keeps a list of the value coercions it makes for *ViInt32* and *ViReal64* attributes. If the driver does not support coercion recording, attempts to set RecordCoercions to `True` will return an error.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ RecordCoercions
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' RecordCoercions Statement
    ' To Get RecordCoercions from the Instrument
    Dim RecordCoercions As Boolean
    RecordCoercions = Driver.DriverOperation.RecordCoercions
    ' To Set RecordCoercions eg. True
    Driver.DriverOperation.RecordCoercions = True
End Sub
```

# ResetInterchangeCheck

### Type

Method

### Function

Set

### Description

This command resets the interchangeability checking algorithms of the driver so that methods and properties that were executed prior to calling this function have no affect on whether future calls to the driver will generate interchangeability warnings or not.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
       └ ResetInterchangeCheck()
```

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' ResetInterchangeCheck Statement
    Driver.DriverOperation.ResetInterchangeCheck()
End Sub
```

# Simulate

### Type

Property

### Function

Get and Set

### Description

If this command is set to `True`, the driver does not perform I/O to the instrument, and returns simulated values for output parameters instead.

### Hierarchy

```
IAgilentU2701A
    └ DriverOperation
      └ Simulate
```

### Return Format

Boolean

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Simulate Statement
    ' To Get Simulate from the Instrument
    Dim Simulate As Boolean
    Simulate = Driver.DriverOperation.Simulate
    ' To Set Simulate eg. True
    Driver.DriverOperation.Simulate = True
End Sub
```

# Description

### Type

Property

### Function

Get

### Description

This command returns a brief description of the implementing component. Description is limited to 256 bytes.

### Hierarchy

```
IAgilentU2701A
    └ Identity
      └ Description
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Description Statement
    Dim Description As String
    Description = Driver.Identity.Description
End Sub
```

# GroupCapabilities

### Type

Property

### Function

Get

### Description

This command returns a comma-separated list of the class capability groups implemented by the driver. Capability group names are documented in the IVI class specifications. If the driver is not class compliant, the driver returns an empty string.

### Hierarchy

```
IAgilentU2701A
    └ Identity
       └ GroupCapabilities
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' GroupCapabilities Statement
    Dim GroupCapabilities As String
    GroupCapabilities = Driver.Identity.GroupCapabilities
End Sub
```

# Identifier

### Type

Property

### Function

Get

### Description

This command returns the case-sensitive unique identifier of the implementing IVI-COM instrument driver.

### Hierarchy

```
AgilentU2701A
    └ Identity
      └ Identifier
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Identifier Statement
    Dim Identifier As String
    Identifier = Driver.Identity.Identifier
End Sub
```

# InstrumentFirmwareRevision

### Type

Property

### Function

Get

### Description

This command returns the firmware revision reported by the physical instrument. If Simulate is enabled or the instrument is not capable of reporting the firmware revision, a string is returned that explains the condition.

### Hierarchy

```
IAgilentU2701A
    └ Identity
       └ InstrumentFirmwareRevision
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' InstrumentFirmwareRevision Statement
    Dim InstrumentFirmwareRevision As String
    InstrumentFirmwareRevision =
     Driver.Identity.InstrumentFirmwareRevision
End Sub
```

# InstrumentManufacturer

### Type

Property

### Function

Get

### Description

This command returns the name of the manufacturer reported by the physical instrument. If Simulate is enabled or the instrument is not capable of reporting the name of the manufacturer, a string is returned that explains the condition. InstrumentManufacturer is limited to 256 bytes.

### Hierarchy

```
IAgilentU2701A
    └ Identity
      └ InstrumentManufacturer
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' InstrumentManufacturer Statement
    Dim InstrumentManufacturer As String
    InstrumentManufacturer =
     Driver.Identity.InstrumentManufacturer
End Sub
```

# InstrumentModel

### Type

Property

### Function

Get

### Description

This command returns the model number or name reported by the physical instrument. If Simulate is enabled or the instrument is not capable of reporting the model number or name, a string is returned that explains the condition. InstrumentModel is limited to 256 bytes.

### Hierarchy

```
IAgilentU2701A
    └ Identity
       └ InstrumentModel
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' InstrumentModel Statement
    Dim InstrumentModel As String
    InstrumentModel = Driver.Identity.InstrumentModel
End Sub
```

# Revision

### Type

Property

### Function

Get

### Description

This command returns the revision of the implementing component. Revision is limited to 256 bytes.

### Hierarchy

```
IAgilentU2701A
    └ Identity
      └ Revision
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Revision Statement
    Dim Revision As String
    Revision = Driver.Identity.Revision
End Sub
```

# SpecificationMajorVersion

### Type

Property

### Function

Get

### Description

For IVI class-compliant drivers, this command returns the major version number of the instrument class specification. If the driver is not class compliant, the driver returns zero.

### Hierarchy

```
IAgilentU2701A
    └ Identity
       └ SpecificationMajorVersion
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SpecificationMajorVersion Statement
    Dim SpecificationMajorVersion As Int32
    SpecificationMajorVersion =
     Driver.Identity.SpecificationMajorVersion
End Sub
```

# SpecificationMinorVersion

### Type

Property

### Function

Get

### Description

For IVI class-compliant drivers, this command returns the minor version number of the instrument class specification. If the driver is not class compliant, the driver returns zero.

### Hierarchy

```
IAgilentU2701A
    └ Identity
       └ SpecificationMinorVersion
```

### Return Format

Long/Int32

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SpecificationMinorVersion Statement
    Dim SpecificationMinorVersion As Int32
    SpecificationMinorVersion =
     Driver.Identity.SpecificationMinorVersion
End Sub
```

# SupportedInstrumentModels

### Type

Property

### Function

Get

### Description

This command returns a comma-separated list of instrument models that the IVI specific driver can control. The string does not include an abbreviation for the manufacturer if it is the same for all models.

### Hierarchy

```
AgilentU2701A
    └ Identity
       └ SupportedInstrumentModels
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SupportedInstrumentModels Statement
    Dim SupportedInstrumentModels As String
    SupportedInstrumentModels =
     Driver.Identity.SupportedInstrumentModels
End Sub
```

# Vendor

### Type

Property

### Function

Get

### Description

This command returns the name of the vendor that supplies the implementing component. Vendor is limited to 256 bytes.

### Hierarchy

```
IAgilentU2701A
    └ Identity
       └ Vendor
```

### Return Format

String/BSTR

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' Vendor Statement
    Dim Vendor As String
    Vendor = Driver.Identity.Vendor
End Sub
```

# SerialNumber

### Type

Property

### Function

Get

### Description

This command returns the instrument serial number.

### Hierarchy

```
IAgilentU2701A
    └ System
       └ SerialNumber
```

### Return Format

String

### Example

```
Sub Main()
    ' Create an instance of the driver
    Dim Driver As New AgilentU2701A
    ' SerialNumber Statement
    Dim SerialNumber As String
    SerialNumber = Driver.System.SerialNumber
End Sub
```

# 7 Application Example



This chapter contains an example program for the U2701A/U2702A USB modular oscilloscopes. The following program uses Microsoft Windows and Microsoft Visual Basic .Net (2003 and later). Refer to the instruction manual of Visual Basic .Net for further details about Visual Basic .Net.

# Introduction

This section describes how the AgilentU2701A IVI driver is used for Visual Basic .Net (2003 and later). All explanations will be using Console Application as the project.

## Referencing the Driver

In order to access AgilentU2701A IVI driver interface, a reference to the driver DLL must be created by following the steps listed below:

1. In the Solution Explorer, right-click your project name and select **Add Reference**.
2. Click the **COM** tab.
3. Select the **IVI AgilentU2701A 1.0 Type Library** and click **OK**.
4. The IVI AgilentU2701A 1.0 Type Library should now appear under the Reference tab by right-clicking your project name and selecting Properties.

All data types (interfaces and enums) are located under namespaces. Usually namespace qualified name must be used, but the “Imports” statement allows the type name to be used directly.

```
Imports System

Imports System.Runtime.InteropServices

Imports Ivi.Driver

Imports Ivi.Scope

Imports Agilent.AgilentU2701A.Interop
```

## Creating an Instance

Create an instance of the Agilent U2701A IVI-COM driver.

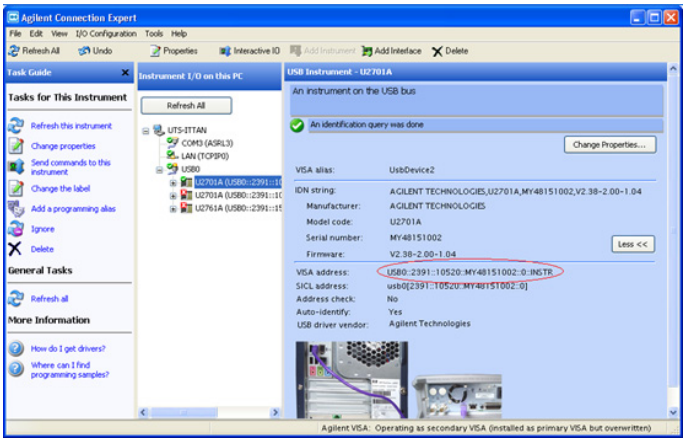
```
' Create an instance of the driver

Dim Driver As New AgilentU2701A
```

## Initializing the Driver

Calling Initialize will establish a connection to Agilent U2701A. You can get your resource name of your instrument by using the Agilent Connection Expert. Please refer to the "Initialize" on page 29 for a more detailed explanation.

This is an example to show you how to get your module resource name by using Agilent Connection Expert.




```
' Initialize the Driver

Driver.Initialize("USB0::2391::10520::MY48151002::0::INSTR",
True, True, "")
```

Calling Close is required to release the driver object.

```
' Close the Driver Object

Driver.Close()
```

# Complete Example

This section illustrates a complete example covering the steps needed to fetch the instrument identity, reset the instrument, automatically scale the instrument, conducting a simple measurement, and finally acquiring and saving the waveform data. All the steps taken are commented in this example program.

```
Imports System
Imports System.Runtime.InteropServices
Imports System.IO
Imports Ivi.Driver
Imports Ivi.Scope
Imports Agilent.AgilentU2701A.Interop

Module Module1

    Sub Main()

       Try

          ' Create the instance of the driver
          Dim Driver As New AgilentU2701A

          ' Initialize the Driver
          Driver.Initialize("USB0::2391::10520::MY48151002::0::INSTR", False,
           False, "")

          ' Get Instrument Identity property
          Dim InstrumentModel As String
          InstrumentModel = Driver.Identity.InstrumentModel
          Console.WriteLine("Instrument Model: {0}", InstrumentModel)

          Dim InstrumentManufacturer As String
          InstrumentManufacturer = Driver.Identity.InstrumentManufacturer
          Console.WriteLine("Instrument Manufacturer: {0}",
           InstrumentManufacturer)
```

```
Dim Revision As String
Revision = Driver.Identity.Revision
Console.WriteLine("Revision: {0}", Revision)

' Set TimeoutMilliseconds = 30000 (30 seconds)
Driver.System.TimeoutMilliseconds = 30000
' Reset U2701A
Driver.Utility.Reset()
Console.WriteLine("Reset Instrument Done")

' Auto Scale U2701A
Driver.Measurements.AutoSetup()
Console.WriteLine("Auto Scale Done")

' Measure Frequency signal input from Channel 1
Dim CH1Name As String = Driver.Channels.Name(1)
Dim FrequencyMeasured As Double

Driver.Measurements.Item(CH1Name).ReadWaveformMeasurement(AgilentU2701AM
  easurementEnum.AgilentU2701AMeasurementFrequency, 5000,
  FrequencyMeasured)
Console.WriteLine("Frequency Measured: {0} Hz", FrequencyMeasured)

' Waveform Acquisition
Dim RecordLength As Int32 = Driver.Acquisition.RecordLength
Dim WaveformArray() As Double
Dim XIncrement As Double
Dim InitialX As Double
Dim XValue As Double = 0
Dim waveformText As String = ""
ReDim WaveformArray(RecordLength - 1)

Driver.Measurements.Item(CH1Name).ReadWaveform(5000, WaveformArray,
  InitialX, XIncrement)
Console.WriteLine("Initial X Value: {0}", InitialX)
Console.WriteLine("X Increment Value: {0}", XIncrement)
Console.WriteLine("WaveformArray: ")
```

```
    Dim i As Int32 = 0
    While i < RecordLength
       XValue = InitialX + i*XIncrement
       Console.Write("{0}, {1}" + vbCrLf, XValue, WaveformArray(i))
       waveformText += XValue.ToString("F6") + vbTab +
         WaveformArray(i).ToString("F4") + vbCrLf
       i = i + 1
    End While
    ' Create a writer and open the file
    Dim tw As TextWriter
    tw = New StreamWriter("C:\waveform.txt")

    ' Write the text stream to file
    tw.Write(waveformText)

    ' Close the stream
    tw.Close()

    ' Read instrument error queue until its empty
    Dim ErrorCode = -1
    Dim ErrorMessage = ""

    Console.WriteLine()
    While ErrorCode <> 0
       Driver.Utility.ErrorQuery(ErrorCode, ErrorMessage)
       Console.WriteLine("ErrorQuery: {0}, {1}", ErrorCode, ErrorMessage)
    End While

    ' Close the Driver Object
    Driver.Close()
    Catch ex As Exception
    Console.WriteLine(ex.Message)

End Try

Console.WriteLine("Done - Press Enter to Exit")
Console.ReadLine()
```

```
    End Sub

End Module
```

# 7 Application Example

# Command Quick Reference

| Root keyword | Second-level | Third-level | Fourth-level | Page |
|---|---|---|---|---|
| IAgilentU2701A | Acquisition | ConfigureRecord | | See page 100 |
| | | Interpolation | | See page 102 |
| | | NumberOfAverages | | See page 104 |
| | | RecordLength | | See page 105 |
| | | SampleMode | | See page 106 |
| | | SampleRate | | See page 107 |
| | | StartTime | | See page 108 |
| | | TimePerRecord | | See page 109 |
| | | Type | | See page 112 |
| | Calibration | Calibrate | | See page 156 |
| | | Date | | See page 157 |
| | | Label | | See page 158 |
| | | Time | | See page 159 |
| | Channels | Count | | See page 12 |
| | | Item | BandwidthLimit | See page 13 |
| | | | Configure | See page 15 |
| | | | Coupling | See page 17 |
| | | | Enabled | See page 19 |
| | | | Offset | See page 21 |
| | | | ProbeAttenuation | See page 23 |
| | | | Range | See page 25 |
| | | Name | | See page 27 |
| | Close | | | See page 28 |

| Root keyword | Second-level | Third-level | Fourth-level | Page |
|---|---|---|---|---|
| IAgilentU2701A | DriverOperation | Cache | | See page 160 |
| | | ClearInterchange Warnings | | See page 161 |
| | | DriverSetup | | See page 162 |
| | | GetNextCoercion Record | | See page 163 |
| | | GetNextInterchange Warning | | See page 164 |
| | | InterchangeCheck | | See page 165 |
| | | InvalidateAllAttributes | | See page 166 |
| | | IoResourceDescriptor | | See page 167 |
| | | LogicalName | | See page 168 |
| | | QueryInstrumentStatus | | See page 169 |
| | | RangeCheck | | See page 170 |
| | | RecordCoercions | | See page 171 |
| | | ResetInterchange Check | | See page 172 |
| | | Simulate | | See page 173 |
| | Identity | Description | | See page 174 |
| | | GroupCapabilities | | See page 175 |
| | | Identifier | | See page 176 |
| | | InstrumentFirmware Revision | | See page 177 |
| | | Instrument Manufacturer | | See page 178 |
| | | InstrumentModel | | See page 179 |
| | | Revision | | See page 180 |
| | | SpecificationMajor Version | | See page 181 |

| Root keyword | Second-level | Third-level | Fourth-level | Page |
|---|---|---|---|---|
| IAgilentU2701A | Identity | SpecificationMinor Version | | See page 182 |
| | | SupportedInstrument Models | | See page 183 |
| | | Vendor | | See page 184 |
| | Initialize | | | See page 29 |
| | Initialized | | | See page 31 |
| | Measurements | Abort | | See page 118/130 |
| | | AutoSetup | | See page 119/131 |
| | | Count | | See page 120/132 |
| | | Initiate | | See page 121/133 |
| | | IsWaveformElement Invalid | | See page 122/134 |
| | | Item | FetchWaveform | See page 124/136 |
| | | | FetchWaveform Measurement | See page 138 |
| | | | ReadWaveform | See page 126/142 |
| | | | ReadWaveform Measurement | See page 144 |
| | | MathFunction | | See page 146 |
| | | Name | | See page 147 |
| | | Status | | See page 148 |
| | Status | Clear | | See page 68 |
| | | ConfigureService Request | | See page 69 |
| | | Preset | | See page 71 |
| | | Register | | See page 72 |
| | | SerialPoll | | See page 74 |

| Root keyword | Second-level | Third-level | Fourth-level | Page |
|---|---|---|---|---|
| IAgilentU2701A | System | SerialNumber | | See page 75/185 |
| | | TimeoutMilliseconds | | See page 75 |
| | | WaitForOperation Complete | | See page 77 |
| | Trigger | Configure | | See page 32 |
| | | Coupling | | See page 33 |
| | | Edge | Configure | See page 35 |
| | | | Slope | See page 37 |
| | | Glitch | Condition | See page 39 |
| | | | Configure | See page 41 |
| | | | Polarity | See page 43 |
| | | | Width | See page 45 |
| | | Holdoff | | See page 46 |
| | | Modifier | | See page 47 |
| | | Level | | See page 49 |
| | | Source | | See page 50 |
| | | Status | | See page 51 |
| | | TV | Configure | See page 52 |
| | | | Event | See page 54 |
| | | | LineNumber | See page 56 |
| | | | SignalFormat | See page 57 |
| | | | Type | See page 59 |
| | | Width | Condition | See page 60 |
| | | | Configure | See page 62 |
| | | | Polarity | See page 64 |
| | | | ThresholdHigh | See page 66 |

| Root keyword | Second-level | Third-level | Fourth-level | Page |
|---|---|---|---|---|
| IAgilentU2701A | Trigger | Width | ThresholdLow | See page 67 |
| | Utility | Disable | | See page 78 |
| | | ErrorQuery | | See page 79 |
| | | LockObject | | See page 80 |
| | | Reset | | See page 81 |
| | | ResetWithDefaults | | See page 82 |
| | | SelfTest | | See page 83 |
| | | UnlockObject | | See page 85 |

# Enumeration Quick Reference

| Enumeration members | Page |
|---|---|
| AgilentU2701AAcquisitionStatusEnum | See page 114 |
| AgilentU2701AAcquisitionTypeEnum | See page 115 |
| AgilentU2701AGlitchConditionEnum | See page 86 |
| AgilentU2701AGlitchPolarityEnum | See page 87 |
| AgilentU2701AInterpolationEnum | See page 116 |
| AgilentU2701AMathOperationEnum | See page 149 |
| AgilentU2701AMeasurementEnum | See page 150 |
| AgilentU2701ASampleModeEnum | See page 116 |
| AgilentU2701ASRQReasonEnum | See page 87 |
| AgilentU2701AStatusRegisterEnum | See page 89 |
| AgilentU2701AStatusSubRegisterEnum | See page 90 |
| AgilentU2701ATimeOutEnum | See page 153 |
| AgilentU2701ATriggerCouplingEnum | See page 91 |
| AgilentU2701ATriggerModifierEnum | See page 91 |
| AgilentU2701ATriggerSlopeEnum | See page 92 |
| AgilentU2701ATriggerTypeEnum | See page 93 |
| AgilentU2701ATVSignalFormatEnum | See page 94 |
| AgilentU2701ATVTriggerEventEnum | See page 95 |
| AgilentU2701AVerticalCouplingEnum | See page 96 |
| AgilentU2701AWidthConditionEnum | See page 97 |
| AgilentU2701AWidthPolarityEnum | See page 97 |

# Appendix

| Property | Possible values | | |
|---|---|---|---|
| IAgilentU2701AAcquisition.NumberOfAverages | Values (Int32) | | |
| | 1 - 999 | | |
| IAgilentU2701AAcquisition.StartTime<br>IAgilentU2701AAcquisition.TimePerRecord | Time/Div | TimePerRecord | StartTime |
| | 1 ns | 10 ns | –5 ns - 5 ns |
| | 2 ns | 20 ns | –10 ns - 10 ns |
| | 5 ns | 50 ns | –25 ns - 25 ns |
| | 10 ns | 100 ns | –50 ns - 50 ns |
| | 20 ns | 200 ns | –100 ns - 100 ns |
| | 50 ns | 500 ns | –250 ns - 250 ns |
| | 100 ns | 1 µs | –500 ns - 500 ns |
| | 200 ns | 2 µs | –1 µs - 1 µs |
| | 500 ns | 5 µs | –2.5 µs - 2.5 µs |
| | 1 µs | 10 µs | –5 µs - 5 µs |
| | 2 µs | 20 µs | –10 µs - 10 µs |
| | 5 µs | 50 µs | –25 µs - 25 µs |
| | 10 µs | 100 µs | –50 µs - 50 µs |
| | 20 µs | 200 µs | –100 µs - 100 µs |
| | 50 µs | 500 µs | –250 µs - 250 µs |
| | 100 µs | 1 ms | –500 µs - 500 µs |
| | 200 µs | 2 ms | –1 ms - 1 ms |
| | 500 µs | 5 ms | –2.5 ms - 2.5 ms |
| | 1 ms | 10 ms | –5 ms - 5 ms |
| | 2 ms | 20 ms | –10 ms - 10 ms |
| | 5 ms | 50 ms | –25 ms - 25 ms |
| | 10 ms | 100 ms | –50 ms - 50 ms |
| | 20 ms | 200 ms | –100 ms - 100 ms |
| | 50 ms | 500 ms | –250 ms - 250 ms |
| | 100 ms | 1 s | –500 ms - 500 ms |

| | | | |
|---|---|---|---|
| **IAgilentU2701AAcquisition.StartTime<br>IAgilentU2701AAcquisition.TimePerRecord** | **Time/Div** | **TimePerRecord** | **StartTime** |
| | 200 ms | 2 s | –1 s - 1 s |
| | 500 ms | 5 s | –2.5 s - 2.5 s |
| | 1 s | 10 s | –5 s - 5 s |
| | 2 s | 20 s | –10 s - 10 s |
| | 5 s | 50 s | –25 s - 25 s |
| | 10 s | 100 s | –50 s - 50 s |
| | 20 s | 200 s | –100 s - 100 s |
| | 50 s | 500 s | –250 s - 250 s |
| **IAgilentU2701AChannels.Item** | **Name (String)** | | |
| | Channel1 | | |
| | Channel2 | | |
| | MathChannel | | |
| **IAgilentU2701AChannel.ProbeAttenuation** | **Value (double)** | | |
| | 1 | | |
| | 10 | | |
| | 100 | | |
| **IAgilentU2701AChannel.Offset<br>IAgilentU2701AChannel.Range** | **Volt/Div** | **Voltage Range (Attenuation = 1x)** | **Offset** |
| | 2 mV | 16 mV | –8 mV - 8 mV |
| | 5 mV | 40 mV | –20 mV - 20 mV |
| | 10 mV | 80 mV | –40 mvV - 40 mV |
| | 20 mV | 160 mV | –80 mV - 80 mV |
| | 50 mV | 400 mV | –200 mV - 200 mV |
| | 100 mV | 800 mV | –400 mV - 400 mV |
| | 200 mV | 1.6 V | –800 mV - 800 mV |
| | 500 mV | 4 V | –2 V - 2 V |
| | 1 V | 8 V | –4 V - 4 V |
| | 2 V | 16 V | –8 V - 8 V |
| | 5 V | 40 V | –20 V - 20 V |
| **IAgilentU2701ATriggerGlitch.Width** | **Condition LessThan (Value)** | **Condition GreaterThan (Value)** | |
| | > 7 ns | > 15 ns | |

| | |
|---|---|
| **IAgilentU2701ATrigger.Holdoff** | **Value (Double)** |
| | > 59 ns |
| **IAgilentU2701ATrigger.Source** | **Value (String)** |
| | Channel1 |
| | Chan1 |
| | CHANNEL1 |
| | Channel2 |
| | Chan2 |
| | CHANNEL2 |
| | Ext |
| | EXT |
| **IAgilentU2701ATriggerTv.LineNumber** | **Value (Int)** |
| | 1 - 263 |
| **IAgilentU2701ATriggerWidth.ThresholdHigh** | **Value (Double)** |
| | > 15 ns |
| **IAgilentU2701ATriggerWidth.ThresholdLow** | **Value (Double)** |
| | > 7 ns |

**www.agilent.com**

**Contact us**
To obtain service, warranty or technical assistance, contact us at the following phone or fax numbers:

United States:
(tel) 800 829 4444 (fax) 800 829 4433
Canada:
(tel) 877 894 4414 (fax) 800 746 4866
China:
(tel) 800 810 0189 (fax) 800 820 2816
Europe:
(tel) 31 20 547 2111
Japan:
(tel) (81) 426 56 7832 (fax) (81) 426 56 7840
Korea:
(tel) (080) 769 0800 (fax) (080) 769 0900
Latin America:
(tel) (305) 269 7500
Taiwan:
(tel) 0800 047 866 (fax) 0800 286 331
Other Asia Pacific Countries:
(tel) (65) 6375 8100 (fax) (65) 6755 0042

Or visit Agilent World Wide Web at:
www.agilent.com/find/assist

Product specifications and descriptions in this document are subject to change without notice. Always refer to Agilent Web site for the latest revision.



Third Edition, August 13, 2012
U2702-90035

**Agilent Technologies**