TABOR ELECTRONICS

PROTEUS
Infinite possibilities

Proteus AWT Pulse Counter – Application Note

Rev. 1.0

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

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 use the Proteus AWT pulse counter and provides an example script.

1.1 Related Documentation

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

2 Pulse Counting with Proteus AWT

2.1 Introduction

A pulse counter is a device or a circuit that counts the number of pulses it receives over time. In electronic and digital systems, pulses are typically short-duration voltage changes that represent digital signals. The pulse counter is commonly used in various applications, including measuring the frequency of a signal, counting events, monitoring rotations in machinery, or tracking the number of occurrences of a specific event.

2.2 Pulse Counter Operation

The pulse counter begins its operation by receiving an input signal. This signal consists of pulses, which could be generated by various sources like sensors, switches, or other electronic devices. The pulse counter has a mechanism to detect the rising or falling edges of the input signal, which typically represent the beginning or end of a pulse. This detection is crucial for accurately counting pulses. Once a pulse is detected, the pulse counter increments the counter by one. The counter keeps track of the total number of pulses received since the counting process started. The counted value may be displayed on a digital display, sent to a microcontroller, or stored for further analysis. In some cases, the pulse counter may have additional features like resetting the count or triggering external events based on a predefined count threshold.

2.3 Pulse Counter in Proteus

The Proteus AWT provides a pulse counter with the following options.

1. “the Pulse-counter Trigger”: INT/EXT
2. “Pulse-counter window type”: FIX/GATE
3. “the Fixed window width”. The range of the window width is: 12.5[ns] to 15[s].

The maximum number of pulses is $2^{32} - 1 = 4294967295$.].

2.3.1 Number of Pulses Equation

- N.P= Number of Pulses
- TW=Time Window
- T=1 Pulse Cycle
- F= Frequency

$$N.P = \frac{TW}{T} = TW * F$$

You shall use the SCPI commands to operate the pulse counter.

2.3.2 How it Works?

A square signal is received at the input of the ADC. The signal is converted to digital data of 12 bits and is sent to the FPGA. The FPGA makes a comparison between the data and a fixed threshold value. Every time that the received data is higher than the fixed value the FPGA increases the counter by 1. Once the “window width” time is elapsed, the FPGA sends a response with the counted number of pulses.
The flow chart below depicts the pulse counter in the Proteus.

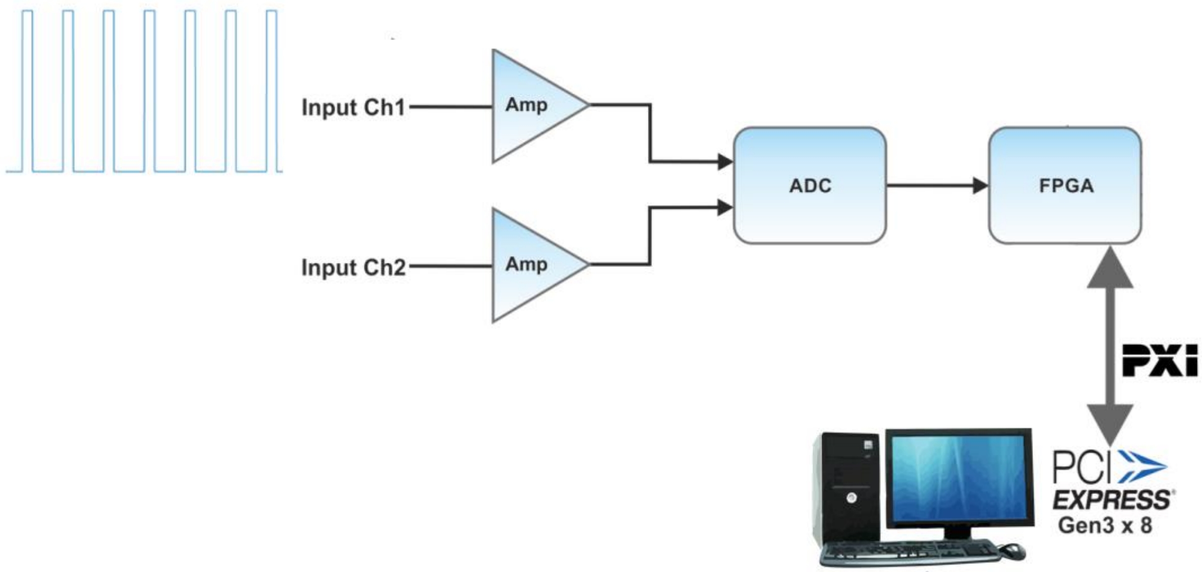

Figure 2.1 Proteus Pulse Counter

3 Pulse Counter Setup

To define a pulse counter in the Proteus AWT, you need to send a couple of SCPI commands to the Proteus. The following picture shows the pulse counter setup:

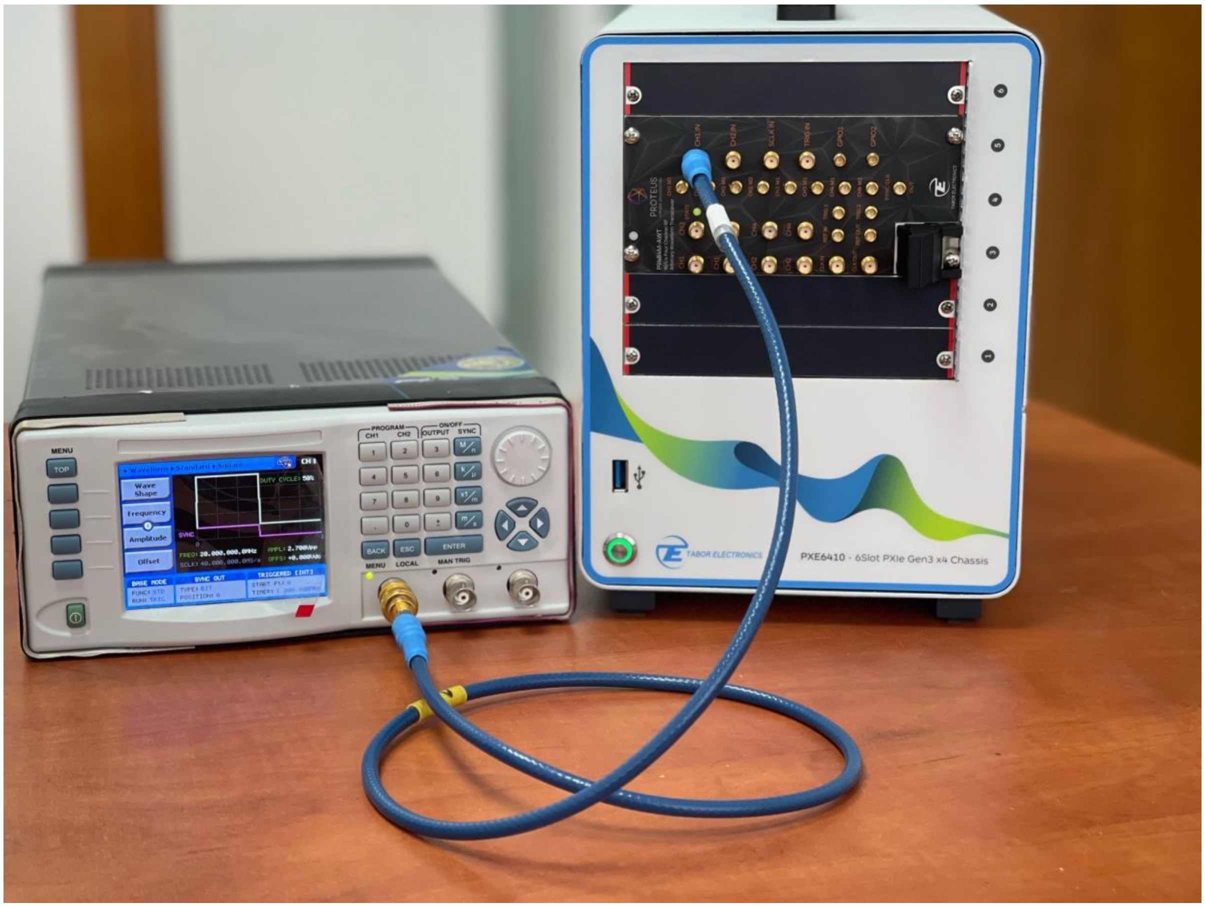

Figure 3.1 PXE6410 6 Slot PXIe Chassis with Proteus PXIe Module P9484M AWT Pulse Counter Setup

1. A square waveform from a function generator is connected to the input of the Proteus ADC (CH1 or CH2).
2. Connect your control PC to the Proteus.
3. Send the next python sample script to get your pulse number.

4 Python Sample Script

The Python script is provided as an attachment.

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 script 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.

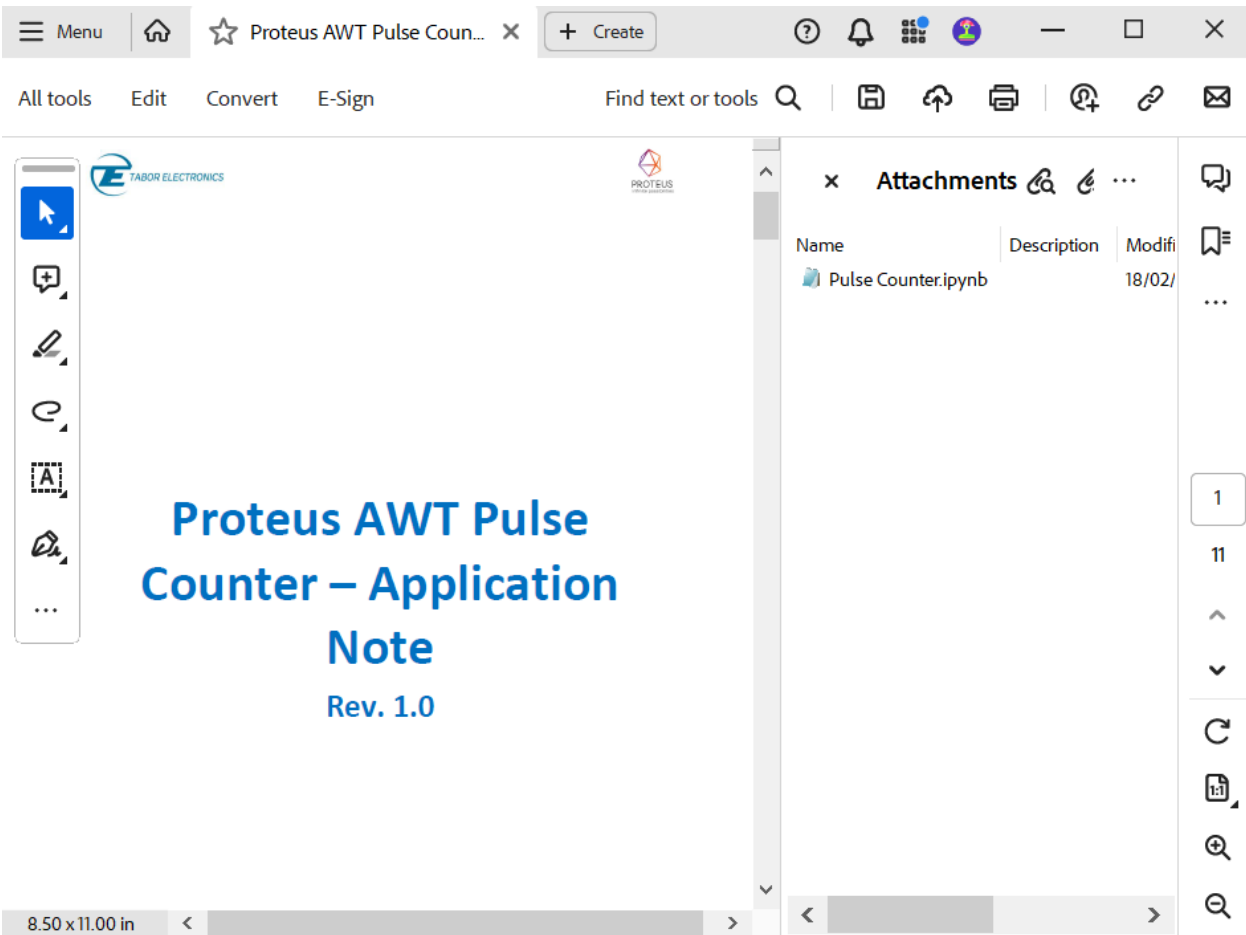

Figure 4.1 Python Script Attachment

The SCPI commands used in the partial Python script is using the Jupyter Notebook:

```
#Setup digitizer
#stop acquisition:
inst.send_scpi_cmd(':DIG:INIT OFF')

#Set CH1 as the active channel:
inst.send_scpi_cmd(':DIG:CHAN CH1')

#Trigger from CH1:
inst.send_scpi_cmd(':DIG:TRIG:SOUR CH1')

#Enable acquisition for the active channel:
inst.send_scpi_cmd(':DIG:CHAN:STAT ENAB')

#Sets LEV1 as the trigger threshold. Slope setting will set the positive and negative edge:
inst.send_scpi_cmd(':DIG:TRIG:TYPE EDGE')

#"Define the parameters of the pulse-counter of the digitizer(source & width):
inst.send_scpi_cmd(':DIG:PULS INT,FIX,0.01')

#Begin acquisition:
inst.send_scpi_cmd(':DIG:INIT ON')

#Force a trigger event for the digitizer:
inst.send_scpi_cmd(':DIG:PULS:TRIG:IMM')
```

```
# Get number of pulses
num_pulses = inst.send_scpi_query(":DIG:PULS:COUN?")
print("Number of PULSES: " + num_pulses)
```

Figure 4.2 SCPI Commands for Pulse Counter

Note

To get the number of pulses you shall wait until the time window has elapsed.