TABOR ELECTRONICS

PROTEUS
infinite possibilities

Proteus Multi-Channel Triggering - Application Note

Rev. 1.0

Simulate, Stimulate, Test...

For more information or to schedule a demo contact info@tabor.co.il | www.taborelec.com

Warranty Statement

Products sold by Tabor Electronics Ltd. are warranted to be free from defects in workmanship or materials. Tabor Electronics Ltd. will, at its option, either repair or replace any hardware products which prove to be defective during the warranty period. You are a valued customer. Our mission is to make any necessary repairs in a reliable and timely manner.

Duration of Warranty

The warranty period for this Tabor Electronics Ltd. hardware is one year, except software and firmware products designed for use with Tabor Electronics Ltd. Hardware is warranted not to fail to execute its programming instructions due to defect in materials or workmanship for a period of ninety (90) days from the date of delivery to the initial end user.

Return of Product

Authorization is required from Tabor Electronics before you send us your product for service or calibration. Call your nearest Tabor Electronics support facility. A list is located on the last page of this manual. If you are unsure where to call, contact Tabor Electronics Ltd. Tel Hanan, Israel at 972-4-821-3393 or via fax at 972-4-821-3388. We can be reached at: support@tabor.co.il

Limitation of Warranty

Tabor Electronics Ltd. shall be released from all obligations under this warranty in the event repairs or modifications are made by persons other than authorized Tabor Electronics service personnel or without the written consent of Tabor Electronics.

Tabor Electronics Ltd. expressly disclaims any liability to its customers, dealers and representatives and to users of its product, and to any other person or persons, for special or consequential damages of any kind and from any cause whatsoever arising out of or in any way connected with the manufacture, sale, handling, repair, maintenance, replacement or use of said products. Representations and warranties made by any person including dealers and representatives of Tabor Electronics Ltd., which are inconsistent or in conflict with the terms of this warranty (including but not limited to the limitations of the liability of Tabor Electronics Ltd. as set forth above), shall not be binding upon Tabor Electronics Ltd. unless reduced to writing and approved by an officer of Tabor Electronics Ltd. This document may contain flaws, omissions, or typesetting errors. No warranty is granted nor liability assumed in relation thereto. The information contained herein is periodically updated and changes will be incorporated into subsequent editions. If you have encountered an error, please notify us at support@taborelec.com. All specifications are subject to change without prior notice. Except as stated above, Tabor Electronics Ltd. makes no warranty, express or implied (either in fact or by operation of law), statutory or otherwise; and except to the extent stated above, Tabor Electronics Ltd. shall have no liability under any warranty, express or implied (either in fact or by operation of law), statutory or otherwise.

Proprietary Notice

This document and the technical data herein disclosed, are proprietary to Tabor Electronics, and shall not, without express written permission of Tabor Electronics, be used, in whole or in part to solicit quotations from a competitive source or used for manufacture by anyone other than Tabor Electronics. The information herein has been developed at private expense and may only be used for operation and maintenance reference purposes or for purposes of engineering evaluation and incorporation into technical specifications and other documents, which specify procurement of products from Tabor Electronics.

Document Revision History

Table 1.1 Document Revision History

Revision	Date	Description	Author
1.0	9-Apr-2024	• Original release.	Saeed Ghanem

Acronyms & Abbreviations

Table 1.2 Acronyms & Abbreviations

Acronym	Description
µs or us	Microseconds
ADC	Analog to Digital Converter
AM	Amplitude Modulation
ASIC	Application-Specific Integrated Circuit
ATE	Automatic Test Equipment
AWG	Arbitrary Waveform Generators
AWT	Arbitrary Waveform Transceiver
BNC	Bayonet Neill–Concelm (coax connector)
BW	Bandwidth
CW	Carrier Wave
DAC	Digital to Analog Converter
dBc	dB/carrier. The power ratio of a signal to a carrier signal, expressed in decibels
dBm	Decibel-Milliwatts. E.g., 0 dBm equals 1.0 mW.
DDC	Digital Down-Converter
DHCP	Dynamic Host Configuration Protocol
DSO	Digital Storage Oscilloscope
DUC	Digital Up-Converter
ENoB	Effective Number of Bits
ESD	Electrostatic Discharge
EVM	Error Vector Magnitude
FPGA	Field-Programmable Gate Arrays
GHz	Gigahertz
GPIB	General Purpose Interface Bus
GS/s	Giga Samples per Second
GUI	Graphical User Interface
HP	Horizontal Pitch (PXIe module horizontal width, 1 HP = 5.08mm)
Hz	Hertz
IF	Intermediate Frequency
I/O	Input / Output
IP	Internet Protocol
IQ	In-phase Quadrature
IVI	Interchangeable Virtual Instrument
JSON	JavaScript Object Notation
kHz	Kilohertz

Acronym	Description
LCD	Liquid Crystal Display
LO	Local Oscillator
MAC	Media Access Control (address)
MDR	Mini D Ribbon (connector)
MHz	Megahertz
MIMO	Multiple-Input Multiple-Output
ms	Milliseconds
NCO	Numerically Controlled Oscillator
ns	Nanoseconds
PC	Personal Computer
PCAP	Projected Capacitive Touch Panel
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PRBS	Pseudorandom Binary Sequence
PRI	Pulse Repetition Interval
PXI	PCI eXtension for Instrumentation
PXIe	PCI Express eXtension for Instrumentation
QC	Quantum Computing
Qubits	Quantum bits
RADAR	Radio Detection And Ranging
R&D	Research & Development
RF	Radio Frequency
RT-DSO	Real-Time Digital Oscilloscope
s	Seconds
SA	Spectrum Analyzer
SCPI	Standard Commands for Programmable Instruments
SFDR	Spurious Free Dynamic Range
SFP	Software Front Panel
SMA	Subminiature version A connector
SMP	Subminiature Push-on connector
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
TFT	Thin Film Transistor
T&M	Test and Measurement
TPS	Test Program Sets
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VCP	Virtual COM Port
Vdc	Volts, Direct Current
V p-p	Volts, Peak-to-Peak
VSA	Vector Signal Analyzer
VSG	Vector Signal Generator
WDS	Wave Design Studio

Contents

Figures

Tables

1 About this Application Note

This application note explains how to trigger all Proteus channels simultaneously.

1.1 Related Documentation

- Proteus Series Arbitrary Waveform Transceiver Programming Manual
- Proteus Module User Manual
- PXE6410 User Manual

2 Trigger all Proteus Channels Simultaneously

2.1 Introduction

While the Proteus system offers versatile waveform creation capabilities these would not be very effective without the ability to tightly control when and how the waveforms start and stop. For this purpose, the Proteus system offers an extremely flexible and sophisticated triggering system that controls the start and stop of the waveforms. The Proteus AWG responds to various trigger sources such as: external trigger signal from the TRIG1 And TRIG2 inputs, Internal trigger generator with programmable trigger period, and BUS trigger for triggering via the controlling PC using software commands. The trigger source level for TRIG 1 and TRIG 2 inputs is programmed per trigger source and is common to all channels. All other trigger source attributes are independent and can be different for each channel. This means that users can set different trigger parameters for each channel on the same trigger input. In addition, trigger signals are used to both enable and abort waveform or task generation.

2.2 Trigger Sources

2.2.1 External Source

An external trigger is a signal sourced from outside the instrument itself, which initiates a specific action or response within the instrument's circuitry. These triggers can originate from a diverse array of sources, including external sensors, signal generators, or even other instruments within a test setup.

2.2.2 Internal Source

An internal trigger is a signal generated within the instrument itself that initiates a specific action or response. From capturing waveforms on an oscilloscope to triggering measurements on a spectrum analyzer, internal triggers serve as the guiding force, ensuring that critical events are detected and processed with precision.

2.2.3 CPU Source

A CPU trigger is a command or event generated by the CPU itself that initiates a specific action or response within the instrument's circuitry. As the computational powerhouse of the instrument, the CPU leverages its processing capabilities to detect, analyze, and act upon incoming signals or stimuli with remarkable speed and accuracy.

2.3 Trigger Modes in Proteus

The options for triggering all channels simultaneously are by a CPU or External trigger. Internal trigger works with each channel individually.

Note

If you work with a CPU trigger, you need to define the SCPI value as INT and not CPU.

3 How it Works?

To trigger all channels simultaneously you need to send some SCPI commands to the Proteus. The trigger can work in arbitrary or task table mode.

3.1 Arbitrary Mode

In arbitrary mode the waveforms that are stored in the memory can be generated one at a time by selecting the segment to be generated. This mode is used when there is no prior knowledge of the order in which the segments should be generated. The selection can be done either by a software command or by external signal if the DJ option is installed. For a detailed explanation on Arbitrary mode refer to the "Proteus Module User Manual".
You can choose which trigger source you want to use, CPU or EXTERNAL.
To trigger all channels simultaneously follow the following instructions:

1. First, you need to connect to your unit.
2. Download your waveforms to the relevant channels.
3. Turn off the continuous mode for each channel.

```
inst.send_scpi_cmd(':INIT:CONT OFF')
```

4. Define the triggers for all channels.

Example

If you choose an EXTERNAL trigger source replace INT with: TRG1 or TRG2.

```
inst.send_scpi_cmd(':INST:CHAN 1')
inst.send_scpi_cmd(':TRIG:COUP ON')
inst.send_scpi_cmd(':TRIG:SOUR:ENAB INT')
inst.send_scpi_cmd(':TRIG:SEL INT')
inst.send_scpi_cmd(':TRIG:STAT ON')
```

The CPU trigger must be sent when channel 1 is activate, after that you need to send the following SCPI commands:

```
inst.send_scpi_cmd(':INST:CHAN 1')
inst.send_scpi_cmd(':TRIG:CPU:MODE GLOBAL')
```

Now, you can send a trigger using the SCPI command:

```
inst.send_scpi_cmd('*TRG')
```

3.2 Task Table Mode

When the Proteus device is set to task mode, the waveforms are generated according to the task table. The task table is made up of different tasks. Each task in the task table contains multiple parameters, such as enable signal, abort signal, and jump condition that define the trigger functionality for the given task. To trigger all channels simultaneously you must choose one of two trigger sources: CPU or EXTERNAL. For a detailed explanation on Task mode refer to the "Proteus Module User Manual".

External

To work with External trigger, a valid trigger signal must be connected to one of the two external trigger inputs. In addition, the first task of all channels must be set to have an enable signal, in the example, TRG1.

CPU

To work so that a single CPU trigger triggers multiple channels, the instrument must be configured so that the internal trigger path is used to transfer the CPU trigger to all channels as follows:

1. Send the SPI command: ":TRIG:COUPLE ON" , this will configure the instrument so that a trigger sent to CH1 is transferred to all channels.
2. Select CH1 as the active channel
3. Send a *TRG

4 Setup

The picture below shows the setup of Arbitrary mode.

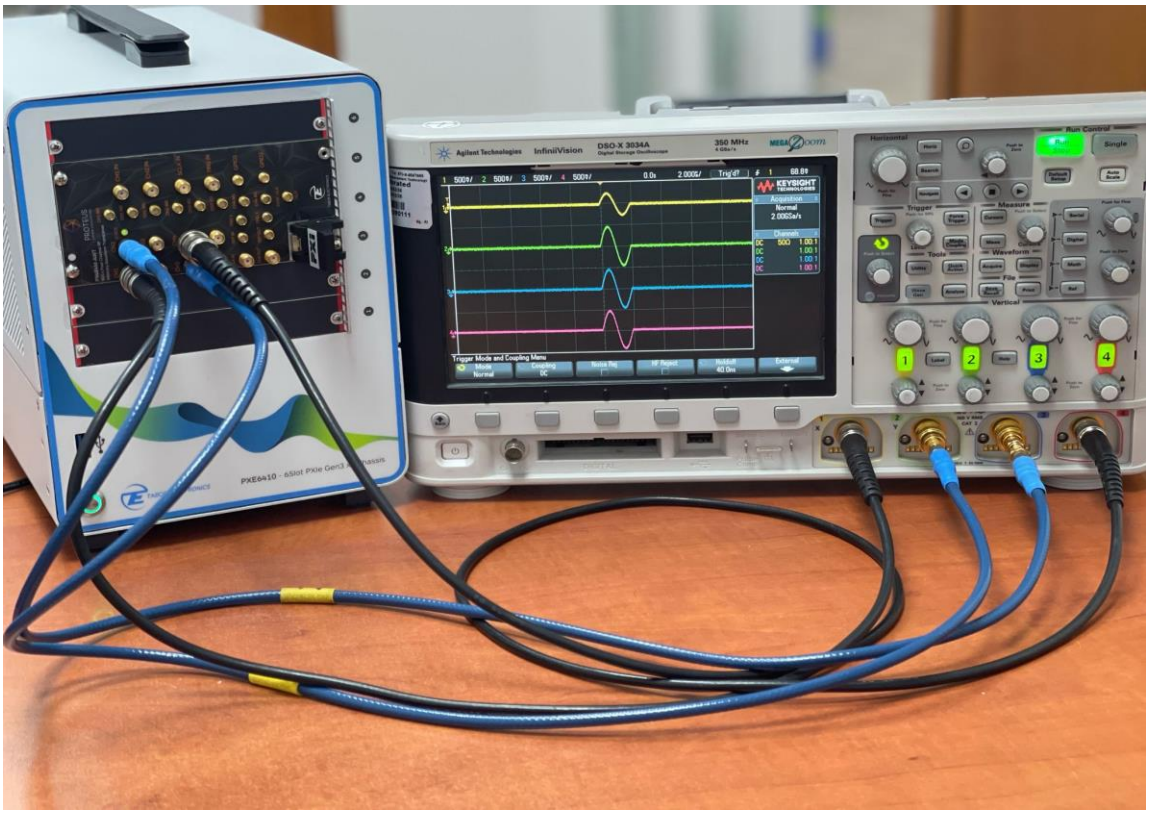

Figure 4.1 PXE6410 6 Slot PXIe Chassis with Proteus PXIe Module Setup Arbitrary Mode

1. Connect all the Proteus output channels to the oscilloscope input channels.
2. If you use an External trigger connect it to the input TRG1 or TRG2 on the unit.
3. Connect your control PC to the Proteus.
4. Send the Figure 5.2 Python sample script to your Proteus

5 Python Sample Scripts

The Python scripts are provided as attachments.

Note

You should open the PDF file using the free Adobe reader. It can be downloaded from https://get.adobe.com/reader/. As an alternative, you can also download the scripts from the Tabor download site at https://www.taborelec.com/Downloads.

1. Click the “paper clip” icon in the attachment pane.
2. Right-click the file and select “Save Attachment…” to download the file.

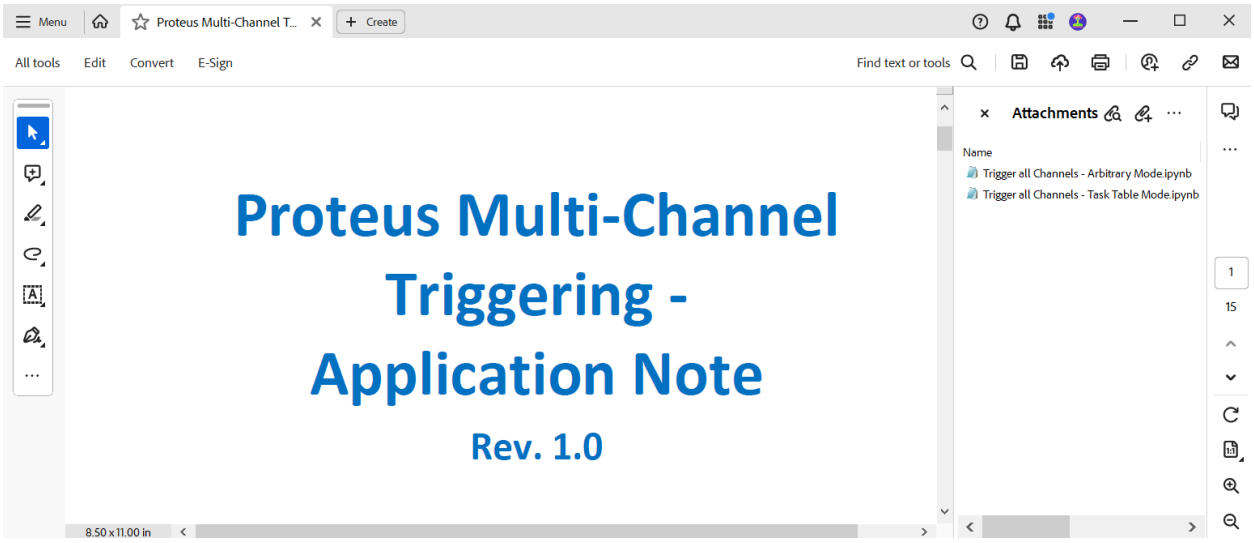

Figure 5.1 Python Script Attachment

5.1 Arbitrary Mode

The following partial script will help you to understand the details deeply. The SCPI commands used in the Python script are using the Jupyter Notebook.

```
#select the programmable channel
inst.send_scpi_cmd(':INST:CHAN 1')
#Turn off the continuous mode
inst.send_scpi_cmd(':INIT:CONT OFF')
#all channels in the instrument receive the trigger from channel1
inst.send_scpi_cmd(':TRIG:COUP ON')
#The source of the enabling signal of the selected channel
inst.send_scpi_cmd(':TRIG:SOUR:ENAB INT')
#Select the programmable  trigger
inst.send_scpi_cmd(':TRIG:SEL INT')
#Enable / Disable the selected external trigger (channel dependent)
inst.send_scpi_cmd(':TRIG:STAT ON')
#all channels receive the same CPU trigger simultaneously.
inst.send_scpi_cmd(':TRIG:CPU:MODE GLOBAL')

inst.send_scpi_cmd(':INST:CHAN 2')
inst.send_scpi_cmd(':INIT:CONT OFF')
inst.send_scpi_cmd(':TRIG:COUP ON')
inst.send_scpi_cmd(':TRIG:SOUR:ENAB INT')
inst.send_scpi_cmd(':TRIG:SEL INT')
inst.send_scpi_cmd(':TRIG:STAT ON')

inst.send_scpi_cmd(':INST:CHAN 3')
inst.send_scpi_cmd(':INIT:CONT OFF')
inst.send_scpi_cmd(':TRIG:COUP ON')
inst.send_scpi_cmd(':TRIG:SOUR:ENAB INT')
inst.send_scpi_cmd(':TRIG:SEL INT')
inst.send_scpi_cmd(':TRIG:STAT ON')

inst.send_scpi_cmd(':INST:CHAN 4')
inst.send_scpi_cmd(':INIT:CONT OFF')
inst.send_scpi_cmd(':TRIG:COUP ON')
inst.send_scpi_cmd(':TRIG:SOUR:ENAB INT')
inst.send_scpi_cmd(':TRIG:SEL INT')
inst.send_scpi_cmd(':TRIG:STAT ON')
```

```
inst.send_scpi_cmd(':INST:CHAN 1')
inst.send_scpi_cmd('*TRG')
```

Figure 5.2 SCPI Commands for Triggering All Channels Arbitrary Mode

5.2 Task Table Mode

The following partial script is an example of how to trigger multiple channels in Task mode either with external or CPU trigger. The SCPI commands used in the Python script are using the Jupyter Notebook.

```
# Set the thrshhold level of the trigger
inst.send_scpi_cmd(':TRIG:LEV 0.1')

resp = inst.send_scpi_cmd(':INST:ACT {0}'.format(1))
for ichan in range(num_channels):
    channb = ichan + 1
    # Select channel- Trigger source: EXT=TRG1 ,INT=CPU
    cmd = ':INST:CHAN {0}'.format(channb)
    inst.send_scpi_cmd(cmd)
    inst.send_scpi_cmd('TRIG:SOUR:ENAB {}'.format('TRG1'))
    inst.send_scpi_cmd('TRIG:SEL {}'.format('TRG1'))
    inst.send_scpi_cmd('TRIG:STAT ON')
```

```
# Define task-table of 3 tasks in each channel.
# The first task shall wait for trigger1.
# In order to

tasklen = 3
for ichan in range(num_channels):
    channb = ichan + 1
    # Select channel
    cmd = ':INST:CHAN {0}'.format(channb)
    inst.send_scpi_cmd(cmd)

    # Compose the task-table rows:
    cmd = ':TASK:COMP:LENG {0}'.format(tasklen)
    inst.send_scpi_cmd(cmd)

    for itask in range(tasklen):
        tasknb = itask + 1
        segnb = itask + 1
        nloops = 2 ** tasknb

        cmd = ':TASK:COMP:SEL {0}'.format(tasknb)
        inst.send_scpi_cmd(cmd)

        inst.send_scpi_cmd(':TASK:COMP:TYPE SING')

        cmd = ':TASK:COMP:SEGM {0}'.format(segnb)
        inst.send_scpi_cmd(cmd)

        cmd = ':TASK:COMP:LOOP {0}'.format(nloops)
        inst.send_scpi_cmd(cmd)
```

```
        if 1 == tasknb:
        # Trigger source: EXT=TRG1 ,INT=CPU
        # in case of :TRIG:COUPLE ON need to put INT instead of CPU??
            cmd = ':TASK:COMP:ENAB TRG1'
            inst.send_scpi_cmd(cmd)
        else:
            cmd = ':TASK:COMP:ENAB NONE'
            inst.send_scpi_cmd(cmd)

        if tasklen == tasknb:
            cmd = ':TASK:COMP:NEXT1 1'
            inst.send_scpi_cmd(cmd)
        else:
            cmd = ':TASK:COMP:NEXT1 {0}'.format(tasknb + 1)
            inst.send_scpi_cmd(cmd)

        # Write the task-table
        inst.send_scpi_cmd(':TASK:COMP:WRIT')

        # Set Task-Mode
        inst.send_scpi_cmd(':FUNC:MODE TASK')

if 0:
    # write the task table rows to the task-table of each channel
    for ichan in range(num_channels):
        channb = ichan + 1
        # Select channel
        cmd = ':INST:CHAN {0}'.format(channb)
        inst.send_scpi_cmd(cmd)
        # Write the task-table
        inst.send_scpi_cmd(':TASK:COMP:WRIT')

        # Set Task-Mode
        inst.send_scpi_cmd(':FUNC:MODE TASK')

resp = inst.send_scpi_query(':SYST:ERR?')
print(resp)
```

```
# :TRIG:COUPLE ON will require trigger from master, if single module in use and single channel, it should be off (default)
# If more then 1 channel requires trigger, it should be ON and tasks trigger should be INT instead of CPU
resp = inst.send_scpi_cmd(':TRIG:COUPLE ON')
print(resp)
```

```
#for trigger source INT send:
inst.send_scpi_cmd(':INST:CHAN 1')
#trigger command:
inst.send_scpi_cmd('*TRG')
```

Figure 5.3 SCPI Commands for Triggering All Channels Task mode