ADWG TCL Library

User's Guide

Byte Paradigm – info@byteparadigm.com

Table of Content

Table of Tables

Table of Pictures

References

[]

History

Version	Date	Description
1.00	31-Aug-2005	Initial revision
1.01	15-Sep-2005	Completed the TCL procedures description
1.02	01-Nov-2005	Reviewed
1.03	15-Feb-2006	Updated to comply with software version 1.03
1.04	22-May-2006	Some corrections
1.05	09-Nov-2006	Update for software revision 1.04
1.06	05-Mar-2007	Update with infinite loop / trigger auto rearm functions
1.07	31-May-2007	Added edge triggering functions
1.08	13-Nov-2007	Update for GP Series introduction
1.09	16-Feb-2010	Review for release 1.08f.
1.10	20-Jan-2012	Added IO voltage selection
1.11	13-Feb-2015	Minor corrections

1 Introduction

The objective of this document is to list and describe all the TCL procedures available in the ADWG library provided to control the ADWG operating mode of the GP Series device. Each procedure functionality and parameters are described in detail. Scripts examples, that use some of these procedures are also provided to help the users build their own test environments.

A section is also dedicated to the TCL interpreter provided with the 8PI Control Panel application. The way to start it and to use it is briefly described.

2 TCL Interpreter

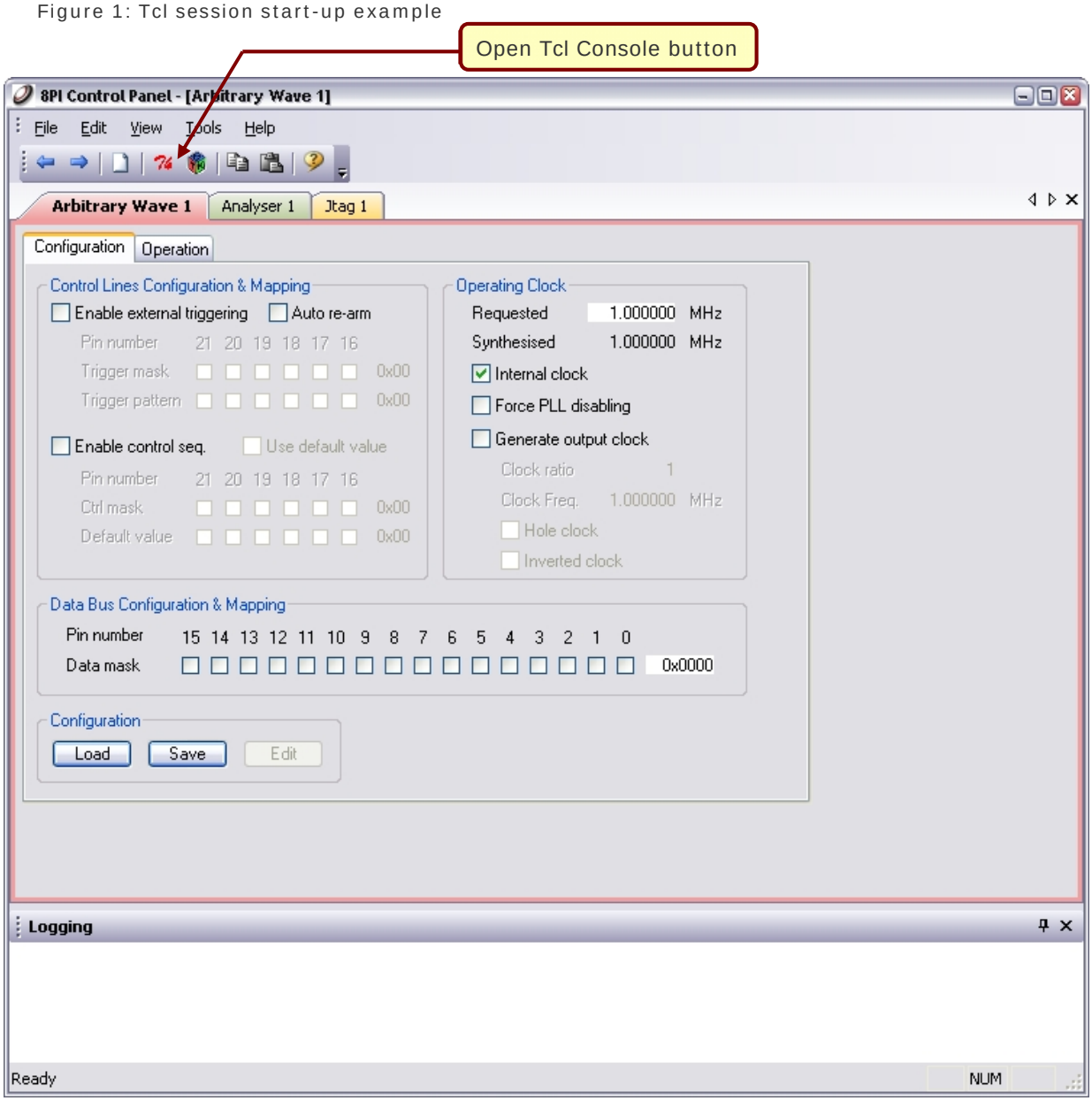
2.1 Starting a TCL Session

To start a TCL session from the 8PI Control Panel GUI:

1. From the 8PI Control Panel GUI, access the desired operating mode sheet.
2. Click on the 'Open Tcl Console' button (Figure 1).

This opens the Tcl console, loads the Tcl libraries relative to the chosen operating mode and initialises the Tcl session.

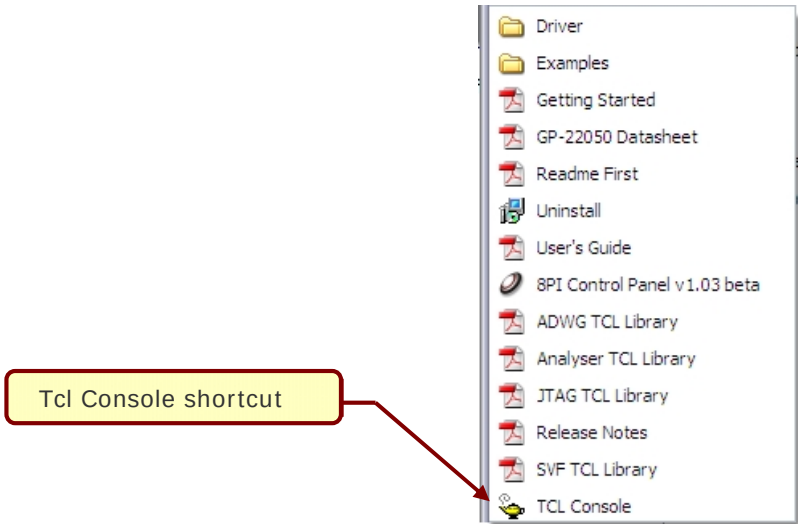
Figure 1: Tcl session start-up example



To start a stand-alone TCL session (without running GUI):

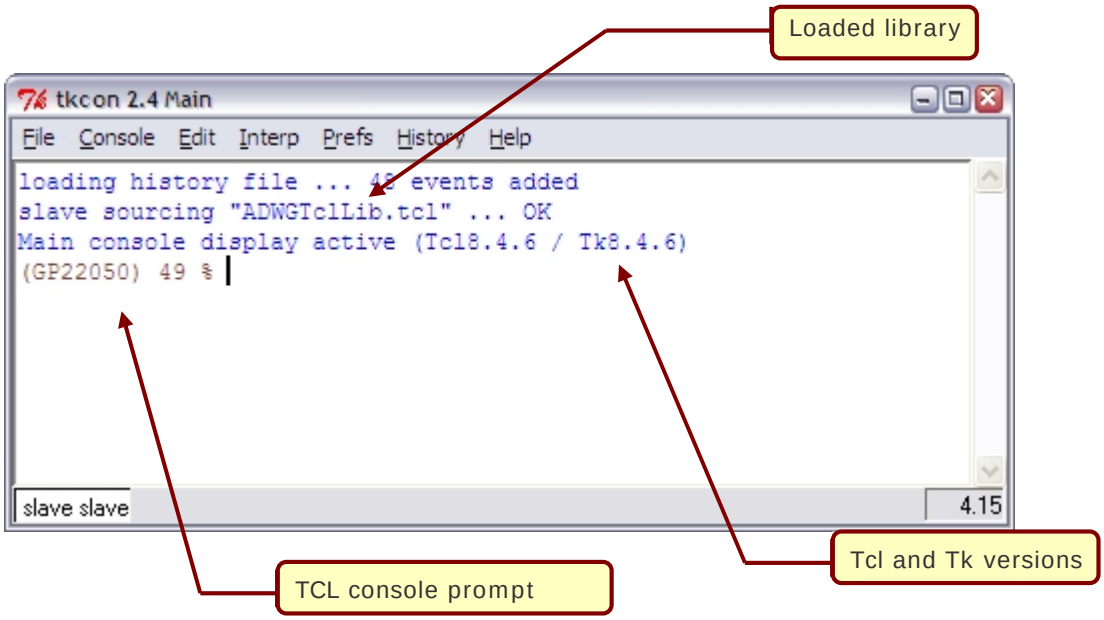
1. From the 'Byte Paradigm > 8PI Control Panel' program group, click on the 'TCL Console' shortcut. This starts the Wish84 interpreter with the tkcon console.
2. In the Tcl console, type: `% source ADWGTclLib.tcl`
   This initialises your TCL session in ADWG operating mode.

Figure 2: TCL console shortcut in 8PI Control Panel program group



By default, the GP Series device software environment uses the WISH interpreter with the TKCON console (interactive mode) (Figure 3). For more information about the TKCON console, please check the following links: http://tkcon.sourceforge.net/ - http://wiki.tcl.tk/1878.
Please note that TCL is case sensitive.

Figure 3: Tcl console at startup



2.2 Getting Information on TCL Procedures

Tcl provides built-in commands for getting information about the elements loaded in memory during a Tcl session. We simply describe a few of them for those unfamiliar to the Tcl language.

To list the libraries loaded in the TCL environment:

```
% info loaded
```

To list all the procedures loaded in the TCL environment:

```
% info procs
```

To list the arguments of a given procedure:

```
% info args <procedure name>
```

To list the body of a given procedure:

```
% info body <procedure name>
```

To learn more about the TCL/TK language, numerous man pages, tutorials and references can be found at the following location: http://www.tcl.tk/doc/ .

3 Arbitrary Waveform Library - ADWGTclLib

This library contains all the procedures available to control the GP Series device when operating in arbitrary digital waveform generator mode (ADWG mode). Using these procedures in scripts or command lines allows the user to send out sequences of samples to the system connector of the GP Series device.

3.1 Quick Reference Table

Table 1 gives a list of all procedures available for the ADWG mode. They are grouped by type and functionality.

Table 1: Quick reference table of ADWG procedures (by functionality)

Procedures	Description
Clock Configuration	
UseIntClock {}	Configures the device to operate with the internal clock as reference.
UseExtClock {}	Configures the device to operate with an externally supplied clock as reference.
GetClockSource {{Display 1}}	Returns the selected clock source configured in the device.
SetReqClock {Freq}	Defines the requested operating clock frequency (in Hz).
GetReqClock {}	Returns the requested operating clock frequency (Hz)
GetSynthClock {}	Returns the synthesised operating clock frequency (Hz). It can differ from the requested frequency due to the limited accuracy of the device frequency divider.
SetClockDisablePLL {Disable {Display 1}}	Forces disabling the PLL or let the device enable it automatically when possible.
GetClockDisablePLL {{Display 0}}	Gets the operating mode of the PLL (disabled/automatic)
SetOutputClock {Output {Display 1}}	Enables or disables the generation of an output clock
GetOutputClock {}	Gets the status of the output clock generation (enabled or disabled)
SetOutClockRatio {Ratio}	Defines the integer output clock ratio with respect to the operating clock.
GetOutClockRatio {}	Returns the current value of the output clock ratio.
GetSynthOutClock {}	Returns the achieved frequency (Hz) of the output clock. This value depends on the operating clock frequency and the output clock ratio.
SetHoleClock {}	Enables the "hole" mode to generate the output clock. In this mode, a clock pulse is generated only when a data is sent out. When no data is sent out, the clock remains low.
SetContinuousClock {}	Sets the output clock generation in continuous mode. In this mode, a permanent and continuous clock is generated independently of the data stream.
GetClockContinuity {{Display 0}}	Returns the current continuity mode used to generate the output clock (hole or continuous)
SetClockEdge {Pos {Display 1}}	Defines the phase relation between the output clock and the device internal clock.
GetClockEdge {}	Returns the phase relation between the device internal clock and the generated output clock.
Triggering Configuration	
SetInternalTrigger {Internal {Display 1}}	Defines the triggering mode (internal or external)
GetInternalTrigger {{Display 0}}	Returns the current triggering mode
SetEdgeTrigger {Enable {Display 1}}	Selects the triggering mode (edge or level)

Procedures	Description
GetEdgeTrigger {{Display 0}}	Returns the triggering mode (edge or level)
SetCtrlTrigMask {Mask {Display 1}}	Defines the triggering mask to select the control lines to use as trigger inputs.
GetCtrlTrigMask {}	Returns the triggering mask
SetCtrlTrigPattern {Pattern {Display 1}}	Defines the pattern to detect on the trigger inputs to generate the trigger event.
GetCtrlTrigPattern {}	Returns the triggering pattern.
SetAutoReArm {AutoReArm}	Enables or disables the trigger 'auto rearm' feature.
GetAutoReArm {{Display 0}}	Return the auto-rearm state.
Control Lines Sequence Configuration	
EnCtrlSeq {}	Enables the generation of static or periodic sequence on the control lines.
DisCtrlSeq {}	Disables the generation of static and periodic sequence on the control lines.
GetCtrlSeqEn {}	Returns the current status for the control sequence generation.
GetCtrlSeqLength {}	Returns the current length of the periodic sequence to apply on the control lines.
EnDefaultCtrl {{Display 0}}	Enables applying default static levels on the control lines.
DisDefaultCtrl {}	Disables applying a forced default level on the control lines.
GetDefaultCtrlEn {{Display 0}}	Returns the status of the default level feature.
SetCtrlMaskOut {MaskOut {Display 1}}	The control mask selects the control lines on which the default static levels or the periodic pattern have to be applied.
GetCtrlMaskOut {}	Returns the control mask value.
SetCtrlDefaultVal {Val {Display 1}}	Defines the default level to apply on the control lines.
GetCtrlDefaultVal {}	Returns the current default level applied on the control lines.
Data Lines Configuration	
SetDataMaskOut {MaskOut {Display 1}}	Selects the data lines to enable as outputs.
GetDataMaskOut {}	Returns the data mask value.
Static Mode Operations	
SetStaticData {Data {Display 1}}	Applies the data pattern to the device system connector pins
GetStaticData {}	Returns last static data value sent to the device.
File Mode Operations	
GetNrSamples {}	Returns the total number of data samples loaded in memory.
AdwgRun {SendSamples {Display 1} {PopUp 0}}	Starts transferring data to the GP Series device.
AdwgRunLoop {SendSamples {Display 1} {PopUp 0}}	Starts transferring data to the GP Series device. This function must be used when the loop features.
Abort {}	Aborts the ADWG running in loop / trigger auto rearm modes.
SetInfiniteLoop {Enable}	Enables / disables the infinite loop mode.
GetInfiniteLoop {}	Returns the enabled/disabled status of the infinite loop mode.
Unsupervised {Enable {Display 1}}	Enables or disables the unsupervised operating mode.
SetAdwgContMode {Continuous {Display 1}}	Enables or disables the continuous operating mode.
GetAdwgContMode {}	Returns the status of the continuous operating mode.
ClearBuffer {}	Clears the data buffer loaded in memory.
WriteSingleSample {Sample {Display 0}}	Appends the memory data buffer with the data sampe specified as argument.
File Management	
ReadConfFile {FileName {Display 0} {PopUp 0}}	Reads the device controls and configuration from a file.
WriteConfFile {FileName}	Writes the current configuration to a file.
MakeConfFileTemplate {FileName}	Creates a configuration file template.
ReadConfAndDataFile {FileName {Display 0} {PopUp 0}}	Reads the device controls, configuration, data header and the data samples to transfer from a file.
WriteConfAndDataFile {FileName}	Writes the device configuration and the data samples to a file.
ReadDataFile{FileName {Display 0} {PopUp 0}}	Reads the data header configuration and a set of data to be sent from the specified file.

Procedures	Description
	Others
SelectIOVoltage {IOVoltage}	Selects the user interface voltage.
GetLastError {}	Returns the last error detected during the data transfer.
ResetCfg {}	Reset the device to its default configuration.
InitVars {}	Initialises local variables to a default value.
ADWGConfig {}	Displays the ADWG session configuration.
Ver {}	Displays the software version.
IsDeviceReady {}	Checks if the device is properly connected to the host PC.
Terminate {}	Closes the ADWG TCL session.

3.2 Procedures Detailed Description

This section gives a detailed description of each procedure available to control the GP Series device arbitrary waveform operating mode. The procedures are listed in alphabetical order.

ADWGConfig { }

parameters : none
returns:
description: Displays the ADWG session configuration.
conditions: None
see also:

AdwgRun {SendSamples {Display 1} {PopUp 0}}

parameters : SendSamples: integer value that specifies the number of samples to transfer for the device.
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.
PopUp: Optional parameter. Enables the pop-up messages when set to 1. By default set to 0.
returns: An integer error code ≥0 if successful.
description: Starts transferring data to the GP Series device. The specified number of samples are applied to the system connector of the device. Only the pins enabled and configured as output are driven, the other ones remain in high impedance state.
conditions: If the device is not configured to operate in ADWG file mode, this procedure automatically configures it with the current defined settings.

see also:

AdwgRunLoop {SendSamples {Display 1} {PopUp 0}}

parameters : SendSamples: integer value that specifies the number of samples to transfer for the device.
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.
PopUp: Optional parameter. Enables the pop-up messages when set to 1. By default set to 0.
returns: An integer error code ≥0 if successful.
description: Starts transferring data to the GP Series device. The specified number of samples is applied onto the device system connector. Only the pins

enabled and configured as output are driven; the other ones remain in high impedance state. This function must be used to start runs with the infinite loop and the trigger auto rearm features.

conditions:

see also:

Abort { }

parameters: None

returns:

description: Aborts the ADWG running in loop / trigger auto rearm modes.

conditions:

see also:

ClearBuffer { }

parameters: None

returns:

description: Clears the data buffer loaded in memory. Before being sent to the device, the data samples are stored in a memory area. This function allows to completely clear the area of the memory where the samples are buffered.

conditions:

see also: WriteSingleSample{ }

DisCtrlSeq { }

parameters: None

returns:

description: Disables the generation of static and periodic sequences on the control lines.

conditions:

see also: EnCtrlseq{ }, EnDefaultCtrl{ }, DisDefaultCtrl{ }

DisDefaultCtrl { }

parameters: None

returns:

description: Disables applying a forced default level on the control lines

conditions:

see also: EnDefaultCtrl{ }, EnCtrlseq{ }, DisCtrlseq{ }

EnCtrlSeq { }

parameters: None

returns:

description: Enables the generation of the static or periodic sequences on the control lines.

conditions:

see also: DisCtrlSeq{ }, EnDefaultCtrl{ }, DisDefaultCtrl{ }

EnDefaultCtrl {{Display 0}}

parameters:	Display:	Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 0.
returns:		
description:	Enables applying default static levels on the control lines. When this feature is enabled, static levels can be set on the selected control lines. Valid for File or Static modes.	
conditions:	To enable the default level feature for the control lines, the control sequence feature has to be enabled. Using EnDefaultCtrl{} automatically calls EnCtrlSeq{} if needed. If Display is set to 1, a message is displayed to report the control sequence enabling.	
see also:	DisDefaultCtrl{}, EnCtrlseq{}, DisCtrlseq{}, SetDefaultVal{}	

GetAdwgContMode {}

parameters:	None
returns:	1 if continuous mode is enabled 0 if continuous mode is disabled
description:	Returns the status of the continuous operating mode.
conditions:	The continuous mode is only defined for File mode.
see also:	SetAdwgContMode{}

GetAutoReArm {}

parameters:	Display: optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0 – by default set to 0.
returns:	0 if the trigger auto rearm feature is disabled. 1 if the trigger auto rearm feature is enabled.
description:	Returns the status of the 'trigger auto rearm' feature.
conditions:	Trigger auto rearm only works with an external trigger.
see also:	SetAutoReArm{}, AdwgRunLoop{}

GetClockContinuity {{Display 0}}

parameters:	Display:	Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 0.
returns:	1 if the Hole clock mode is enabled 0 if the Continuous clock mode is enabled If the Display parameter is set to 1, the following text messages are displayed depending on the clock continuity status. 'HOLE' clock 'CONTINUOUS' clock	
description:	Returns the output clock operating mode.	
conditions:	None	
see also:	SetHoleClock{}, SetContinuousClock{}	

GetClockDisablePLL {{Display 0}}

parameters:	Display: optional parameter. Displays additional information when set to 1. By default set to 0.
returns:	1 when the internal device PLL is disabled 0 when the automatic PLL control mode is enabled
description:	Returns the control mode of the device internal PLL.

conditions:
see also: SetClockDisablePLL{}

GetClockEdge {}

parameters: None

returns:
1 Output data lines are generated in phase with the rising edge of the output clock.
0 Output data lines are generated in phase with the falling edge of the output clock

description: Returns the phase relation between the device internal clock and the generated output clock.

conditions:
see also: SetClockEdge{}

GetClockSource {{Display 1}}

parameters: Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

returns:
1 Internal reference clock is used
0 External reference clock is used
When Display is set to 1, and additional text message is returned depending on the selected clock source.
Source: INTERNAL clock
Source: EXTERNAL clock

description: Returns the reference clock source used to generate the device operating clock.

conditions:
see also: UseIntClock{}, UseExtClock{}

GetCtrlDefaultVal {}

parameters: None

returns: A hexadecimal value prefixed with 0x

description: Returns the current default level pattern defined for the 6 control lines. The LSB of the value corresponds to control line 0 and the MSB corresponds to control line 5.

conditions:
see also: SetCtrlDefaultVal{}, EnDefaultCtrl{}

GetCtrlMaskOut {}

parameters: None

returns: A hexadecimal value prefixed with 0x

description: Returns a mask value representing the control lines enabled as output. The LSB of the value corresponds to control line 0 and the MSB corresponds to control line 5. When a bit is set to 1, the corresponding control line is configured as an output. When a bit is set to 0, the corresponding pin remains in high impedance state and can be used as an input for external triggering.

conditions:
see also: SetCtrlMaskOut{}, SetCtrlTrigMask{}

GetCtrlSeqEn {}

parameters:	None
returns:	1 when the control sequence feature is enabled 0 when the control sequence feature is disabled
description:	Returns the status of the 'enable control sequence' feature.
conditions:	
see also:	EnCtrlSeq{}, EnCtrlDefault{}

GetCtrlSeqLength {}

parameters:	None
returns:	A decimal value
description:	Returns the current length of the defined control sequence.
conditions:	
see also:	ReadConfAndDataFile{}

GetCtrlTrigMask {}

parameters:	None
returns:	A hexadecimal value prefixed with 0x
description:	Returns the mask applied on the control lines to detect the external trigger pattern.
conditions:	SetCtrlTrigMask{}, SetCtrltrigPattern{}, SetInternalTrigger{}

GetCtrlTrigPattern {}

parameters:	None
returns:	A hexadecimal value prefixed with 0x
description:	Returns the pattern to detect on the selected control lines to generate a trigger event to start applying data samples on the GP Series device system connector.
conditions:	
see also:	SetCtrlTrigPattern{}, SetCtrlTrigMask{}, SetInternalTrigger{}

GetDataMaskOut {}

parameters:	None
returns:	A hexadecimal value prefixed with 0x
description:	Returns a mask value representing the data lines enabled as output. The LSB of the value corresponds to data line 0 and the MSB corresponds to data line 15. When a bit is set to 1, the corresponding data line is configured as an output. When a bit is set to 0, the corresponding pin remains in high impedance state.
conditions:	
see also:	SetDataMaskOut{}

GetDefaultCtrlEn {{Display 0}}

parameters:	Display: optional parameter. Displays additional information in the TCL shell when set to 1. By default set to 0.
returns:	1 when the default level feature is enabled 0 when the default level feature is disabled
description:	Returns the status of the default level feature for the control lines.

conditions:
see also: EnDefaultCtrl{}, EnCtrlSeq{}

GetEdgeTrigger {{Display 0}}
parameters: Display: Optional parameter. Displays additional information messages when set to 1. By default set to 0.
returns:
description: Returns the current triggering mode: 1 for 'edge triggering mode'; 0 for 'level triggering mode'.

see also:

GetInternalTrigger {{Display 0}}
parameters: Display: optional parameter. Displays additional information in the TCL shell when set to 1. By default set to 0.
returns: 1 when internal triggering mode is enabled
0 when external triggering mode is enabled
description: Returns the triggering mode in use.
conditions:
see also: SetInternalTrigger{}, SetExternalTrigger{}

GetInfiniteLoop {}
parameters: None
returns: 1 when the 'infinite loop' mode is enabled
0 when the 'infinite loop' mode is disabled
description: Returns the enable/disable status of the 'infinite loop mode'.
conditions:
see also: SetInfiniteLoop{}

GetLastError {}
parameters: None
returns: A decimal value
description: Returns the last error code returned by the device.
1 : underflow error: the system available bandwidth is not sufficient. Please reduce the used clock frequency.
Any other value different from 0: system error. Please report error to Byte Paradigm.
conditions: This procedure can be used in unsupervised mode, after having used AdwgRun, to check if the last set of data was sent without error.
see also: Unsupervised{}, AdwgRun{}

GetNrSamples {}
parameters: None
returns: A decimal value
description: Returns the total number of data samples loaded in memory.
conditions:
see also: ReadConfAndDataFile{}

GetOutClockRatio {}

parameters:	None
returns:	A decimal value
description:	Returns the integer ratio used to generate the output clock.
conditions:	
see also:	SetOutClockRatio{}

GetOutputClock {}

parameters:	None
returns:	1 when the output clock is enabled 0 when the output clock is disabled
description:	Returns status of the output clock.
conditions:	
see also:	SetOutputClock{}, SetOutClockRatio{}

GetReqClock {}

parameters:	None
returns:	A decimal value
description:	Returns the requested frequency for the device operating clock (unit = Hz)
conditions:	
see also:	GetSynthClock{}, SetReqClock{}

GetStaticData {}

parameters:	None
returns:	A hexadecimal value prefixed with 0x
description:	Returns the last static data transferred to the device.
conditions:	
see also:	SetStaticData{}

GetSynthClock {}

parameters:	None
returns:	A decimal value
description:	Returns the achieved frequency for the system operating clock. As the operating clock is generated using an integer ratio to divide a reference clock, not all frequencies can be generated. This procedure returns the achieved frequency value. This value can differ from the requested clock frequency. If the requested frequency can not be achieved, the device uses the first achievable frequency smaller than the requested one.
conditions:	
see also:	SetReqClock{}, GetReqClock{}

GetSynthOutClock {}

parameters:	None
returns:	A decimal value
description:	Returns the achieved frequency of the output clock (unit = Hz).
see also:	SetOutClockRatio{}

IsDeviceReady {}

parameters:	None
returns:	1 if device is connected.
description:	Checks if the device is properly connected to the host PC.
see also:	

InitVars {}

parameters:	none
returns:	
description:	Initialises local variables to a default value.
conditions:	None
see also:	

MakeConfFileTemplate {FileName}

parameters:	FileName: specifies the procedure output file name (including path).
returns:	0 when file generation is complete
description:	Creates a standard template to be used as configuration file with the device.
conditions:	
see also:	

ReadConfAndDataFile {FileName {Display 0} {PopUp 0}}

parameters:	FileName: specifies the procedure input file name (including path). Display: optional parameter. When set to 1, it enables TCL shell information messages. By default set to 0 (disabled). PopUp: optional parameter. When set to 1, it enables the information and error pop up windows. By default set to 0 (disabled).
returns:	0 if read succesful; another value if read failed.
description:	Reads the device configuration, control sequence configuration, data header and a set of data to be sent from the specified file.
conditions:	
see also:	ReadConfFile{}, ReadDataFile{}

ReadConfFile {FileName {Display 0} {PopUp 0}}

parameters:	FileName: specifies the procedure input file name (including path) Display: optional parameter. When set to 1, it enables TCL shell information messages. By default set to 0 (disabled). PopUp: optional parameter. When set to 1, it enables the information and error pop up windows. By default set to 0 (disabled).
returns:	0 if read succesful; another value if read failed.
description:	Reads the device configuration and the control sequence configuration from the specified file. Skips the data header and any set of data defined in the input file.
conditions:	
see also:	ReadConfAndDataFile{}, ReadDataFile{}

ReadDataFile {FileName {Display 0} {PopUp 0}}

parameters:	FileName: specifies the procedure input file name (including path) Display: optional parameter. When set to 1, it enables TCL shell information messages. By default set to 0 (disabled). PopUp: optional parameter. When set to 1, it enables the information and error pop up windows. By default set to 0 (disabled).
returns:	0 if read succesful; another value if read failed.
description:	Reads the data header configuration and a set of data to be sent from the specified file. Skips the device and the control sequence configurations.
conditions:	
see also:	ReadConfAndDataFile{}, ReadConfFile{}

ResetCfg {}

parameters:	None
returns:	
description:	Resets the device to a default state.
conditions:	

SetAdwgContMode {Continuous {Display 1}}

parameters:	Continuous: Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.
returns:	
description:	Enables or disables the continuous operating mode. In continuous mode, the amount of data specified with the AdwgRun{} parameters should be transferred as a single continuous run. If the device fails to do so, an underrun condition is reported. In non-continuous mode, pause times are allowed inside the data run to provide enough data to the device.
conditions:	
see also:	AdwgRun{}

SetAutoReArm {AutoReArm}

parameters:	AutoReArm: boolean value (0 or 1)
Returns:	An integer error code ≥ 0 when successful.
description:	Enables (AutoReArm = 1) or disables (AutoReArm = 0) the trigger 'auto rearm' feature. When an external trigger is selected, the 'auto rearm' feature allows to automatically send the same ADWG run, loaded into the GP Series device embedded memory, upon the arrival of a new external trigger. Two operating modes exist: • GP Series device driven: the pattern fits un the 16Mbytes memory of the GP-20050, as such it is locally stored in the device and independent of the USB bandwidth • Software driver: the pattern is too large to fit in the GP Series device memory, in this case it is streamed from the host computer and the throughput is dependant of the USB bandwidth Not that the selection between both operating modes is done automatically based on the size of the pattern.
conditions:	The trigger 'auto rearm' feature is only available when an external trigger is used.
see also:	GetAutoReArm{}, AdwgRunLoop{}

SetClockDisablePLL {Disable {Display 1}}

parameters: Disable: valid values: 0 or 1. Specifies whether the PLL must be disabled (1) or left enabled when needed (0).
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

Returns:

Description: Forces the disabling of the device internal PLL used for the connector clock generation. For the generation of the connector clock, a PLL embedded in the device can be used. When the Disable parameter is set to 0, the PLL will be automatically enabled if possible. When Disable is set to 1, the PLL is not used.

Conditions:

SetClockEdge {Pos {Display 1}}

parameters: Pos: valid values: 0 or 1. Specifies the output clock edge on which the output data toggle. 1 for rising edge; 0 for inverted clock.
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

returns:

description: Defines the phase relation between the output clock and the device internal clock. When the clock edge is set to 1, the output clock is in phase with the device internal clock; when set to 0, there is a 180° phase shift between the clocks.

conditions:

SetContinuousClock {}

parameters: None.

returns:

description: Sets the output clock generation in continuous mode. In this case, a permanent and continuous clock is generated independently of the data stream.

conditions:

SetCtrlDefaultVal {Val {Display 1}}

parameters: Val: hexadecimal (decimal) input value ranging from 0x00 (0) to 0x3F (63).
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

returns:

description: The static pattern is applied on the control lines with the first data sent out to the device. The level remains constant until a new control pattern is defined. If a new pattern is defined, it will only be applied on the control lines with the next outgoing data sample.

conditions:

SetCtrlMaskOut {MaskOut {Display 1}}

parameters: MaskOut: hexadecimal (decimal) input value ranging from 0x00 (0) to 0x3F (63).

Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

returns:

description: The control mask selects the device control lines on which the default static levels or the periodic patterns have to be applied. When a mask bit is set to 0, the corresponding control line is masked. The mask is given as a hexadecimal value; each of its bits corresponds to one of the six control lines, MSB to LSB.

conditions:

SetCtrlTrigMask {Mask {Display 1}}

parameters: Mask: hexadecimal (decimal) input value ranging from 0x00 (0) to 0x3F (63).
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

returns:

description: When the external triggering is used, the trigger mask selects the control lines to be used as trigger inputs. When a mask bit is set to 0, the corresponding control line is masked for triggering. The mask is given as a hexdecimal value; each of its bits corresponds to one of the six control lines, MSB to LSB

conditions:

SetCtrlTrigPattern {Pattern {Display 1}}

parameters: Pattern: hexadecimal (decimal) input value ranging from 0x00 (0) to 0x3F (63).
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

returns:

description: Defines the pattern to detect on the trigger inputs to generate the trigger event.

conditions:

SetDataMaskOut {MaskOut {Display 1}}

parameters: MaskOut: hexadecimal (decimal) input value ranging from 0x0000 (0) to 0xFFFF (65535).
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

returns:

description: Applies a mask onto the data output lines to enable / disable them. The mask MSB and LSB respectively correspond to data line 15 and data line 0. A mask bit set to 1 enables the data line; a mask bit set to 0 disables it: the corresponding data line is in high-impedance.

conditions:

SetEdgeTrigger {Enable {Display 1}}

parameters: Enable: Valid values: 0 or 1 – 0 for 'level trigger', 1 for 'edge trigger'
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By

default set to 0.

returns:

description: Selects the triggering mode: in edge triggering mode (Enable = 1), the ADWG run will be triggered upon detection of a transition to the programmed trigger pattern. In level triggering mode (Enable = 0), the ADWG run is triggered upon detection of the programmed trigger pattern.

see also:

SetHoleClock {}

parameters: None

returns:

description: Enables the 'hole' mode to generate the output clock. In this mode, a clock pulse is generated only when a data is sent out. When no data is sent out, the output clock remains at a low level.

conditions:

SetInfiniteLoop {Enable}

parameters: Enable: boolean value (0 or 1).

returns:

description: Enables / disables the infinite loop mode. In this mode, the device infinitely loops over the set of loaded data.

conditions: The amount of data to transmit must not exceed the device embedded memory.

SetInternalTrigger {Internal {Display 1}}

parameters: Internal: valid values: 0 or 1.
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.

returns:

description: Defines the triggering mode. When Internal is set to 1, the internal triggering mode is selected; when Internal is set to 0, the external triggering mode is selected.

conditions:

SelectIOVoltage {IOVoltage}

parameters: IOVoltage: integer value representing the IO voltage, the IO voltage can be internally generated of user applied. The voltage level is defined in millivolts.

returns:

description: This function only takes the following predefined values: 3300, 2500, 1800, 1500 and 1200. The nearest value must be selected when the user applies a different external voltage level. For example, set IOVoltage to 2500 when 2.7V is applied.
The default value is 3300.

conditions:

SetOutClockRatio {Ratio}

parameters: Ratio: decimal value ranging from 1 to 65535.

:
returns:
description: Defines the output clock ratio with respects to the operating clock: Frequency(Output Clock) = Frequency(Operating Clock) / Ratio.
conditions:

SetOutputClock {Output {Display 1}}
parameters : Output: valid values: 0 or 1.
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.
returns:
description: Enables or disables the generation of an output clock. Output set to 1 enables the generation of the output clock; Output set to 0 disables it.
conditions:
see also: SetOutClockRatio{}, SetHoleClock{}, SetClockEdge{}

SetReqClock {Freq}
parameters : Freq: integer value representing the requested frequency in Hz. Range: from 800 (800 Hz) to 100000000 (100 MHz), according to the used device's specifications.
returns:
description: Defines the requested operating clock frequency. The device will actually set the operating clock frequency to the closest frequency available.
conditions:
see also: GetReqClock{}, GetSynthClock{}

SetStaticData {Data {Display 1}}
parameters : Data: hexadecimal (decimal) input value ranging from 0x0000 (0) to 0xFFFF (65535).
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.
returns:
description: Applies the specified data pattern to the device system connector pins.
conditions:

Terminate {}
parameters : None
returns:
description: Closes the ADWG TCL session. This function unloads from memory the class instance used to interact with the GP Series device in ADWG mode, and hence unconnects the ADWG session from the 'Smart Router'.
conditions:

Unsupervised {Enable {Display 1}}
parameters : Enable: valid values: 0 or 1.
Display: Optional parameter. Displays additional information messages when set to 1. Display disabled when set to 0. By default set to 1.
returns:
description: Enables or disables the unsupervised operating mode. In this mode, the

traffic part reserved to the control between the host and the device is reduced to a minimum. As a consequence, more bandwidth is available for the data.

conditions:

UseExtClock {}

parameters: None.

returns:

description: Configures the device to operate with an externally supplied operating clock as reference.

conditions:

UseIntClock {}

parameters: None.

returns:

description: Configures the device to operate with the internal operating clock as reference.

conditions:

Ver {}

parameters: None.

returns: A decimal value representing the software version.

description: Returns the software version as a decimal value. Example: software version 1.04 ➔ returns 0x104 converted as a decimal value: 260.

conditions:

WriteConfAndDataFile {FileName}

parameters: FileName: specifies the procedure output file name (including path).

returns: 0 when write succesful.

description: Writes the device configuration and the data samples to a file.

conditions:

WriteConfFile {FileName}

parameters: FileName: specifies the procedure output file name (including path).

returns: 0 when write succesful.

description: Writes the device configuration to a file.

conditions:

WriteSingleSample {Sample {Display 0}}

parameters: Sample: Value of the sample to be written into memory.
Display: optional parameters. When 1, enables the display of messages in the TCL shell.

returns: 0 when write succesful.

description: Appends the memory data buffer with the data sample specified as argument. With multiple calls to this functin, it is possible to create a set of samples to be transmitted with the device without having to use a data file, potentially offering a fast way to load data into memory.

conditions:

3.3 Reference Sequences and Scripts

Examples of the TCL scripts are provided with the TCL library. They can be accessed in the Examples group located under the 8PI Control Panel program group created in your start menu during the installation of the application and driver on your computer.

Figure 4: Program group with TCL script examples