

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

PG3A
User's Manual

Version 1.13 – January 2018

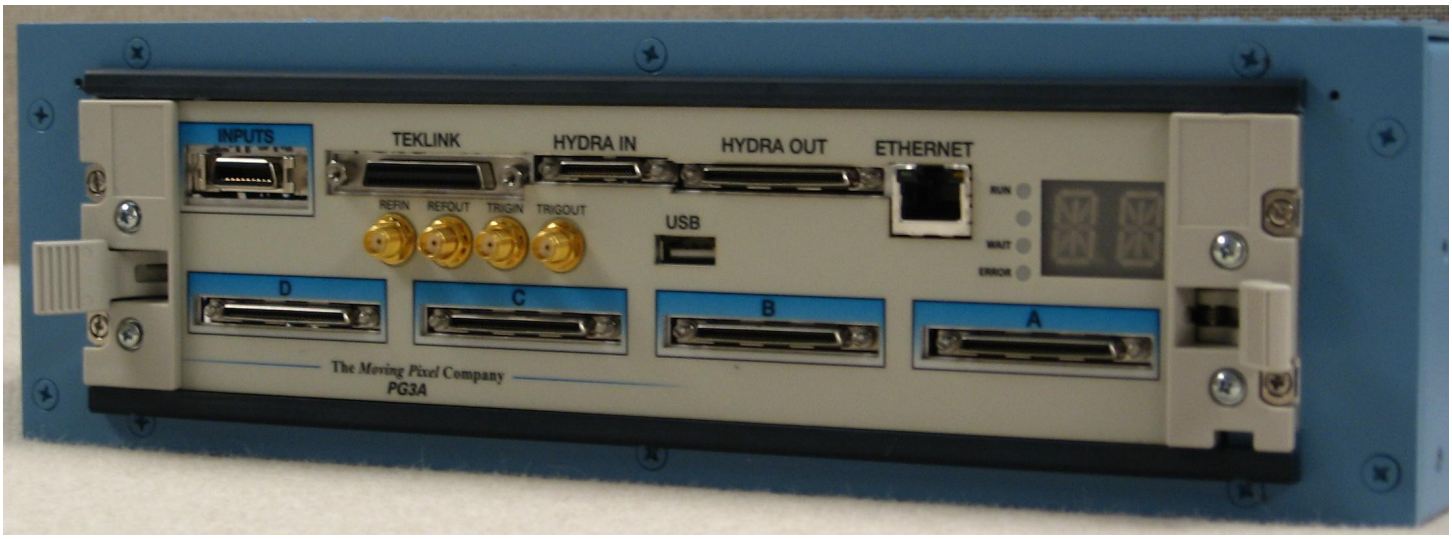


Table of Contents

1.1 Important Notes- by Serial Number....iv
1.1.1 Larger Memory....iv
1.1.2 New Connector Pinout....iv

1.2 General Safety Summary....v
1.2.1 To Avoid Fire or Personal Injury or Damage to Equipment....v
1.2.2 Symbols and Terms....vii

1.3 Environmental Considerations....viii

1.4 Before Operating the Instrument....ix

1.5 Contacting The Moving Pixel Company....x

2 Overview....1

2.1 Terms and Definitions....3

3 Pattern Generator Hardware....5

3.1 PG3A Module Front Panel....5

4 PGAppDotNet Software....8

4.1 Initial Installation....8
4.1.1 Vista Users....8
4.1.2 License files....8

4.2 Subsequent Installation/Upgrade....9

4.3 Quick Start....9

4.4 General Description....10

4.5 PGAppDotNet Windows....11
4.5.1 Connection Dialog....11
4.5.2 System Window....12
4.5.3 Setup Window....12
4.5.3.1 Module Setup Page....13
4.5.3.2 Channel Setup Page....16
4.5.3.3 Probe Setup Page....18
4.5.3.3.1 P375 Variable Probe Setup Window....21
4.5.3.3.2 P311 Dual-DAC Probe Dialog....22
4.5.3.3.3 P321 Serial Probe Support....24
4.5.3.4 Signals Setup Page....25
4.5.4 Program Window....26
4.5.4.1 Block Definition Page....26
4.5.4.1.1 Listing Window....27
4.5.4.1.2 Waveform Window....29
4.5.4.2 Sequence Definition Page....31
4.5.4.3 Event Definition Page....33
4.5.5 Status Monitor Window....33

4.6 PG Concepts....37
4.6.1 Merging Modules....37
4.6.2 Offline Operation and Offline Modules....37
4.6.3 Output Clocking....38
4.6.4 Events....38
4.6.5 Inhibits....39
4.6.6 Strobes....40
4.6.7 DDR Output....41
4.6.8 Branch Restrictions....41
4.7 Running the PG....42
4.8 Importing and Exporting Data....44
4.9 Menus....46
4.10 Legacy Tektronix PG Comparison....48
5 Updating the PG Firmware....51
6 Probe Notes....53
6.1 P300 Inputs probe....53
7 Troubleshooting notes....57
7.1 "PG is locked by another user. Please select a different PG."....57
7.2 Trouble with "CyUSB.sys"....57
8 Application Notes....58
8.1 Timing up a system....58
8.1.1 User-provided clock via the P300 Inputs Probe....58
8.1.2 PG to FIFO connection....58
9 Appendix A – PG3A Specifications....59
10 Appendix B – Filling Data From A File....63
10.1 Fill-From-File Operation....63
10.2 Fill-From-File File Definition....63
11 Warranty....66
12 Software License Agreement....67

1.1 *Important Notes- by Serial Number*

Larger Memory

Starting with Serial number P01233, units are field-upgradable to a larger memory size. All units with standard memory are equipped with ~32MVectors of memory. For units with Serial Number P01233 and greater, they can be upgraded with option PG3A-64M to ~64MVectors of memory

New Connector Pinout

Starting with serial number P01301, the pinout on the PG3A probe outputs and the Inputs Probe connector changed. The new cables are incompatible with the older PG units and the older PG accessories. The new PG is known as "PG3Axxx, Option -TB". The older PGs are now known as "PG3Axxx, Option TA". You can order all accessories as option TA (Type A) for the old style, and option TB (Type B) for the new style.

To distinguish the old from the new, all new cables and probes have the letter "B" near each affected connector, on each new cable, and on each new probe. Only "B"-marked cables should be used with "B"-marked PGs and "B" marked accessories. The old cables will mechanically fit on the new units and vice versa, however, the unit will not function.

None of the functionality of the PG changed and none of the specifications changed as a result of the pinout change. The new connector/cable pinout on the data output connectors are industry-standard VHDCI.

When ordering accessories, please specify the serial number of the PG or explicitly call out option TA or option TB. Type A cables and accessories must be used with Type A PGs; Type B cables and accessories must be used with Type B PGs.

If you require an old-style PG, these can still be ordered for at least the next year. Please contact us directly to place such an order.

1.2 General Safety Summary

Please review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it.

To avoid potential hazards, use this product only as specified.

Only The Moving Pixel Company should perform service procedures. Do not attempt service yourself. Return the unit to The Moving Pixel Company for service.

While using this product, you may need to access other parts of a larger system. Read the safety sections of the other component manuals for warnings and cautions related to operating the system.

To Avoid Fire or Personal Injury or Damage to Equipment

Qualified Personnel to Operate this Product
This manual and the product marking assumes that the user has more than basic electrical/electronic circuit knowledge and is a person comfortable by experience and education to safely connect probes to electrical circuits and systems. Do not operate this equipment nor connect it to any circuit or system if you are not qualified to do so by either education or experience.

Use Proper Power Cord.
Use only the power cord specified for this product and certified for the country of use.

Connect and Disconnect Properly.
Do not connect or disconnect probes or test leads while they are connected to a voltage source.

Ground the Product.
This product is grounded through the grounding conductor of the power cord. To avoid electric shock, the grounding conductor must be connected to earth ground.
Before making connections to the input or output terminals of the product, ensure that the product is properly grounded.

Observe All Terminal Ratings.
To avoid fire or shock hazard, observe all ratings and markings on the product. Consult the product manual for further ratings information before making connections to the product.
The inputs are not rated for connection to mains or Category II, III, or IV circuits.
Connect the probe reference lead to earth ground only, no floating grounds allowed.

Power Disconnect.
The power cord disconnects the product from the power source. Do not block the power cord; it must remain accessible to the user at all times.

Do Not Operate Without Covers.
Do not operate this product with covers or panels removed.

Do Not Operate With Suspected Failures.
If you suspect that there is damage to this product, have it inspected by **The *Moving Pixel* Company.**

Avoid Exposed Circuitry.
Do not touch exposed connections and components when power is present.

Do Not Operate in Wet/Damp Conditions.

Do Not Operate in an Explosive Atmosphere.

Keep Product Surfaces Clean and Dry.

Provide Proper Ventilation.
Refer to the manual's installation instructions for details on installing the product so it has proper ventilation.

Fusing Setup
The fusing for this equipment was setup based on the country for delivery and the line cord provided. The fuse(s) are user-accessible from the rear of the instrument. To examine or change the fusing setup:

- remove the line cord from the instrument's line entry unit
- using a small, flat blade screwdriver, pry the fuse holder cover up, away from the line entry unit. It will pivot to expose the fuse carrier.
- remove the fuse carrier from the body of the line entry unit. The fuse is on the opposite side. The carrier can be configured for either a single ¼ x 1 ¼ inch fuse or two IED 5 x 20mm fuses.

Symbols and Terms

These terms may appear in this manual:

WARNING. *Warning statements identify conditions or practices that could result in injury or loss of life.*

CAUTION. *Caution statements identify conditions or practices that could result in damage to this product or other property.*

These terms may appear on the product:

DANGER indicates an injury hazard immediately accessible as you read the marking.

WARNING indicates an injury hazard not immediately accessible as you read the marking.

CAUTION indicates a hazard to property including the product.

The following symbol(s) may appear on the product:



1.3 Environmental Considerations

This section provides information about the environmental impact of the product.

Product End-of-Life Handling
Observe the following guidelines when recycling an instrument or component:

Equipment Recycling. Production of this equipment required the extraction and use of natural resources. The equipment may contain substances that could be harmful to the environment or human health if improperly handled at the product's end of life. In order to avoid release of such substances into the environment and to reduce the use of natural resources, we encourage you to recycle this product in an appropriate system that will ensure that most of the materials are reused or recycled appropriately. The symbol shown below indicates that this product complies with the European Union's requirements according to Directive 2002/96/EC on waste electrical and electronic equipment (WEEE).



For information about recycling options, check the company Web site (www.movingpixel.com) or contact **The *Moving Pixel* Company.**

Restriction on Hazardous Substances Directive 2011/65/EU
EN 50581:2012 – Technical documentation for the assessment of electrical and electronic products with respect to the restrictions of hazardous substances.

1.4 Before Operating the Instrument

- Verify the ambient temperature: 5 °C to +45 °C.

- Verify the operating humidity:
  8% to 80% relative humidity at up to +32 °C
  5% to 45% relative humidity above +32 °C up to +45 °C noncondensing, and is limited by a maximum wet-bulb temperature of +29 °C
  (derates relative humidity to 32% at +45 °C)

- Verify the operating altitude: 3,000 m, derate maximum
  operating temperature by 1 °C per 300 meters above 1500 meters altitude.

CAUTION.
PGxxMod: Make sure the Mainframe air intakes and exhaust ports are clear. To ensure proper cooling, keep the bottom and sides of the instrument clear of obstructions.
PGxxCab: operate in upright position only, keeping cooling holes clear of obstructions.

Power Supply Requirements:

Source Voltage	100–240 VAC
Frequency	50–60 Hz
Power Consumption	100 W maximum

It may be helpful to check our website for any updates to this documentation before you operate the instrument: http://www.movingpixel.com/PG3A.html

Also, we have a Frequently Asked Questions page for this instrument:
http://www.movingpixel.com/PGFAQ.html

If you have any questions at all about the warnings or operating conditions listed above:

STOP!

Do not operate the instrument.

Contact **The *Moving Pixel* Company** and resolve your question before continuing.

1.5 Contacting The Moving Pixel Company

Phone	+1.503.626.9663	US Pacific Time Zone
Fax	+1.503.626.9653	US Pacific Time Zone
Address	**The *Moving Pixel* Company** 4905 SW Griffith Drive, Suite 106 Beaverton, Oregon 97005 USA	
Email	information@movingpixel.com	
Web site	http://www.movingpixel.com	
Documentation	http://www.movingpixel.com/PG3A.html http://www.movingpixel.com/PGFAQ.html	

2 Overview

This manual and the product marking assumes that the user has more than basic electrical/electronic circuit knowledge and is competent by either education or experience to operate this equipment.

The pattern generator (PG) is a digital instrument that outputs data patterns created off-line by a user, downloaded into the hardware, and played out in real-time under the control of application software.

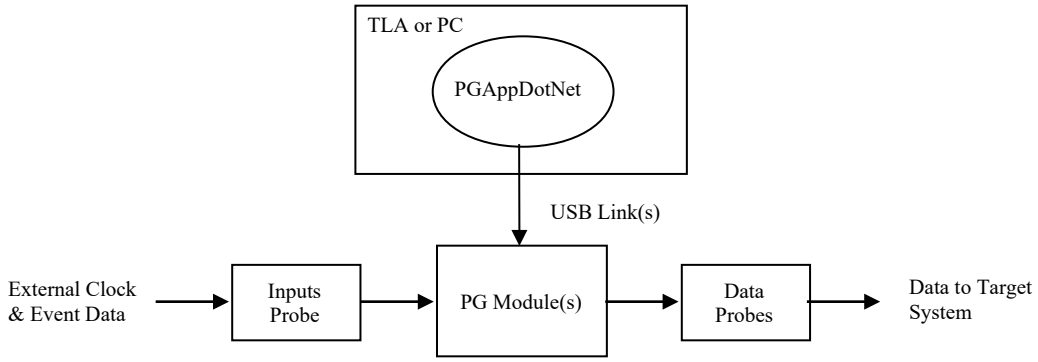


Figure 1 – PG Functional Components

Figure 1 shows the main functional components of the PG, which are

- One to four PG Modules, e.g. PG3As
- Data Probes
- Inputs Probe
- TLA or PC running PGAppDotNet under Microsoft XP or Windows 7 operating systems[1]

A quick overview of each of the components follows below.

PG Module

The PG module is the primary hardware engine of the pattern generator instrument. Controlled by the controlling PG application via a USB connection, the PG module plays data sequences comprising up to 64 data signals out to 4 data probes connected to the system under test (SUT).

A PG module may be provided in a form suitable for operation inside a Tektronix Logic Analyzer 7xx or 7xxxx mainframe (TLA) or provided in a cabinet for stand-alone operation. *Note that in the case where a PG module is installed in a*

[1] In the past, Microsoft Vista and Windows 2000 was supported. Also, the PG control application that is now shipped with each unit is PGAppDotNet to signify that remote control is now achieved through Microsoft .NET framework. The previous version was PGApp, which provided remote control through a COM/DCOM model.

TLA, a USB connection is still used for PG module control and the Tektronix TLA application is not in any way aware of the presence of the PG module. In general, there is no functional distinction between these two PG types[2] except for the availability of the TLA's backplane signals and reference clock. A master PG module installed in a TLA can use the backplane signals as input events, sources for the PG run trigger, and SignalOut destinations. In addition, the TLA 10 MHz reference clock can be used as the master clock reference for the PG.

As PG modules are connected to the host computer via USB, there is not practical limit to the number of PG modules that can be connected to a single host computer using a USB hub. Up to four modules can be "merged" to form a logical PG instrument wider than 64 data channels. In addition, multiple copies of the PG application software can run on a single host computer and control multiple PG instruments independently.

See Appendix B of this document for the specifications and characteristics of the PG3A pattern generator.

Data Probes

The pattern generator data probes connect the PG module to the SUT, providing a clock, configurable strobe, and up to 16 data channels to simulate signals in a test environment. Output data can be tri-stated as part of the programmed sequence or via external inhibit signals. A variety of different probes are available, each conforming to a particular I/O standard. Probes of different types can be attached to a single PG module. Specifications and operating instructions for all the probes and accessories was provided in a separate document. Users can also download the most recent version of these documents at www.movingpixel.com/PG3A.html.

Inputs Probe

The Inputs Probe receives an external clock and 8 external event signals from the SUT and passes them in to the master PG module. If present, the external clock from the Inputs Probe can be used as the PG's synchronizing timebase for data output and external event signals can be used for sequence flow control (i.e. stalling output or branching when an event is true). Note that the external event signals of the Inputs Probe are not used directly for sequence flow control but rather are used to define logical events by the user, consisting of a boolean sum-of-products of the external event signals.

See Appendix B of this document for the specifications and characteristics of the Inputs Probe.

PGAppDotNet

[2] In the rare instance that this manual references the Tektronix TLA mainframe, the information provided does not apply to a PG3A module configured for stand-along operation (i.e. in a cabinet).

PGAppDotNet is the controlling software for the pattern generator and runs under the Microsoft Windows operating system (XP, Win7-32, Win7-64). The host for GPAppDotNet may be the TLA in which a module resides, but it may also be a separate PC, where the user may find larger screen size useful.

PGAppDotNet allows users to aggregate output signals into logical groups, define data patterns for groups organized as blocks, and describe how blocks should be output as a sequence definition. Looping is supported and sequence flow control (i.e. waiting and branching) is achieved through the use of logical events, which are derived from boolean combinations of external signal inputs.

PGAppDotNet is freely available on our website. Users should note that it is updated often with new features and fixes:
www.movingpixel.com/PGUpgrade.html.

PGAppDotNet requires Microsoft .NET Framework 3.5.

Microsoft.NET3.5 is available from the web at:
http://download.microsoft.com/download/6/0/f/60fc5854-3cb8-4892-b6db-bd4f42510f28/dotnetfx35.exe
Please note that it is over 200MB!.

2.1 Terms and Definitions

The following list defines common terms used in this document.

Block – a user-defined and named set of vectors to output. Blocks have a minimum of 4 vectors and a maximum of 16 MVectors. They are the building units for sequences, where each line in the sequence directs the output of one block (possibly multiple times). Blocks are defined in the Block page of the Program window.

Branch – a conditional jump, i.e. one that is taken only when an event evaluates true. A branch is designated when the **JumpTo** and **JumpIf** fields of a sequence line are both specified. An important distinction between branches and jumps is that the number of unique branch destinations is limited to 64 (in PGAppDotNet release version 2.0). There are no such restrictions on the number of destinations.

Branch Hold-Off – the minimum distance in vectors a second branch may be placed from a branch destination. For the PG3A, the branch hold-off is 12 vectors (see section 4.6.8 for more discussion)

Cabinet – External frame with power supply for operating a PG3A module in stand-along mode, as opposed to plugged into a TLA mainframe slot.

Event – one of up to eight logical events used for sequence flow control, specified in the **WaitFor** and **JumpIf** fields of a sequence line. Events are named and are defined in the

Event page of the Program window as a sum-of-products, Boolean combination of event signals.

Event Signal – an input signal that may be used in the Boolean definition of a logical event. For PGApp initial release 2.0, there are nine potential event signals: the TrigIn input and eight external input events from the Inputs probe. In the future, TLA backplane signals for modules plugged into a TLA mainframe may be available as well.

Inhibit – a vector data bit defined in the sequence or an external input signal on a probe used to tri-state output data of a probe. If the external input inhibit signal is high or an inhibit data bit during playback is a '1', a probe will tri-state its outputs (i.e. an OR function of the two signals is implemented).

Instrument – Logical collection of 1-4 merged modules to form various vector widths for output. The merging of modules is achieved in software simply by selecting multiple PG modules in the PGAppDotNet connection dialog. The first module is designated as the master module, and the remaining modules as slave modules. In addition, merged modules must be connected together using a Hydra cable, which and distributes the master module's clocks and events to slave modules for synchronization.

Jump – designated by the **JumpTo** field in a sequence line to unconditionally force program flow to play from the given destination sequence line.

Module – the main hardware engine of the pattern generator instrument. Controlled by the controlling PG application via a USB connection, the PG module plays data sequences comprising up to 64 data signals out to 4 data probes connected to the SUT.

Probe – I/O devices that connect a PG module to the SUT. Data probes provide the output path for pattern data. The Inputs probe provides an external clock time-base for playback and eight input event signals to use for sequence flow control.

Strobe – an output signal that can be used as an enable or gated clock by the SUT. Data that controls the strobe is defined exactly as other output data.

Sequence – a description of how blocks are output from the PG, specifically the order of block output, how many times each block loops, and a designation of how events should control playback (i.e. via WaitFor stalls and conditional branches). Each line in a sequence describes the output of one block and there is no restriction on the number of lines that can reference the same block. Only one sequence is loaded and active in PGAppDotNet at any time. The sequence is defined in the Sequence page of the Program Window.

SUT – System Under Test

TLA – Tektronix Logic Analyzer

TMPC – The *Moving Pixel* Company

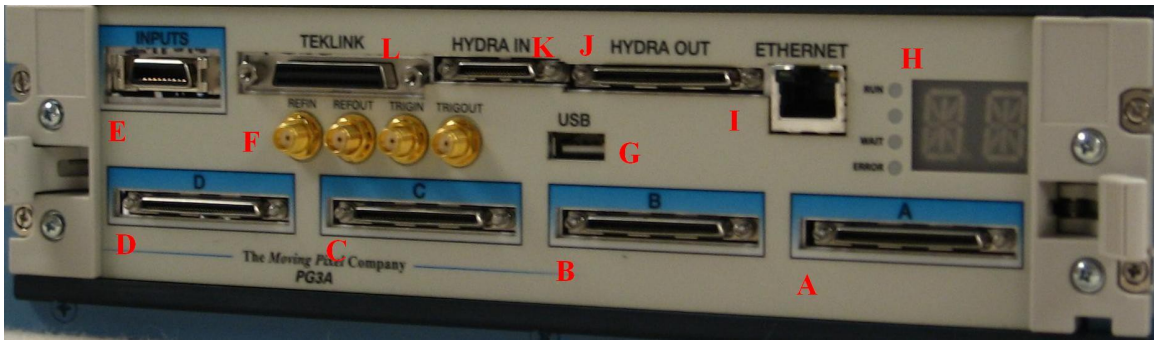
USB – Universal Serial Bus: a type of hardware/software communication method for passing data from/to the PG from a host computer, whether on a TLA or some other computer. Always required for operation

Vector – the data output during one clock cycle. For single modules, a vector consists of 64 data bits, 8 inhibit bits and 4 strobe bits. In merged modules, a vector consists of 64xN data bits, 8xN inhibit bits, and 4xN strobe bits where N is the number of merged modules (up to 4). Vectors are defined in a Listing Window or the Waveform window.

3 Pattern Generator Hardware

3.1 PG3A Module Front Panel

Please refer to the following picture while reading the description of the hardware front panel.



A – D:
These are the connections to the probes. No connection other than a connection to a PG output probe via the output probe cable should be made to this connector.

E:
This is the connection to the Inputs Probe. The Inputs Probe provides optional connections from the user system to the PG for the user clock (either differential or single-ended) and the 8 external event inputs. No connection other than a connection to the Inputs Probe via the Inputs Probe cable should be made to this connector.

F:
These four SMA connections provide connection for a time base reference and a trigger. The timebase reference connections ("Ref In", "RefOut") allow the user to provide a 10 MHz reference signal to the PG or accept a 10 MHz reference signal from the PG. The trigger connections ("Trig In", "TrigOut") allow the user to provide a trigger signal to the PG or accept a trigger signal from the PG. The specification of the signal characteristics for these connectors are listed in Appendix A, Table 2 and Table 3

G:
The USB connection is required for the control computer to communicate with the PG.

H:
The Status Display consists of 4 discrete LEDs and a two digit alpha-numeric display. The Run LED is green when on and indicates when the PG is actively outputting a pattern and has not stopped.
The Wait LED is yellow when on and indicates the PG is actively outputting a single line of the pattern, waiting on an external event (or a control from the GUI) to cause it to move on to the next pattern in the sequence.
The Error LED is red when on and indicates some kind of error condition when on. An error code appears in the alphanumeric display to indicate the error type. A table is listed in Appendix C that associates the error code with a description of the nature of the error.

The alphanumeric display normally displays a code that indicates the mode in which the PG is operating. For the PG3A and in the case of no error, it always displays "P1", meaning "PG unit #1".

I:
Ethernet connection is reserved for future use. It is not functional in the PG3A and may never be made functional.

J:
The Hydra Out connection is used on a PG module when it is configured as the master module of an instrument with multiple modules (and is not used otherwise). In this case, a Hydra cable is used to connect the master module to itself and its slaves.

K:
The Hydra In connection is used on a PG module when it is configured as part of an instrument with multiple modules (and is not used otherwise). In this case, a Hydra cable is used to connect the master module to itself and its slaves.

L:
TekLink[3] connection is reserved for future use. It is not functional in the PG3A and may never be made functional.

[3] TekLink is a trademark of Tektronix, Inc.

4 PGAppDotNet Software

4.1 Initial Installation

Install by running Setup.exe off of the distribution CD ROM, which by default locates the application directory in "C:\Program Files\TMPC\PGAppDotNet\PGApp".
Make you have the prerequisite Microsoft .Net package installed (see the end of section 2 above) before starting the installation. Please also read paragraph 7.2 in the troubleshooting section of this manual.

License files

For some PG options, a license file is required. At this writing, options –P3DDR and options running with the "PGRemote" application require license files. For license file installation, consult the appropriate documentation.

4.2 Subsequent Installation/Upgrade

- Uninstall the current version of PG software (Start->Settings->Control Panel... Add or Remove Programs option).
- Download and run the PGAppDotNet installer from the website: http://www.movingpixel.com/PGUpgrade.html. Upgrades are also available via CDROM. Please contact us if you prefer this method.
- Follow the installation instructions. The installation directory by default is "c:\Program Files\TMPC\PGAppDotNet".
- Reboot your machine.

4.3 Quick Start

There is much detail in the rest of this manual that might obscure the fact that the PG3A is quite easy to operate! Once the software is installed, it takes only a few simple steps to get a basic system going. If you don't understand any of the steps below, you should not complete the "Quick Start". Skip this section and continue with the manual.

- After reviewing the **safety** information at the beginning of this manual (Error: Reference source not foundError: Reference source not found), connect it to a **power** source.
- Attach an **output probe** via an **output cable** to the **"A"** connector.
- Attach the **USB cable** to the PG and the computer/TLA that will control the PG.
- Attach a **leadset** to the **output probe** and connect it to some instrument that can see activity – maybe connect a probe to the clock output and another one to any data output.
- Turn the PG **on**.
- Run PGAppDotNet.
- In the connection dialog that pops up, select the serial number of your PG module as the "Master Module". (If your PG's serial number is not present in the drop-down listbox, check cabling and power and try again.)
- At initial start up, a default system is defined to loop on a block that consists of 4 vectors all with zero data.
- Press **"Run"** in the PG app. Should now have a clock on the output of the probe! But the data is all zeros. Boring!
- Now, to change the data, **"Stop"** the PG and select "Prog" button in the System Window. Double click in the number field to the left of "Block". This should bring up the editor. **Edit** the first line, first group to all FFFFs.
- **"Run"** the PG. Your data line should be **toggling** at a 25% duty cycle
- Quick start now **complete**! See sections 4.6 and 4.7 for a more complete description on PG concepts and running the PG.

Hints:

1) Define groups carefully and with thought. If you change group definitions later, after you have defined vectors, you will find that the vectors will need to be entirely re-entered!
2) When holding a spin arrow in the GUI (delay, for example), the hardware will not reflect the state of the spin arrow setting until it is released.

4.4 General Description

The pattern generator program PGAppDotNet is the heart of the pattern generator product. Using PGAppDotNet, you can create blocks of data vectors that can be used to build complex pattern generator programs. The program uses external and internal events to determine
specific actions such as loops and branches to other data blocks. The program clock rate can be controlled by several clock sources.

You can also set up the pattern generator to single step through programs and output a single set of vectors with each clock cycle. This is useful for troubleshooting or debugging setups.

Finally, through the use of Microsoft's .NET Framework, PGAppDotNet can be controlled remotely from client applications. This document does not discuss further the details of remote operation. For complete information, please contact TMPC.

Central tasks in PGAppDotNet can be broadly categorized into three categories: Hardware Setup, Program Definition, and Operation.

Hardware setup is configuring the hardware. Configuration can be broken down into the following tasks:

- **Module Setup** – configuration of clock settings, input clock/event signal filtering and thresholds, and run modes.
- **Channel Setup** – logical group definition: names, widths, and channel assignments.
- **Probe Setup** – configuration of probe options such as output level, data delay, and strobe shape.
- **Signal Setup** – routing of signals for logical event definition, run triggering, and output of the logical SignalOut.

Program setup describes data content and playback flow control. The description is hardware independent, though it does rely on group definitions from channel setup. Program setup can be broken down into the following tasks:

- **Block Definition** – describes pattern data and logical grouping of vectors into named blocks.
- **Sequence Definition** – describes output data flow, i.e. how blocks are to be output, including waiting, looping, and branching constructs.
- **Event Definition** -- describes how input event signals should be combined to form the logical events used for flow control.

Finally, Operation simply describes control and behavior of the PG when running. The **Status** window shows the current module state and provides buttons for stepping (during step mode operation) and forcing events true. Also, many controls from the setup window and pages remain operational for the user to adjust while running, useful for overall system calibration and setup.

Subsequent sections of this chapter describe in more detail how PGAppDotNet is used.

4.5 PGAppDotNet Windows

The key to using PGAppDotNet is navigating and using its windows and dialogs. This section describes the windows of PGAppDotNet, their function and controls.

Connection Dialog

The first time PGAppDotNet is started, a selection dialog will appear for you to select the PG module set to connect to (see Figure 2). The module selections you make here define a particular logical instrument configuration having a particular width. For example, the most typical case is to select a specific master module and to set the slave modules to "None", which defines a single-module instrument with 64 data channels. For details on using multiple "merged" modules, see section 4.6.1.

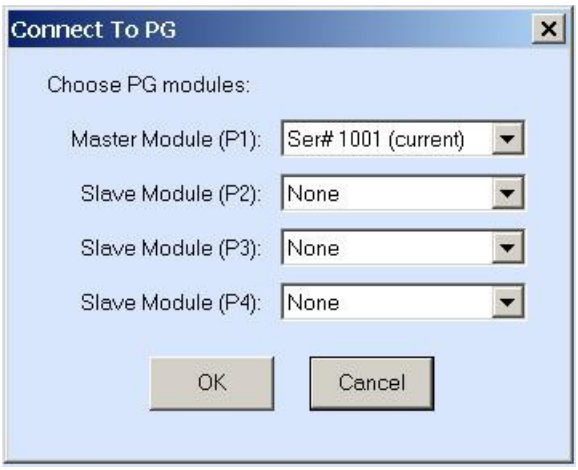


Figure 2 – Connection Dialog

The serial numbers of all powered modules connected via USB to the local machine will appear in the selection lists, as well as entries to select Offline modules (see section 4.6.2 for details about Offline modules). It is possible some modules will be marked "(in use)" which means another PGAppDotNet is running and currently connected to the module indicated. As only one PGAppDotNet may connect to a particular module at a time; modules marked (in use) are unavailable for connection and are listed for information purposes only.

Subsequent launches of PGAppDotNet will automatically check for the availability of modules from the last instrument configuration used. If the module(s) are available,

PGAppDotNet will connect automatically and bypass the dialog. Otherwise, PGAppDotNet will bring the dialog up normally for module selection.

In addition, the Connection dialog is also used to switch to a different instrument configuration. Use the menu option "**File->Choose PG…**" to bring up the dialog. The application will ask you if you wish to discard the current setup and data. Clicking "Yes" allows PGAppDotNet to proceed and immediately connect to the new module. If you click "No", a Save System dialog will be brought up first to allow you to save your setup and data. If you click "Cancel", nothing is saved and no new connections are made.

System Window

Once connection has been made to an instrument's modules, the main window with its top-level System window is displayed. The System window (Figure 3) shows a graphic representation of the pattern generator. A single module instrument shows the module serial number, a text entry field to name/rename the instrument, and buttons to bring up the two main configuration windows of PGAppDotNet: the Program Window and the Setup Window (also a vestigial button to turn on and off the instrument that is no longer functional). A multiple module instrument adds slave module outlines to the main graphic, each labeled with a slave module serial number.

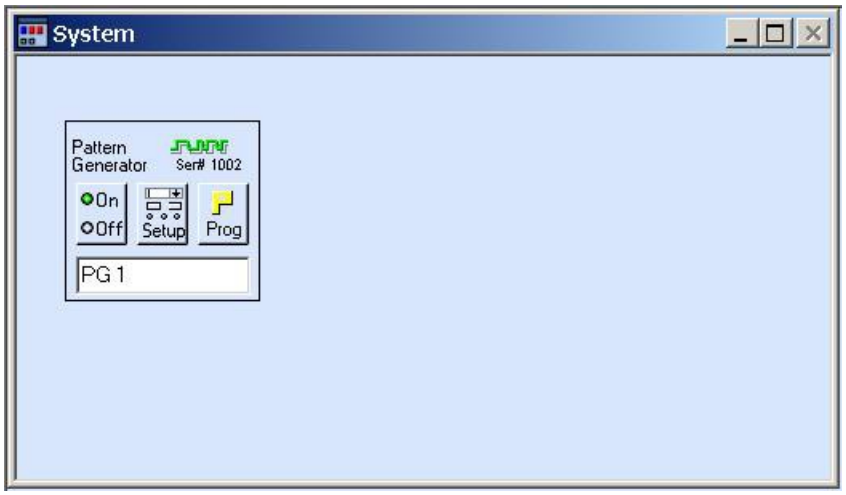


Figure 3 – System Window

In the legacy Tektronix PG application, the System Window had more functionality, potentially showing multiple instruments and allowing the user to switch among them. In PGAppDotNet, the System Window is static, and is mostly used to navigate to the Program and Setup windows.

Setup Window

Clicking on the **Setup** button in the System window brings up the Setup window. The Setup window is a tabbed window with four pages: Module Setup, Channel Setup, Probe Setup, and Signals Setup. While most setup configuration can be done at any time, channel configuration should be performed first before setting up the pattern generator

program in the Program window. The following sections describe each page in more detail.

4.5.3.1 Module Setup Page

The Module Setup page controls clock configuration, various event configuration options (but not event definition), and options controlling the behavior of the PG when running and stopped. Figure 4 shows an example of the Module Setup window and its fields are described below:

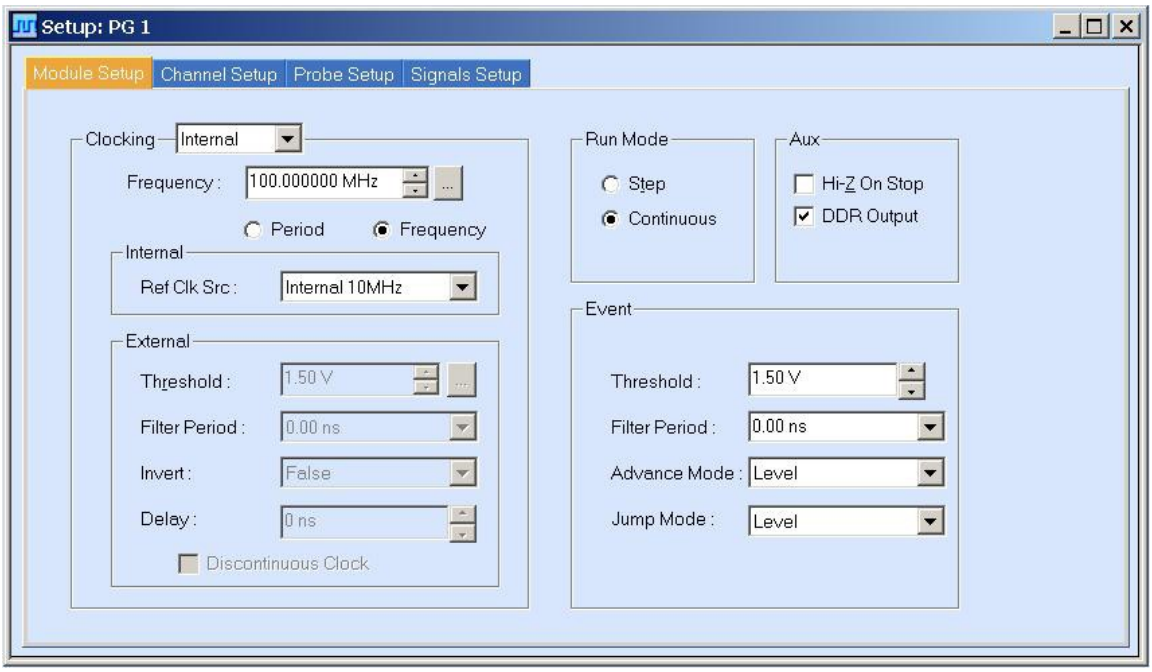


Figure 4 – Module Setup Page (Setup Window)

Clocking – Use the Clocking list-box to select whether you want internal or external clocking. Internal clocking means that the PG module generates the master clock. External clocking requires that the Inputs probe is connected to the master PG module and means that the Inputs probe clock is used as the master clock.

Period/Frequency – The period/frequency control sets the clock period or frequency, depending on which option box below it is selected. When typing directly into the control, you may follow the value with a units designation (hz, Khz, Mhz for frequency or ns, us, ms for period). Alternatively, you may use the up/down arrows to make fine adjustments or the ellipsis button to the right to bring up a popup menu of options. The supported frequency range of the PG3A module is 100 Hz to 300 MHz.

Please note that the PGAppDotNet still requires an approximate period/frequency setting *even if external clocking is used*[4]. There are certain optimizations and limitations the PGAppDotNet creates and enforces based on clock frequency.

[4] For a discontinuous clock, the maximum burst period/frequency should be entered

Ref Clk Src – When internal clocking is selected, the "Ref Clk Src" list-box is enabled. Use this control to select the reference source for the clocking system when using internal clocking. Options include Internal 10 MHz, External Ref Clk (which uses external REFIN connector), and Backplane Ref Clk (for master PG modules plugged into a TLA mainframe). Generally, the internal 10 MHz option is the appropriate selection.

Clock Threshold – When external clocking is selected, the clock "Threshold" control is enabled. This control sets the threshold voltage for the external clock input (Inputs Probe) with a valid range from -2.0V to 2.5V in 100 mV increments (but with ~2 mV quantization for values typed in directly).

Clock Filter Period – When external clocking is selected, the clock "Filter Period" control is enabled. This control adjusts the filter period for glitch filtering on the external clock input (Inputs Probe) with options of 0, 1, 4, and 10 ns.

Clock Invert – When external clocking is selected, the clock "Invert" control is enabled. When set true, this control inverts the external clock input (Inputs Probe) before it is used by the PG. In conjunction with the clock "Delay" control, this allows for a global data delay adjustment of at least one full clock period for all continuous frequencies greater than 1 MHz.

Clock Delay – When external clocking is selected, the clock "Delay" control is enabled. This control adjusts the delay of the external clock input (Inputs Probe) before it is used by the PG. In conjunction with the clock "Invert" control, this allows for a global data delay adjustment of at least one full clock period for all continuous frequencies greater than 1 MHz.

The delay range supported by the PG3A hardware depends on the input clock frequency. For clock frequencies less than 29 MHz, the delay may be set from 0 to 500 ns. For clock frequencies greater than or equal to 29 MHz, the delay may be set from 0 to 17.25 ns. Precision of delay specification is very fine, 10 ps, however absolute accuracy is more coarse. Fine adjustment of delay is intended to be a relative and not an absolute control.

Because of the different supported delay ranges depending on frequency, the hardware measures clock frequency to ensure settings remain within specification. This is also a useful check for the user if using external clocking, validating that the input clock is present and running correctly. The user is notified when the PG is running if the clock frequency measures outside a certain range centered around the declared frequency in PGAppDotNet.[5] The user is also notified when the clock period/frequency is changed in the Module Setup window such that the clock delay (or data delays in the probe window) is outside the new valid range.

[5] The range at high frequencies (> 20 MHz) is +/- 3% and expands toward lower frequencies to 50% below 1 MHz.

Lastly, the user should be aware that adjusting the input clock delay in real-time *when the PG is running* can have undesirable side effects. Because changes to the clock delay are asynchronous to the clock, it is possible to generate a glitch. If a glitch occurs, subsequent vectors output by the PG cannot be guaranteed to be correct (until the sequence is stopped and restarted). Thus, the real-time adjustment of clock delay is currently intended to be used for calibration purposes only (e.g. to align edge positions relative to reference signals when using a looping test sequence).

We are working to improve this situation. Please call us to discuss your critical application needs for glitch-less, run-time, clock delay adjustment.

Discontinuous Clock – When external clocking is selected, the "Discontinuous Clock" check-box is enabled. Checking this box notifies PGAppDotNet that the external clock input (Inputs Probe) is not continuous. This option disables notification of measured clock frequency discrepancies and also prevents the inversion of the probe output clocks for centered data clocking by the SUT.

This means that the user is responsible for managing setup/hold timing for the output data as it reaches the SUT. Depending on the setup/hold requirements of the SUT, options include continuing to use the non-inverted output clock, using a clock from the SUT, or using a delayed clock from an unused data probe, to clock PG data in to the SUT.

Run Mode – Set the run mode to "Step" to output the pattern vectors one at a time. Use the Step button in the Status Monitor window to manually generate one clock cycle. Set the run mode to "Continuous" to output all vectors at the selected clock rate.

Hi-Z on Stop – Check the "Hi-Z on Stop" box to cause probe data and strobe outputs to go to a high-impedance state when the PG is stopped.

DDROutput – Check the "DDROutput" box to cause the PG to output data on both clock edges. This feature is enabled via a separately purchasable option, i.e. a license string in the file TMPCLicense.txt in the PGAppDotNet directory. If the license option is present and the firmware version of all modules is at least 1.4 then the DDR Output checkbox is enabled. See section 4.6.7 for more details about DDR Output operation.

Event Threshold – The event threshold control sets the threshold voltage for the external event inputs (Inputs Probe). The valid range is -5.0V to 5.0V in 100 mV increments (but with ~2 mV quantization for values typed in directly).

Event Filter Period – The event filter control adjusts the filter period for glitch filtering of event equation evaluation. Valid options are 0, 25, and 50 ns.

Advance Mode – Selects how WaitFor events are processed, either Level or Edge (see section 4.6.4 on event processing).

Jump Mode – Selects how branch events are processed, either Level or Edge (see section 4.6.4 on event processing).

4.5.3.2 Channel Setup Page

The Channel Setup page functions like Channel Setup in the TLA application. Use this window to define the channel group names, the logical grouping of channels, and the individual channel names. Figure 5 shows an example of the Channel Setup page.

The Channel Setup page is divided into two panes: the group definition pane (top-half) and the channel definition pane (bottom-half). The group definition pane lets you enter group names and make channel assignments to each group. The channel definition pane lets you assign logical names to probe channels and see which channels are currently unassigned and which channels are assigned to the current group.

It is very important that some careful thought be given signal groupings at the beginning of the configuration process. Once groups are established, they cannot be changed without rendering the data defined for those groups useless. It is possible to render all entered data useless!

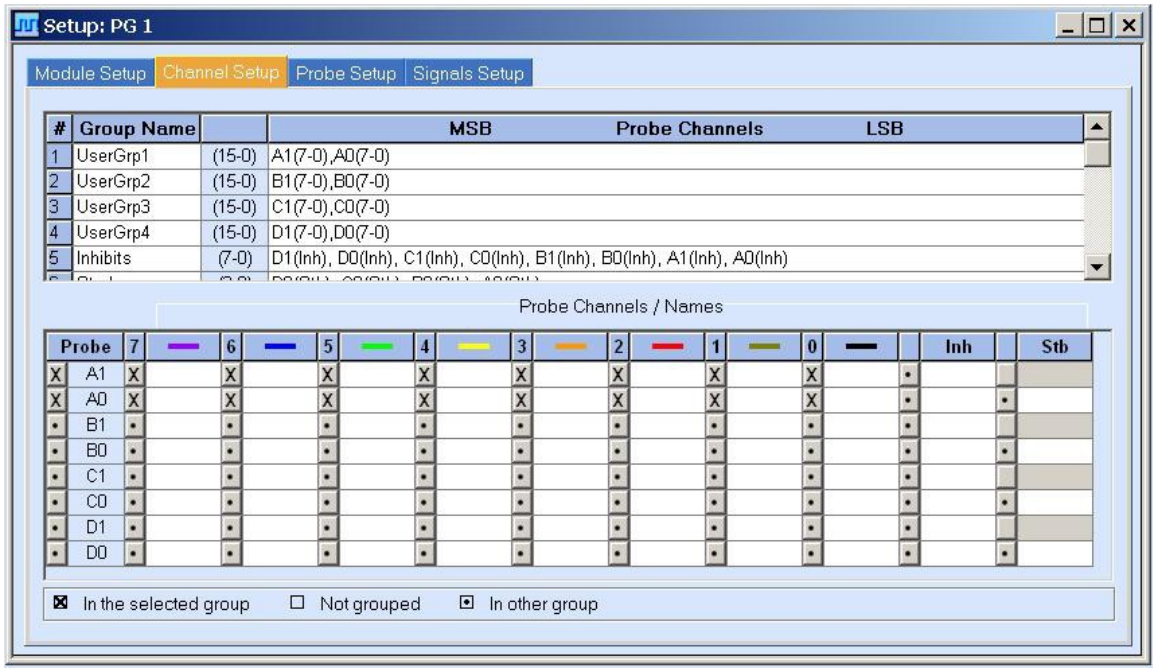


Figure 5 – Channel Setup Page (Setup Window)

Channel References

Channels may be referred to by their physical name or logical name (if assigned). The physical name of a data channel is constructed as follows:

1. the probe letter (A, B, C, or D)
2. followed by the byte lane (1 or 0)
3. followed by the channel index in parentheses (0, 1, 2, 3, 4, 5, 6, or 7)

e.g. A0(6), D1(0). To refer to a range of data within a byte-lane of a probe, you may replace the single channel index with a channel range of <msb> "-" <lsb> or nothing for a range of "7-0". Examples: A1(5-1), B0().

The physical name of an inhibit channel is constructed as follows:

1. the probe letter (A, B, C, or D)
2. followed by the byte lane (1 or 0)
3. followed by "Inh" in parentheses (not case sensitive)

e.g. A1(Inh), C0(Inh).

The physical name of a strobe channel is constructed as follows:

1. the probe letter (A, B, C, or D)
2. followed by "0"
3. followed by "Stb" in parentheses (not case sensitive)
   e.g. B0(Stb), C0(Stb), D0(Stb).

Logical names may be given to a probe channel by typing the name in the text field next to a channel in the channel definition pane. These names may then be used in lieu of the physical channel name in the channel assignment lists of the group definition pane.

Creating Groups

Default groups and channel assignments are made at startup (if a saved system is not automatically loaded) and when the File->Defaults menu option is selected. The default groups consist of four data groups (UserGrp1, UserGrp2, UserGrp3, and UserGrp4) assigned to the 16 data bits of each probe, an Inhibits group with eight inhibits (2 per probe), and a Strobes group with four strobes (1 per probe).

To change a group name, you may simply edit the group name field in the group definitions window. To create a new group, scroll to the bottom of the channel names and type in a new group name.

Assigning Channels

To assign a channel to a group in the group definitions window, simply list the physical or logical channel names (or channel ranges) separated by commas in MSB to LSB order. However, as channels may not be assigned to more than one group, you must first remove channels assigned to one group before assigning them to another.

Alternatively, you may click on individual channel buttons that are marked as "Not grouped" in the channel definition pane to add individual channel names to the current group list. In addition, buttons from the first column of the channel definition pane allow you to add/delete the corresponding row of data channels (not including inhibit and

strobe). These buttons are just a shorthand for clicking each data channel in the row to add them to or delete them from the current group.

Please note that when channel assignments are changed for a group, *existing program data defined for that group does not automatically transfer*. As the software was originally written, data is stored and associated with channels and not groups (which are treated merely as aliases for channels). Thus, it is important that you make the proper channel assignments to groups before defining any data, otherwise data is effectively lost when subsequently changing channel assignments.

4.5.3.3 Probe Setup Page

Use the Probe Setup page to specify the probe details such as the output voltage level, data delays, and strobe information. It also reads back the actual setting of a given parameter and the probe type. Figure 6 shows an example of the Probe Setup window.

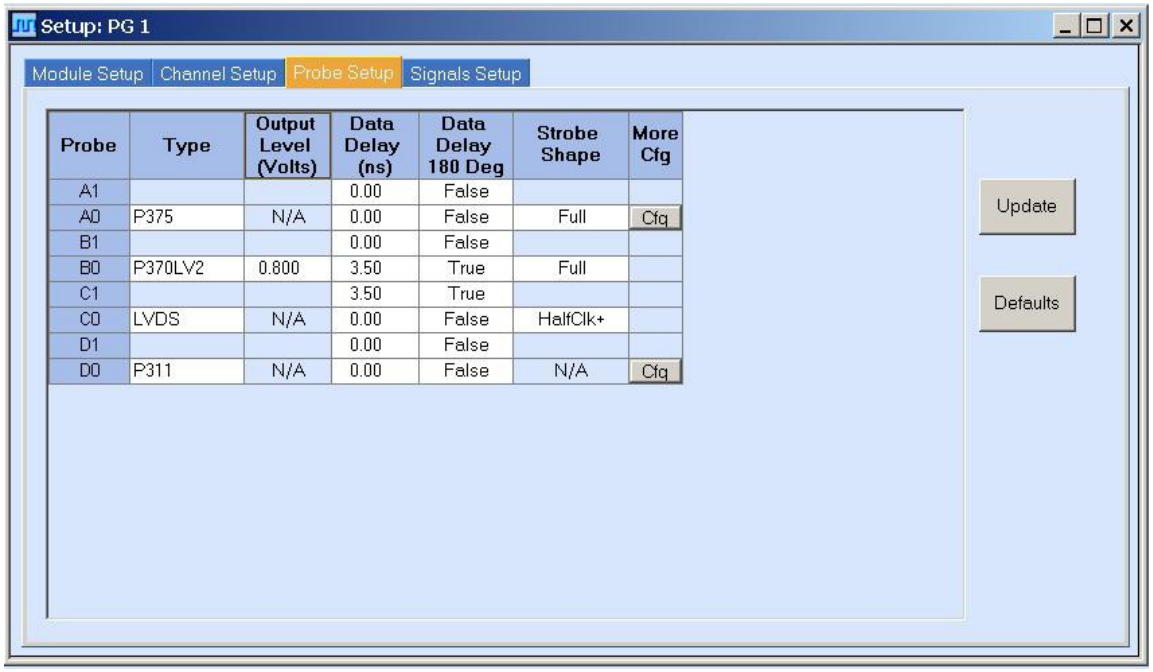


Figure 6 – Probe Setup Page (Setup Window)

Each row in the Probe Setup page corresponds to a byte-lane of a given probe. Thus, each pair of rows provides the controls available for one probe. Data delay controls are distinct for each byte-lane, thus they are enabled every row. However, there is only one control for output level and strobe shape and these controls apply to the entire probe. Note that all controls in the probe setup page remain active when the PG is running, allowing you to interactively adjust probe settings while you monitor output and the SUT. Probe Setup page controls are described below:

Update Button – For longest service life, you should **NOT** connect and disconnect probes while the PG is powered on. Nonetheless, there are times when probe configuration in PGAppDotNet does not match currently connected hardware, e.g.

possibly when a saved system is loaded. PGAppDotNet verifies that probe configuration matches the probes currently connected to the PG at the following times:

- when a saved system is loaded
- when the PG is run
- when the Probe Setup page is activated
- when a probe control is adjusted
- when the Update button in the Probe Setup page is clicked

If probe verification fails, the user is notified and asked either to go to the Probe Setup page and update configuration (first two cases) or if it is okay to automatically update the probe configuration (last three cases). Updating configuration consists of reading in any new probe types and resetting output levels to the minimum voltage of the probe's recommended range. Delay and strobe configuration remain unchanged.

Defaults Button – Clicking the Defaults button restores all probe settings to their defaults (note that the Default System menu option does *not* reset probe settings). For variable probes, this includes restoring their settings to defaults as well. The tables below document the default probe settings:

Table 1 – Probe Defaults

Output Level	minimum in-spec level for the given probe type.
Data Delay	0 ns
Data Delay 180 Deg	false
Strobe Shape	full

Table 2 – Variable Probe Defaults

Group	None
High Level	0 V
Low Level	0 V
Slew Rate	Normal
Differential	No
Delay	0 ns

Type – For offline modules, the probe type is configurable using the drop-down selection box in this field. Otherwise, the probe type for real PG modules is fixed. Simply select the probe type from the selection box provided. Note also that offline modules will automatically inherit the probe types when loading system files if the default option: "Offline Modules Inherit File Pod Types" is set to "Yes" (System->Options menu, defaults page).

Output Level – Output level specifies the output voltage level of the probe. The range supported depends on probe type. Additionally, for some probes, the output level is allowed to be set below its stated operating range. In this case, the user will be notified and the output level will be displayed in red, but the setting will be allowed. While probe behavior is not absolutely guaranteed at this setting, it still should prove useful to some customers who are limit- testing their SUT. Note that when holding a spin arrow control, the display updates continuously but the value is not sent to the hardware until the arrow is released.

Data Delay – Data delay adjusts the output timing of a probe data byte-lane. In, conjunction with the "Data Delay 180 Deg" control, this allows for a byte-lane data delay adjustment of at least one full clock period for all continuous frequencies greater than 1 MHz.

The delay range supported by the PG3A hardware depends on the input clock frequency. For clock frequencies less than 29 MHz, the delay may be set from 0 to 500 ns. For clock frequencies greater than or equal to 29 MHz, the delay may be set from 0 to 17.25 ns. Precision of delay specification is very fine, 10 ps, however absolute accuracy is more coarse. Fine adjustment of delay is intended to be a relative and not an absolute control. Note that when holding a spin arrow, the display updates continuously but the value is not sent to the hardware until the arrow is released.

Because of the different supported delay ranges depending on frequency, the hardware actually measures clock frequency to ensure settings remain within specification. At clock frequencies greater than 20 MHz, the user is notified when the PG is running if the clock frequency measures greater than +/- 3% from the declared frequency in the Module Setup page. Also, when the declared clock period/frequency is changed such that any of the data delays are outside the new valid range, the user is notified.

The user should be aware that when adjusting this control, if the user is monitoring the data and its associated clock signals, the user will not see any change in the relationship of these signals as the delay value is changed! The clock and data move together and change relative to all the other clocks and data bytes. To "see" the delay changing, the user will need to monitor the high byte in any probe relative to the clock OR monitor data bits in the low byte relative to the high byte OR any data relative to the clock or data in another probe.

Lastly, the user should be aware that adjusting the data delay in real-time *when the PG is running* can have undesirable side effects. Internally, data delays are implemented by delaying the clock associated with the data. Because changes to the data delay are asynchronous to the associated data clock, it is possible to generate a clock glitch. If a glitch occurs, subsequent vectors output by the PG cannot be guaranteed to be correct (until the sequence is stopped and restarted). Thus, the real-time adjustment of data delay is currently intended to be used for calibration purposes only (e.g. to align edge positions relative to reference signals when using a looping test sequence).

We are working to improve this situation. Please call us to discuss your critical application needs for glitch-less, run-time, data delay adjustment.

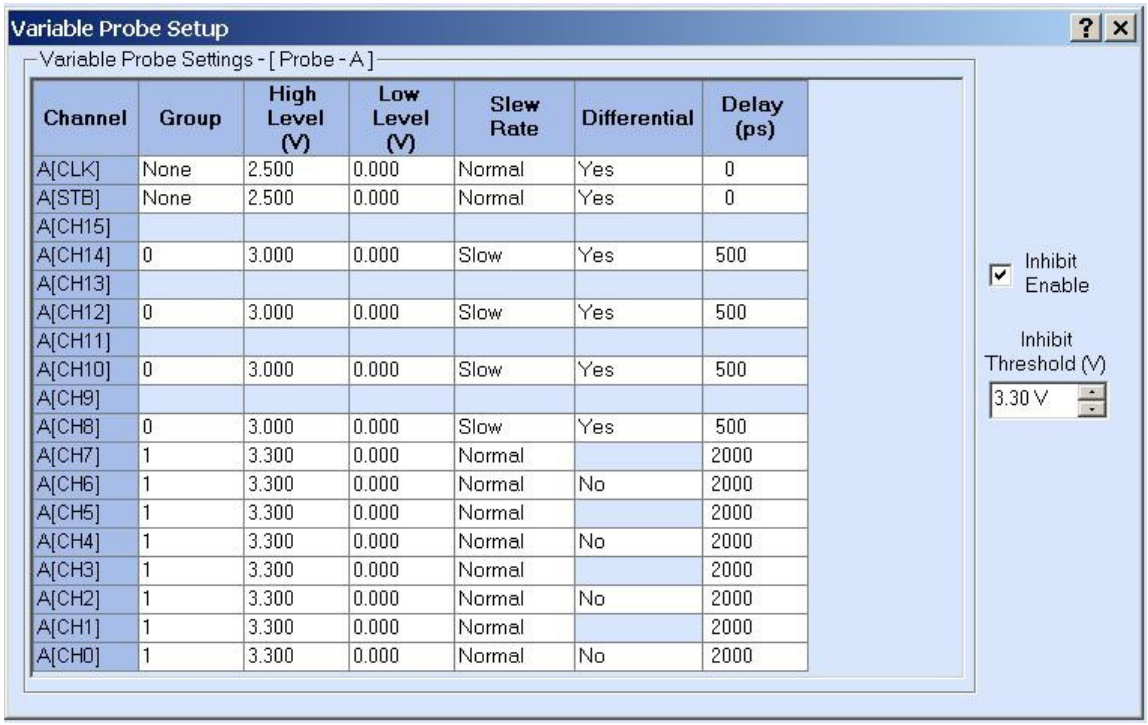
Data Delay 180 Deg – When set true, this control inverts the data clock used to output a probe data byte-lane. In, conjunction with the corresponding "Data Delay" control, this allows for a byte-lane data delay adjustment of at least one full clock period for all continuous frequencies greater than 1 MHz.

Strobe Shape – The strobe shape control selects a probe's output strobe shape pattern. See section 4.6.6 for more information.

More Config – This column provides a button for bringing up auxiliary configuration windows such as the Variable Probe Setup window for variable probes (e.g. P375) and the Dual-DAC Probe dialog for the P311. Only one instance of the Variable Probe Setup window remains open at a time. If the window is already open when a different variable probe button is clicked, the window will be reinitialized with the new probe's information.

4.5.3.3.1 P375 Variable Probe Setup Window

Figure 7 shows the Variable Probe Setup window that configures a variable probe such as the P375.



Channel	Group	High Level (V)	Low Level (V)	Slew Rate	Differential	Delay (ps)
A[CLK]	None	2.500	0.000	Normal	Yes	0
A[STB]	None	2.500	0.000	Normal	Yes	0
A[CH15]						
A[CH14]	0	3.000	0.000	Slow	Yes	500
A[CH13]						
A[CH12]	0	3.000	0.000	Slow	Yes	500
A[CH11]						
A[CH10]	0	3.000	0.000	Slow	Yes	500
A[CH9]						
A[CH8]	0	3.000	0.000	Slow	Yes	500
A[CH7]	1	3.300	0.000	Normal		2000
A[CH6]	1	3.300	0.000	Normal	No	2000
A[CH5]	1	3.300	0.000	Normal		2000
A[CH4]	1	3.300	0.000	Normal	No	2000
A[CH3]	1	3.300	0.000	Normal		2000
A[CH2]	1	3.300	0.000	Normal	No	2000
A[CH1]	1	3.300	0.000	Normal		2000
A[CH0]	1	3.300	0.000	Normal	No	2000

Figure 7 – Variable Probe Setup window

The P375 variable probe has 18 output signals: 16 data channels, a strobe, and a clock. Two inhibit inputs are also provided, which are enabled when the Inhibit Enable check

box is selected. The inhibit threshold voltage is then set using the Inhibit Threshold control.

Each channel may be independently configured with five configuration parameters – High Level, Low Level, Slew Rate, Differential, and Delay. These parameters plus the "Group" setting are described below.

Group – Each variable probe channel may be assigned to a group, which propagates parameter changes made to one channel in a group to all channels in the group (this includes parameter setting via PPI). Four groups are available 0-3 and "None". "None" is the initial default group assignment for all channels and does not propagate any settings.

High Level – This parameter sets the channel's output voltage level when output data equals 1. For the P375, the allowed voltage range is -2.2 to 6.5 V in 20 mV steps.

Low Level – This parameter sets the channel's output voltage level when channel data equals 0. For the P375, the allowed voltage range is -2.2 to 6.5 V in 20 mV steps

Slew Rate – The slew rate setting can select between "Normal" or "Slow". For the P375, these settings correspond to typical slew rates of 3500 V/us and 1750 V/us respectively.

Differential – The differential setting can select between "Yes" and "No" and is allowed for even data channels only (plus strobe and clock). If set to "Yes", data corresponding to the selected channel will be output as a differential signal on the even/odd channel pair (i.e. if <chan> is the even data channel, the positive differential component signal will output on <chan> and the negative differential component signal will output on <chan> + 1).

Delay – This parameter sets the channel's individual delay, which is an adjustment in addition to the probe's Data Delay setting on the Probe Setup Page. For the P375, the delay range adjustment is from 0 to 2.4 ns in 20 ps steps.

Note that delay adjustments to a channel in differential mode will maintain the current delta delay between channels and will limit the adjustment range accordingly.

For example, let's say Ch0 is set to non-differential mode and Ch0 has a delay of 100 ps, and Ch1 has a delay of 200 ps. If Ch0 is then set to differential mode and its delay is set to 500 ps then Ch1, which has become the negative differential channel to Ch0, will automatically be set to 600 ps. Moreover, the Ch0 delay maximum will now be 2300 ps (as opposed to 2400 ps in non-differential mode).

4.5.3.3.2 P311 Dual-DAC Probe Dialog

The Dual-DAC Probe Configuration dialog is shown in Figure 8.

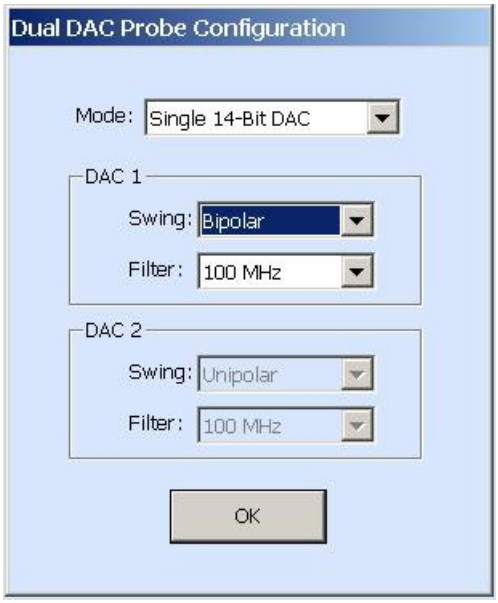


Figure 8 – Dual-DAC Probe Dialog

This dialog is modal and is opened when the "Cfg" button is clicked in the Probe Setup page for the P311 probe. The following fields are provided:

Mode – This option selects one of two modes of operation for the P311 probe. The first mode, "Dual 8-bit DAC", configures the probe to use both DAC outputs fed by the MS and LS bytes of its data stream. DAC1 is fed by data bits (7-0) and DAC2 is fed by data bits (15-8) of the probe. The second mode, "Single 14-bit DAC", configures the probe to use only DAC1, fed by data bits (15-2) of the probe.

Each DAC has two configurable options – Swing and Filter. In single-DAC mode, the dialog disables the options for DAC2.

Swing – This option determines the voltage levels of the DAC output. "Unipolar" sets the low output level to 0 volts and the high output level to 3 volts. "Bipolar" sets the low output level to -1.5 volts and the high output level to 1.5 volts.

Filter – This option selects the reconstruction filter used for the DAC output. Four choices are available: 300 MHz, 30 MHz, 3 MHz, 300 KHz.

4.5.3.3.3 P321 Serial Probe Support

While the P321 serial probe does not have its own setup window, PGAppDotNet does offer the Fill-From-File feature to help enter in serial data in the listing window. Please see Section 10 for more information on using the Fill-From-File function.

The P321 serial probe supports RS422/RS232, SPI, and I2C protocols. While the hardware actively drives all protocols simultaneously when the PG is running, only one will be correct depending on data content. It is assumed the user will make the relevant signal connections between the SUT and the serial probe, leaving other protocol signals unconnected. Note that, currently, only protocol writes are supported; no mechanism for protocol reads is implemented. The following channel assignments are assumed for the P321 probe:

RS422/RS232:
- Bit0: TxD
- Bit1: DTR
- Bit2: RTS
- Bit3: Disable TxD (tri-state TxD)
- Bit4: Disable DTR(tri-state DTR)
- Bit5: Disable RTS (tri-state RTS)

SPI:
- Bit0: SCLK
- Bit1: MOSI
- Bit2: GP0 (general purpose output)
- Bit3: GP1 (general purpose output)
- Bit4: GP2 (general purpose output)
- Bit5: GP3 (general purpose output)
- Bit6: GP4 (general purpose output)
- Bit7: GP5 (general purpose output)

I2C:
- Bit0: SCL
- Bit1: SDA

For simple applications of the serial probe, the user can define a single group of sufficient width to hold the relevant protocol bits. For example, the group SerBits defined with channels "C(7-0)" would work for all the above protocols (assuming the probe was connected to port C on the PG). "C(1-0)" would also work for both I2C and SPI, assuming none of the general-purpose outputs for SPI were to be used.

For RS422/RS232, the data compiler invoked via Fill-From-File always sets DTR and RTS high and clears the disable bits. For SPI, all general-purpose outputs are set to 0. Advanced users who want to set these bits differently should assign them to a different group, which can than be manually defined separately from the Fill-From-File operation (the data compiler simply drops MS bits from the definition if the fill-group is too narrow). For example, if you define group SerBits as C(1-0) and group GPBits as C(7-2)

using the SPI protocol, then you can fill the SerBits group from a protocol file and manually edit GPBits.

4.5.3.4 Signals Setup Page

Use the Signal Setup window to define input and output signal routing (see Figure 9). The following discussion only applies to the master module of the PG. Slave modules may be in stand-alone cabinets or plugged into a TLA without affecting signal routing options.

In addition to the 8 event signals from the Inputs probe (which must be connected to the master module), one additional signal source can be specified here in defining logical events for program flow control. For a stand-alone master PG module in a cabinet, only ExtTrigIn is available for this function. Additionally, for a master PG module plugged into a TLA mainframe, the four TLA backplane signals (Signal1-Signal4) are also available.

The sequence definition page defines a logical signal called SignalOut, where each sequence line (i.e. block) defines SignalOut as high or low. Here, in the Signals Setup page, SignalOut is routed to an output (and optionally inverted if the signal destination is ExtTrigOut). For a stand-alone master PG module in a cabinet, only ExtTrigOut is available for this function. Additionally, for a master PG module plugged into a TLA mainframe, the four TLA backplane signals (Signal1-Signal4) are also available.

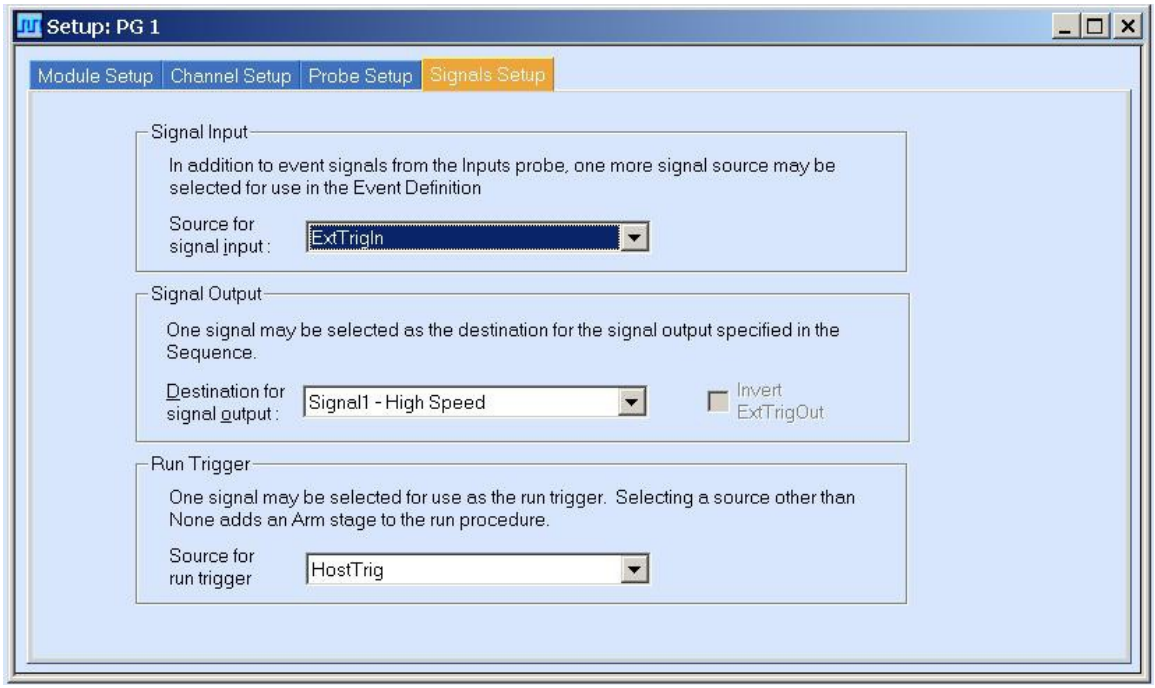


Figure 9 – Signals Setup Page (Setup Window)

The Run Trigger source is also selected in the Signal Setup window. The Run Trigger source defines whether or not the PG provides an Arm stage when run, and if so, which signal the PG should wait on before enabling the output clock (refer to section Error: Reference source not found for more details about arm/run timing). For a stand-alone master PG module in a cabinet, only ExtTrigOut and HostRunTrigger are available for this function. Additionally, for a master PG module plugged into a TLA mainframe, the four TLA backplane signals (Signal1-Signal4) are also available.

Program Window

Clicking on the **Prog** button in the System window brings up the Program window. The Program window is a tabbed window with three pages: Block Definition, Sequence Definition, and Event Definition. Because of the dependency of program data on group definitions, you should make sure channel configuration has been performed before setting up the pattern generator program.

Use the Program windows in the following sequence:

1. Use the Block Definition window together with the Listing or Waveform window to define the data blocks and the vectors in each block.

2. Use the Sequence Definition window to define a high-level sequence flow of the pattern generator program.

3. Use the Event Definition window to define how events are used with the pattern generator program.

The following sections describe each page in more detail.

4.5.4.1 Block Definition Page

Use the Block definition page to define blocks of output data. You can define the size of each block and assign each block a meaningful name (such as Init, Read Cycle, Interrupt). Figure 10 shows an example of the Block Definition window.

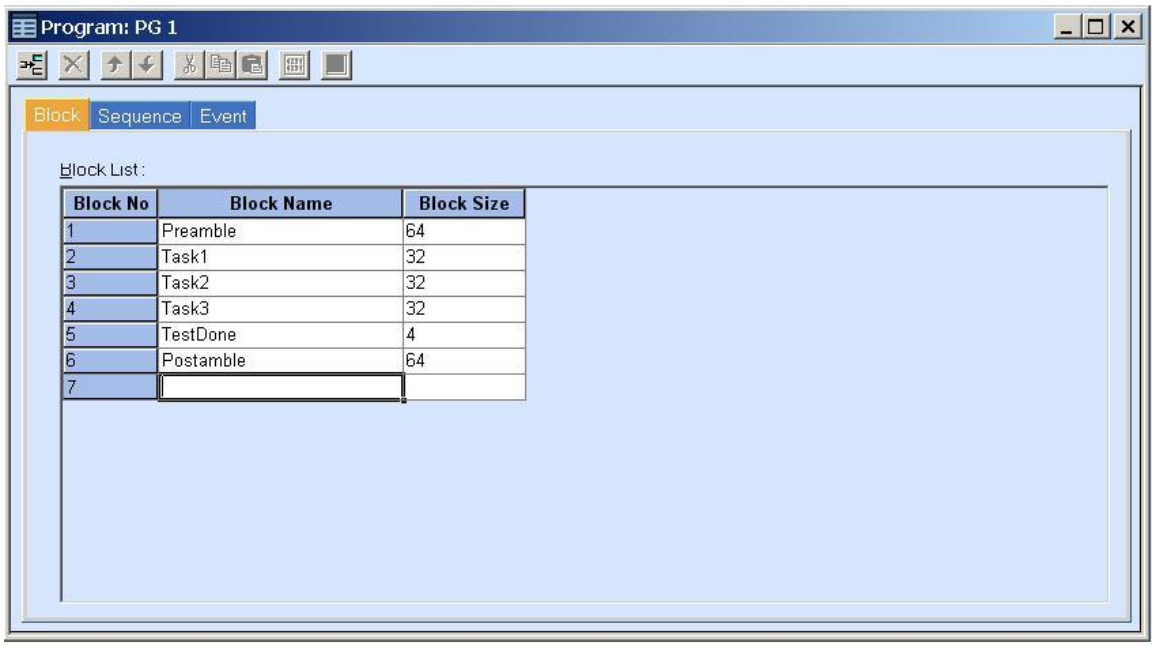
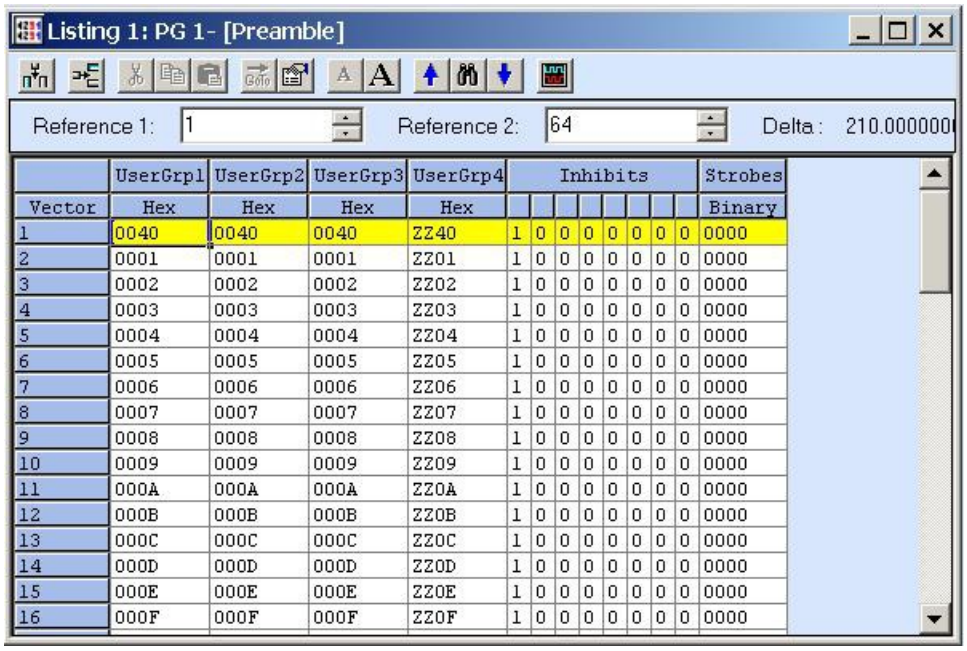


Figure 10 – Block Definition page (Program Window)

Each block has an associated Listing and Waveform window. The Listing window may be brought up via a button on the Program window or Waveform window toolbars (if enabled) or by double-clicking the block number in the Block Definition page. The Waveform window may be brought up via a button on the Program Window or Listing toolbars (if enabled).

4.5.4.1.1 Listing Window

Use the Listing window to define group data for a block. Data may be defined in a number of ways: editing cell data directly, cutting and pasting rows/cells/columns, filling with data patterns via context menus, or importing data from a file using the File->Import main menu option. Figure 11 shows an example of the Listing window.



	UserGrp1	UserGrp2	UserGrp3	UserGrp4	Inhibits								Strobes
Vector	Hex	Hex	Hex	Hex									Binary
1	0040	0040	0040	ZZ40	1	0	0	0	0	0	0	0	0000
2	0001	0001	0001	ZZ01	1	0	0	0	0	0	0	0	0000
3	0002	0002	0002	ZZ02	1	0	0	0	0	0	0	0	0000
4	0003	0003	0003	ZZ03	1	0	0	0	0	0	0	0	0000
5	0004	0004	0004	ZZ04	1	0	0	0	0	0	0	0	0000
6	0005	0005	0005	ZZ05	1	0	0	0	0	0	0	0	0000
7	0006	0006	0006	ZZ06	1	0	0	0	0	0	0	0	0000
8	0007	0007	0007	ZZ07	1	0	0	0	0	0	0	0	0000
9	0008	0008	0008	ZZ08	1	0	0	0	0	0	0	0	0000
10	0009	0009	0009	ZZ09	1	0	0	0	0	0	0	0	0000
11	000A	000A	000A	ZZ0A	1	0	0	0	0	0	0	0	0000
12	000B	000B	000B	ZZ0B	1	0	0	0	0	0	0	0	0000
13	000C	000C	000C	ZZ0C	1	0	0	0	0	0	0	0	0000
14	000D	000D	000D	ZZ0D	1	0	0	0	0	0	0	0	0000
15	000E	000E	000E	ZZ0E	1	0	0	0	0	0	0	0	0000
16	000F	000F	000F	ZZ0F	1	0	0	0	0	0	0	0	0000

Figure 11 – Listing Window

In Figure 11, note the effect an inhibit bit set high has on the data in UserGrp4. The data bits associated with the inhibit bits are displayed as 'Z' to show the bits will be set to a high-impedance state during output. Editing group data with inhibited bits is not allowed; you must first disable the inhibit bit before editing.

The Listing toolbar provides the following functions (left to right)

- Add column to display new group
- Insert new row
- Cut selected rows/columns
- Copy selected rows/columns/cells to the clipboard
- Paste clipboard contents to the selected rows/columns/cells
- Go to one of the reference rows, a particular vector, or group
- See or adjust the properties of the current group, specifically its radix, whether to expand it into individual bit columns (see Inhibits group in Figure 11), and whether it should be visible.
- Adjust the font size larger or smaller
- Search up/down a group for a particular value
- Bring up the Waveform window

Right-clicking in a column header brings up a context menu to do the following:

- Add column to display new group
- Copy column contents
- Paste clipboard contents to the selected rows/columns/cells
- Hide column

- Invert data bits
- Mirror data horizontally (msb <> lsb), or vertically in column (top <> bottom)
- Shift data in column up or down
- Rotate data in column up or down
- Fill column with data pattern (clock, pre-defined, user-defined, or from file[6])
- See or adjust the properties of the current group, specifically its radix, whether to expand it into individual bit columns (see Inhibits group in Figure 11), and whether it should be visible.

Right-clicking in a row header brings up a context menu to do the following:

- Move one of the reference marks to the current row
- Add column to display new group
- Copy selected rows to clipboard
- Paste clipboard contents
- Insert row

Right-clicking in a cell brings up a context menu to do the following

- Move one of the reference marks to the current row
- Add column to display new group
- Copy selected rows to clipboard
- Paste clipboard contents
- Insert row
- Go to one of the reference rows, a particular vector, or group
- Search up/down a group for a particular value
- See or adjust the properties of the current group, specifically its radix, whether to expand it into individual bit columns (see Inhibits group in Figure 11), and whether it should be visible.

4.5.4.1.2 Waveform Window

The Waveform window graphically displays groups and probe signals as waveforms, and provides another method for data entry and visualization (see Figure 12). Double-clicking on a position in a group waveform highlights the position with vertical and horizontal blue bars and brings up a data entry box to set the value. Double-clicking on a position in a probe signal waveform highlights the position and toggles the value there (0 to 1, 1 to 0).

[6] See Appendix C for documentation on filling a data column with data from a file.

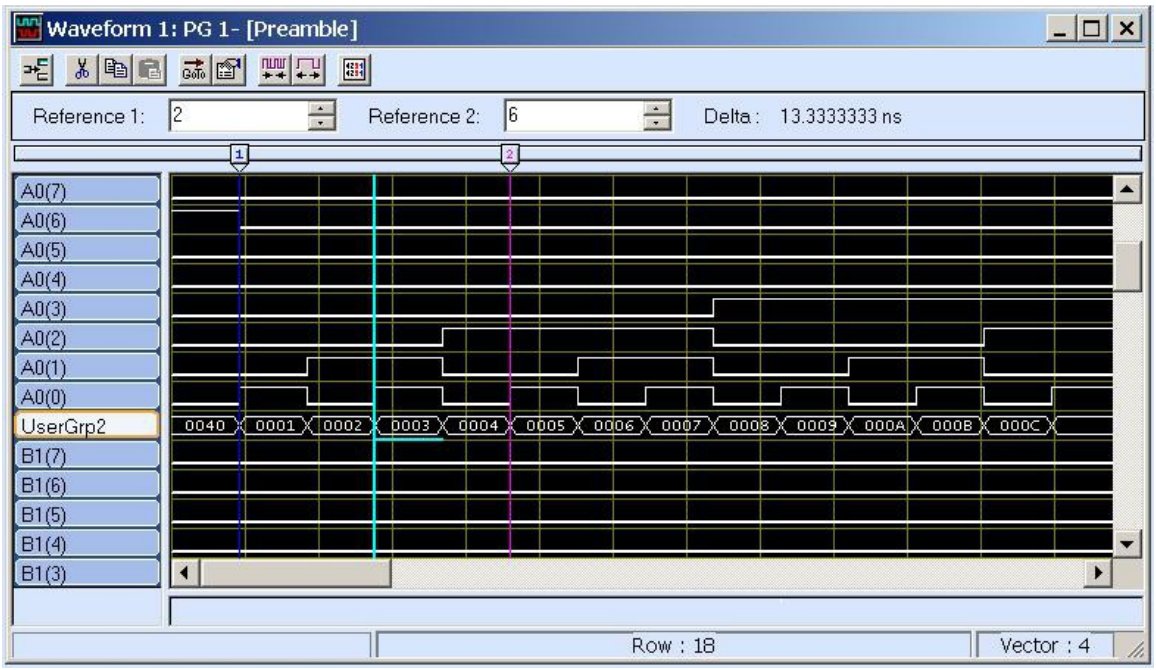


Figure 12 – Waveform Window

Probe signals that have their inhibits set high are displayed as dashed lines and can't be edited until their inhibits are set low. Similarly, groups that have at least some of their bits inhibited are displayed with a 'Z' (all bits in nibble are inhibited) or '?' (some bits in nibble are inhibited) and cannot be edited until all bits in the group are uninhibited.

The Waveform toolbar provides the following functions (left to right)

- Add waveform to display new group or probe signal
- Cut selected row
- Copy selected row
- Paste row
- Go to one of the reference rows, a particular vector, or waveform
- See or adjust the properties of the current group, specifically its radix and whether it is visible
- Adjust the waveform scale shorter/longer (i.e. zoom in/out)
- Bring up the Listing window

Right-clicking in a row header brings up a context menu to do the following:

- Add waveform to display new group or probe signal
- Cut selected row
- Copy selected row
- Paste row
- Hide row
- Invert data bits
- Fill waveform data with pattern

- Go to one of the reference rows, a particular vector, or waveform
- See or adjust the properties of the current group, specifically its radix and whether it is visible.

Right-clicking in a cell brings up a context menu to do the following

- Adjust the waveform scale shorter/longer (i.e. zoom in/out)
- Go to one of the reference rows, a particular vector, or waveform
- Add waveform to display new group or probe signal
- Move one of the reference marks to the current row
- See or adjust the properties of the current group, specifically its radix and whether it is visible.

4.5.4.2 Sequence Definition Page

After defining the data blocks, use the Sequence Definition window to create a high-level overview of the pattern generator program. Figure 13 shows an example of the Sequence Definition window.

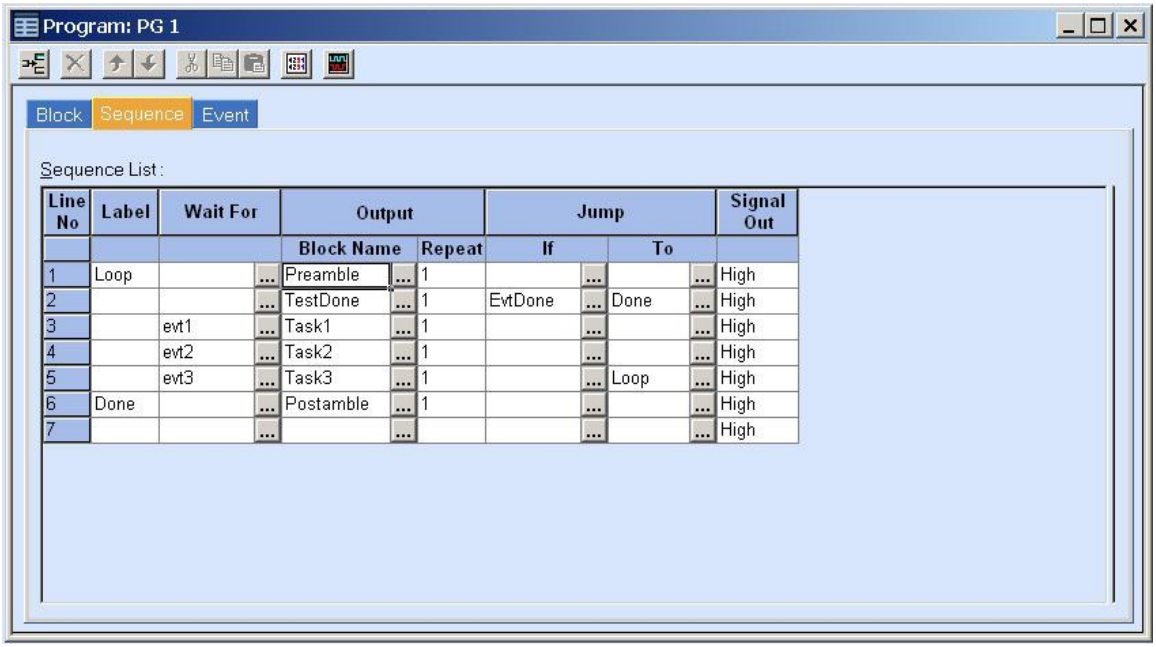


Figure 13 – Sequence Definition page (Program Window)

The sequence definition tells the PG the order in which blocks should be output and how many times blocks should be looped, defines flow control conditions (stalls, jumps, and branches), and defines an output signal based on the current sequence line being played.

When the PG is run, each sequence line directs the PG to output a block defined in the Block Definition window. Starting with line one, each sequence line is played in order, unless otherwise directed by flow-control logic (which may cause a stall, jump, or a conditional branch to a different sequence line). When the last sequence line has been played, if it ever does, program flow stops.

Note that a graphical representation the logic for a single sequence is displayed to the right of the sequence list but is not visible with the default Program window width. To see it, expand the Program window by left-clicking the right edge and dragging it to uncover the graphical display. The appearance of the sequence flow depends on the sequence definition and each sequence line has its graphic

The fields of a sequence line are described below:

Label – identifies branch/jump destinations or can be used as a descriptive comment field.

WaitFor – if not empty, specifies a logical event to use for flow control, instructing the sequencer to wait for the event to become true before outputting the block. If the WaitFor event is false when the sequencer evaluates it, the last vector from the preceding block is repeatedly output until the WaitFor event becomes true (see section 4.6.4 on Events for more information). *Note that a WaitFor event specified for a looping block is evaluated each time through the loop (this is different than the Tek PG which evaluated the WaitFor only once)*.

If you want, use the ellipsis button to the right of the WaitFor field to select from a drop-down list of defined events.

Block Name – indicates the block to output and can be either typed in directly or filled in using the block selection dialog by clicking on the ellipsis button to the right of the Block Name field.

Repeat – specifies the number of times to loop a block during output. The limit is one MIterations or infinite (enter zero).

JumpIf – if not empty, specifies a logical event test to use for conditional branching. The event can be either typed in directly or filled in using the JumpIf selection dialog by clicking on the ellipsis button to the right of the JumpIf field. If the branch event is true when the sequencer evaluates it, program flow switches to the sequence line named in the JumpTo field after an iteration of the block is played. Otherwise, program flow continues on to the next sequence line. *Note that a branch event specified for a looping block is evaluated each time through the loop.*

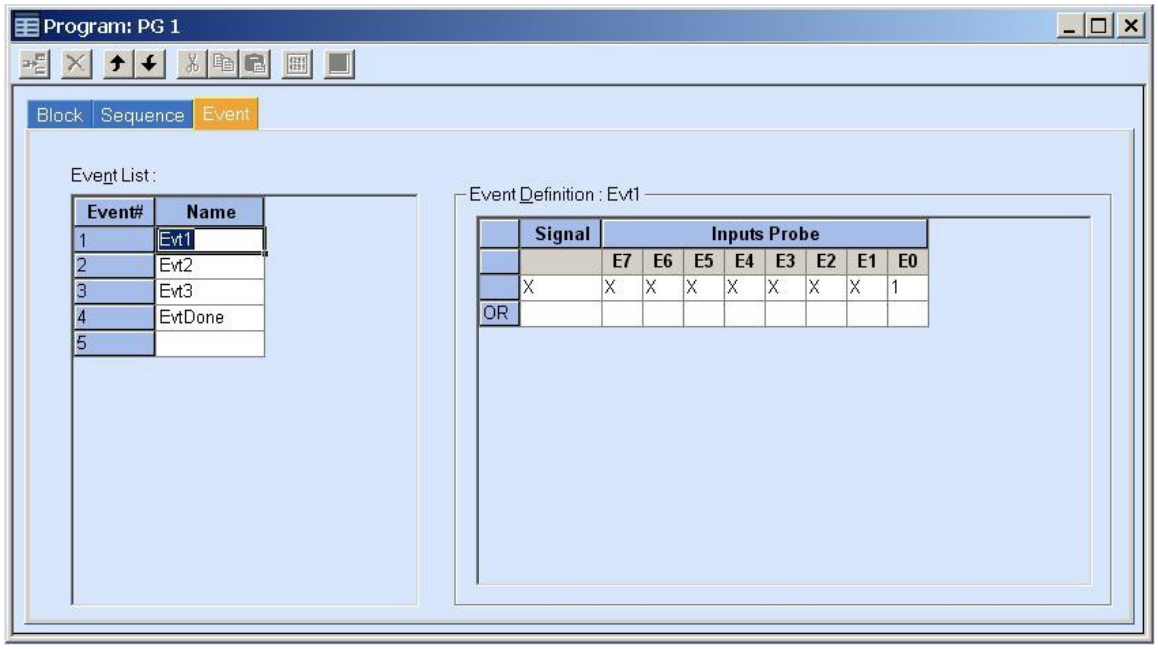
JumpTo – if not empty, specifies a jump or conditional branch destination (i.e. a sequence line label). The destination can be either typed in directly of filled in using the JumpTo selection dialog by clicking on the ellipsis button to the right of the JumpTo field.

SignalOut – defines a signal state (low or high) for the SignalOut user signal while the sequence line is playing. The routing for SignalOut is defined in the Signal Setup page of the Setup window. In addition, the signal may be optionally inverted if the selected destination is ExtTrigOut. If SignalOut is routed to one of the TLA backplane signals,

the state "High" corresponds to "True" and "Low" corresponds to "False" when setting up trigger conditions on the TLA.

4.5.4.3 Event Definition Page

Use the Event Definition window to define named logical events to be used in the Sequence Definition page. Figure 14 shows an example of the Event Definition window. The events in each row are logically ANDed together while the rows are logically ORed together. Event signals may have values '1', '0' or 'X' (for don't care).



Event#	Name
1	Evt1
2	Evt2
3	Evt3
4	EvtDone
5	

	Signal	Inputs Probe							
		E7	E6	E5	E4	E3	E2	E1	E0
	X	X	X	X	X	X	X	X	1
OR									

Figure 14 – Event Definition Page (Program Window)

Enter an event name in the left side of the window and then define events in the right side of the window. The event labeled "Signal" is the signal input source you selected in the Signals Setup page (for example, TrigIn). The events labeled E7 through E0 are the external input event signals from the Inputs Probe, which must be connected to the master module of the PG instrument.

Status Monitor Window

The status monitor shows the current state of the PG and provides buttons for the user to control program flow. Figure 15 shows an example of the Status Monitor window.

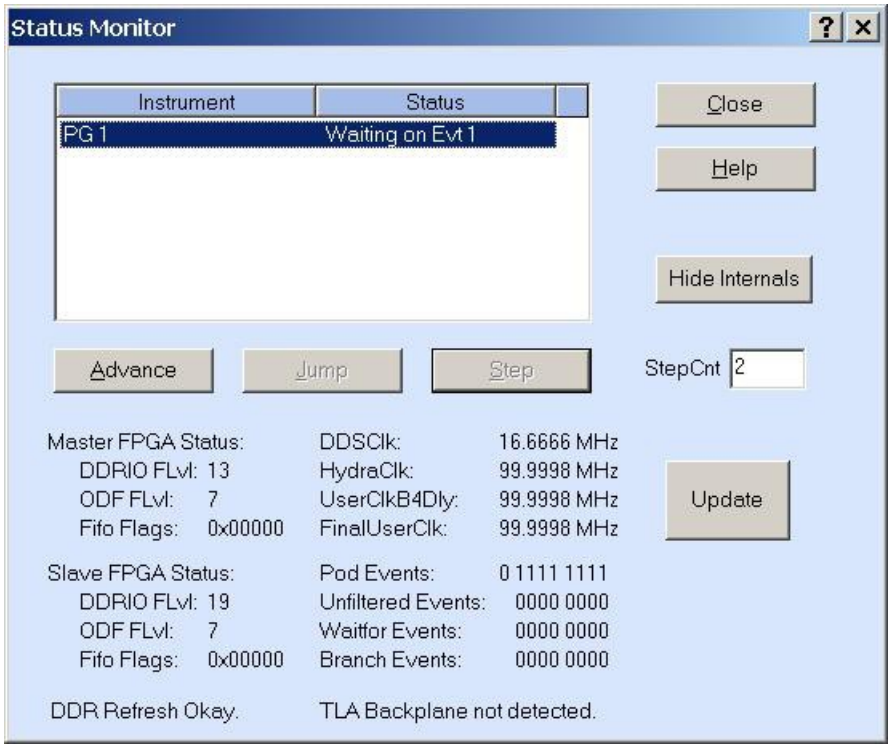


Figure 15 – Status Monitor Window

The top pane in the status monitor shows the PG instrument and its run status: Idle, Waiting, Running, or Stepping:

> **Idle** – the PG is not running; probe output clocks are set low and data outputs are set to high-impedance if the "Hi-Z on Stop" option is checked in the Module Setup page
>
> **Waiting** – the PG's pipeline is stalled waiting for the indicated logical event. This event is one that was specified in the WaitFor field of a sequence line. In this case, the last vector from the previous block is repeatedly output until the event evaluates true.
>
> **Running** – the PG is actively outputting vectors and proceeding through the sequence.
>
> **Stepping** – the PG is outputting a vector while in step-mode and is waiting for the step button to be pressed to output the next vector(s). Also displayed in parentheses is the number of vectors stepped since the PG was run.

Three buttons below the status pane control flow-control of the PG program:

> **Advance** – if the PG is waiting for an event, clicking on the "Advance" button enables the PG to continue immediately.

Step – if the PG is in step-mode (configured as the run-mode in the Module Setup page), the clock will be held constant until the Step button is clicked. When the step button is clicked, the output clock transitions based on the step count field, stepping through vectors in the sequence.

The units of the step count field are **vectors**, which equates to clock **edges** or **cycles** depending on whether the PG is outputting vectors in DDR mode. (The PG outputs vectors in DDR mode if the PG option for DDR output is present and the DDR Output check-box in the Module Setup page is checked. See section 4.6.7 for more information.

Note that clock transitions are controlled from software and when the step count field is greater than one occur at approximately 200 us per transition.

Jump – One possibility when the PG is running is that the PG is looping on a block infinitely waiting for a branch event. In this case, clicking on the "Jump" button will cause the branch to be taken at the end of the next iteration.

The bottom half of the status monitor window displays hardware internals. Toggle the "Show/Hide Internals" button to reveal and hide them as desired. Some internals such as the master/slave FPGA status readouts are not relevant to the user. However, values that might be of interest are:

HydraClk – shows the measured input clock frequency from the Inputs probe if clocking is set to External or the internally generated clock frequency if clocking is set to Internal (Module Setup page).

FinalUserClk – shows the measure output clock frequency. This will be 0 Hz if the PG is not running since the clock is turned off at this time. Otherwise, when running, the FinalUserClk value should be equal to the HydraClk value.

PodEvents – shows the activity and level of the event signals from the Inputs probe as well as the signal input source defined in the Signals Setup page. The display value is in the same order as they are shown in the Event Definition page (i.e. Signal, E7, E6,… E0). Activity is shown with a '+'; static levels are shown with a '0' or '1'.

WaitFor Events – shows the activity and level of the WaitFor logical event register. The bits of this register are tested for WaitFor event flow control. Thus, these bits follow event filtering and using the Advance Mode Level/Edge detection specified in the Module Setup.

Branch Events – shows the activity and level of the Branch logical event register. The bits of this register are tested for Branch event flow control. Thus, these bits follow event filtering and using the Jump Mode Level/Edge detection specified in the Module Setup.

When the PG is running, the status monitor is updated automatically (once every ½ second). However, when the PG is not running, the status monitor is not automatically updated, as most fields are irrelevant when the PG is stopped. However, the HydraClk measurement and the Pod Events display might still be useful when the PG is stopped, and so the Update button is provided for the user to refresh the status monitor display.

4.6 PG Concepts

Merging Modules

While one PG module provides 64 channels of output data, up to four PG modules can be connected together to form an instrument with up to 256 channels. Sometimes this is referred to as "merging" modules. An instrument comprising more than one module has one module designated as a master (P1) and the rest as slaves (P2, P3, P4). This designation is determined by your ordering of modules in the Connection dialog.

To run the PG as an instrument with merged modules, perform the following steps:

1) Power off PG modules.
2) Connect a USB cable from your computer to each PG module. If necessary, you can use a USB hub.
3) Connect the modules together using a P390 Hydra cable. The P390 has one large 68-pin master connector and four cables emanating from the master connector each terminating in a 36-pin slave connector. The master connector plugs into the "Hydra Out" port of the master module and a slave connector should be plugged into the "Hydra In" ports of each of the slave modules. In addition, a slave connector must be plugged into the master module "Hydra In" port.
4) Power on PG modules.
5) Use the connection dialog (see section 4.5.1) to select which modules to merge into the instrument. The order of modules selected here determines which data channels map to which physical module. Module P1 will be associated with channels 1A, 1B, 1C, 1D. Module P2 will be associated with channels 2A, 2B, 2C, and 2D. And so on.

The merged instrument is now ready for use. The System window shows the instrument with an outline for each merged module, including serial number. Additional channels defined in the channel setup window for you to include in group definition. The System Properties window shows multiple modules and their properties. Otherwise, operation is the same as for a single-module instrument.

Please note that only the TrigIn, TrigOut, RefIn, RefOut connectors on the master module are properly active in the merged instrument. Additionally, only the inputs probe connected to the master module can be used for events and external clocking.

Offline Operation and Offline Modules

Offline modules are place-holders for real-hardware. They are always an available option for master/slave modules in the PG connection dialog and may be mixed in with real PG modules in defining an instrument. With the exception of simulating running the PG, offline modules otherwise simulate the presence of a real PG module to allow program definition, debugging, and PGAppDotNet demonstration.

While offline modules do not simulate running the PG, they are benign as slave modules when the master PG module is real. Thus, for example, you could load and run a 4-module system into an instrument configuration having one real PG module as master and three offline modules as slaves.

On the other hand, if the master module of a PG instrument is an offline module, the instrument is considered to be in Offline mode. In this case, running the PG is not an option and the Run button is grayed (even if one or more slave modules are real). However, pressing the grayed run button does compile the current program and report any errors or warnings.

Unlike the legacy Tektronix PG application, offline modules are not configurable using .ini files. Instead, offline modules have configurable probe types (Probe Setup page) and automatically inherit probe types when loading system files if the default option: "Offline Modules Inherit File Pod Types" is set to "Yes" (System->Options menu, defaults page). This allows the loading, viewing, editing, and saving of system files having any instrument configuration.

Output Clocking

Each probe has one output clock but two delay controls, one for each byte-lane of the 16-data bits. The clock that is output is the one that corresponds to the least-significant byte-lane (i.e. D0-D7). Note that strobe timing is also coincident with this clock.

Now, since this clock is synchronous with D0-D7 (and, indeed, in the common case when all delays are set to zero or the same value), it is hard to guarantee any reasonable setup/hold timing to the SUT (as the clock edge will be near the data transition). Accordingly, the PG inverts the output clocks of all probes as a matter of course, to provide maximal data setup/hold timing windows to the SUT.

However, in the case of a discontinuous clock, this method presents a problem, since there is no guarantee when (or whether) a falling edge will occur, and inverting it will not have desired behavior. *Thus, in the discontinuous, external clocking case, probe output clocks are not inverted*. In this case, this means that the user cannot use the probe output clock to clock data into the SUT and must either use a clock sourced from the SUT with a measured phase relationship to the output data, or an appropriately delayed output clock from a different probe (whose data byte is unneeded or at least insensitive to arbitrary delay) to clock data in to the SUT.

Events

An event is a logical signal derived from a sum-of-products Boolean combination of nine input event signals. The eight external input events from the Inputs Probe are always available plus one of potentially five other possible events: ExtTrigIn and four TLA backplane events (only if the master module is plugged into a TLA). Which of the five is used is determined by the Signal Input configuration on the Signal Setup page. Up to 8 distinct events can be named and defined in the Event page of the Program window.

Events can be referenced in the **WaitFor** and/or **JumpIf** fields of a sequence line and are the mechanism for flow control when the PG is running. An event specified in the **WaitFor** field of a sequence line that evaluates false when processed will stall output (i.e. play the last vector of the previous block) until the event becomes true. An event specified in the **JumpIf** field of a sequence line that evaluates true when processed will branch to the sequence line specified by the **JumpTo** field. If it is necessary to distinguish in which field an event is referenced, the terms WaitFor and branch events shall be used respectively.

While waitfor and branch event processing share many stages (including event signal conditioning, logical event computation, and filtering), at the last stage they diverge, where each type is configured individually for use in **Edge** or **Level** mode.[7] Level events are the simpler of the two types, where the Boolean expressions that define them are evaluated at the time they are needed. In contrast, edge events are detected and latched when the Boolean expression that defines them changes from false to true. This latched value remains held until the event is successfully tested; then it is cleared.

While a single event can be used as both a waitfor and branch event, be aware that the setup time requirements for each type are different as they are evaluated at different stages of the output pipeline. Also, it is not recommended to use an event this way if both waitfor and branch events are configured for **Edge** mode, since two distinct event registers are used to latch edge events. While the branch may clear the event bit in the branch event register, it will remain set in the waitfor event register until it too is successfully tested.

Inhibits

The inhibit function on a probe sets a byte lane of its output to a high-impedance state (thus a 16-bit probe generally has two inhibit signals). The inhibit function of a probe is controlled when the PG is running in two ways:

1. Through data inhibit bits in the output vector. For the 64-bit PG3A module, 8 data inhibits are provided for the user to programmatically control data buses to tri-state at clock granularity during output.

2. Through input inhibit signals in the probe. The SUT can choose to inhibit data by asserting a signal on probe inhibit inputs, forcing data output into high-impedance for as long as the probe inhibit signal remains high.

Note that data inhibits and probe inhibits are ORed in the probe such that either function will tri-state the output data of the probe. Also, the "Hi-Z On Stop" check-box in the Module Setup page acts as a global inhibit function for all probes when the PG is not running.

[7] see the "Advance Mode" and "Jump Mode" fields of the Module Setup page in the Setup window

Data inhibit bits are defined in PGAppDotNet exactly as any other data bit. They are all assigned to a default group called "Inhibits" at startup (but, of course, may be reassigned if desired). Note that the legacy Tek PG application requirement that D channel data bits be used for data inhibits is no longer applicable.

Strobes

A strobe is an output signal that can be used as an enable or gated clock by the SUT. For the 64-bit PG3A module, 4 strobes are provided under program control, one per 16-bit probe.

Data that controls the strobe is defined exactly as other output data. They are all assigned to a default group called "Strobes" at startup (but, of course, may be reassigned if desired). Note that the legacy Tek PG application requirement that D channel data bits be used for strobes is no longer applicable.

Strobes operate differently than data in that, during output, when strobe bits are '1', a pulse is output according to the configured strobe shape. When strobe bits are '0', the default level is output (low or high depending on strobe shape). Strobe shapes are selected in the Probe Setup page and can have the following types: Full, HalfClk+, HalfClk-, InvFull, InvHalfClk+, and InvHalfClk-.

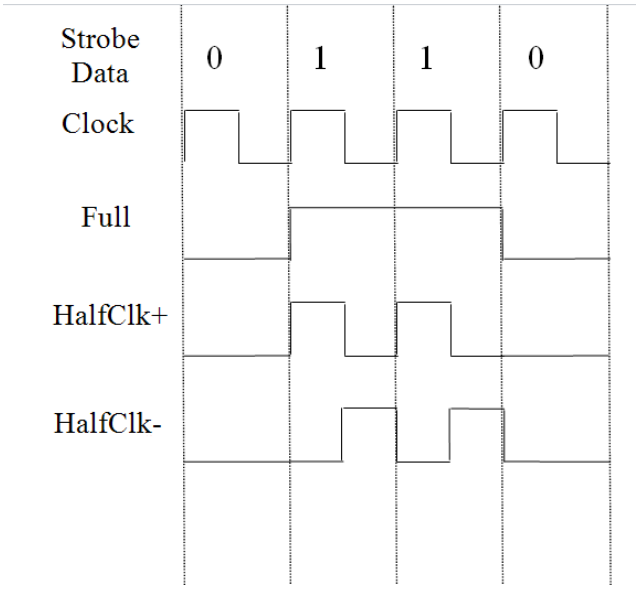


Figure 16
shows strobe shapes and their output would look given the data pattern '0110'.

DDR Output

As of PGAppDotNet release 2.0.013 and firmware release 1.4, the PG supports a licensable option to output vectors on both rising and falling edges of the clock. This mode is called DDR Output mode and is enabled by a license string in the file TMPCLicense.txt in the PGAppDotNet directory. Contact the Moving Pixel Company for more information about purchasing this option.

When the DDR Output option is enabled, the Module Setup page enables the "DDR Output" checkbox. Checking this box enables vectors to be output both on rising and falling clock edges when the PG is run. The following points should be noted when operating in DDR Output mode:

1) When PGAppDotNet downloads vectors into the PG modules, vector memory from alternating probe channels are used to store the falling edge vector data. This means vector data defined for probe B and D channels *will not be output* (instead, the data output on probe A is duplicated for probe B and data output on probe C is duplicated for probe D). Note that this process does not affect the user block and sequence definitions at all.
2) All blocks must be at least 8 vectors in length.
3) All blocks must have an even number of vectors.
4) If the setup/hold time from the clock accompanying the DDR data does not meet the user's system requirements, an alternative is to construct a clock with arbitrary phase relative to the DDR data by using the clock from one of the alternate (unused) probes, setting the DataDelay and DataDelay180 values as appropriate.

Branch Restrictions

A branch is defined to be a conditional jump in the sequence, specified when both the **JumpIf** and **JumpTo** fields are entered in a sequence line of the Sequence Definition page. This is distinguished from an unconditional jump when only the **JumpTo** field is entered. Due to implementation constraints in the hardware pipeline architecture, branches (but not jumps) have two usage restrictions in a sequence definition:

1. The number of branch destinations allowed in a sequence (i.e. unique **JumpTo** entries) is 64. Note that multiple branches to the same destination only count as one against the limit.
2. A second branch may not be specified within 12 vectors of a branch destination. This is called the branch hold-off requirement.

The compiler will flag errors at run-time if these conditions are not met.

To illustrate the branch hold-off requirement, consider the following sequence:

Label	Block Name	Repeat	JumpIf	JumpTo
Line1	Block1	Infinite	Evt1	Line2
Line2	Block2	Infinite	Evt2	Line3
Line3	Block3	1		

Block1 loops until Evt1 is true, then the sequence branches to Line2. Block2 loops until Evt2 is true, then the sequence branches to Line3. Since Line2 is a branch destination, the branch hold off restriction applies and no branch may be specified for at least 12 vectors. Thus, if Block2 is 12 vectors or longer, the compiler will allow it. Otherwise, an error will be flagged.

4.7 Running the PG

Putting it all together, a typical PGAppDotNet session begins with the following steps:

1) Launch PGAppDotNet, either explicitly connecting to PG module(s) through the connection dialog or implicitly connecting to PG module(s) from the previous session.
2) Open the Channel Setup page. Define and name logical groups, assigning probe channel bits to each group. These groups are presumably relevant to the specific SUT, i.e. Address, Data, Control, etc.
3) Optionally, open the Signals Setup page to configure the routing for the external event signal source and SignalOut signal destination.
4) Use the Block Definition page to create blocks of data vectors. For each block, name it, give it a length, and use the Listing or Waveform window to enter its data.
5) Optionally, use the Event Definition page to define logical events for sequence flow control.
6) Use the Sequence Definition page to set up the program for block output, looping, waiting, and branching as appropriate. If SignalOut is to be used, set its state during the output of each sequence line.
7) Now, configure the PG module using the Module Setup page. Define clocking and event parameters and PG behavior when running and stopped.
8) Finally, use the Probe Setup page to configure the PG probes, specifically output level, data delays, and strobe shape.

Once the program has been defined and the hardware configured, clicking the Arm/Run button on the main window toolbar kicks off the PG.[8] The following tasks occur during run:

Compilation – First, the compiler is invoked to process the block, sequence, and event definitions and check for errors. Progress dialogs are brought up to show what phase of compilation the compiler is in. When complete, the Output window is brought up to list any warnings and errors that occur during compilation.[9]

Download – If no errors occur, then sequence and vector data are downloaded to the PG module, with status shown via a progress dialog. Download time is variable depending

[8] The Run button is labeled "Arm" if a run trigger is specified in the Signals Setup page of the Setup window and "Run" otherwise.

[9] Note that if the Output window was brought up automatically, the Output window is also dismissed automatically if no errors occur. To bring up the Output window manually, for example to see warnings, you can manually bring up the Output window using the View->Output menu option.

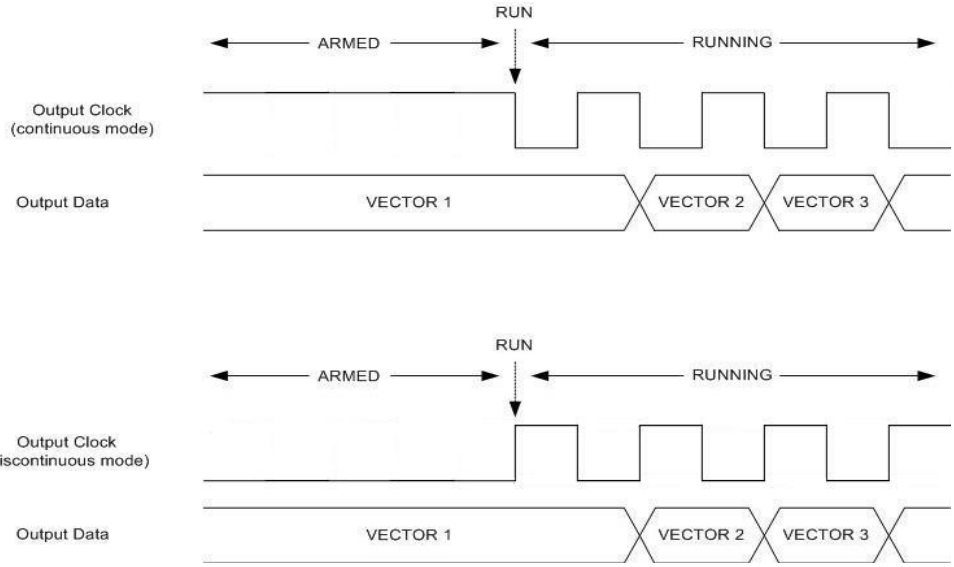
on the number of data vectors. The current download speed is roughly 20 seconds per 1 million vectors. Note that for subsequent runs, if no changes have been made to block and sequence data, download does not reoccur.

Arming – If the Run Trigger source in the Signals Setup page (Setup Window) is set to "None", this stage is skipped and the PG moves to the output stage. Otherwise, the PG enters the armed state and waits for the specified run trigger to occur (any occurrences of the run trigger prior to reaching this stage are ignored).

The arm state is such that the first vector of the sequence is asserted at the probe outputs but the output clock is quiescent.[10] Then, when the run trigger source asserts, the output clock is enabled and the PG will begin sequencing. One option for the run trigger is "HostTrig", which is sent by the user by pressing the "Run" button when the PG is armed. Moreover, the host trigger can be used as an override of any other run trigger source, forcing the PG from the armed state to output.

Output – Once data has been downloaded to the PG module(s), and the PG has been armed, the PG begins to output vectors. Up until this time, the output clock has been held quiescent. When the PG is triggered, the clock is allowed to transition and behaves as documented in Figure 16[11]:

Once running, the Status Monitor window can be used to see and control sequencer and instrument state (see section 4.5.5 for more information). Vectors are output until the last sequence line in the sequence has been played (or the Stop button is pressed).



[10] When the PG is armed but not running, the output clock may be high or low depending on the Discontinuous Clock setting in the Module Setup Window. Normally, the output clock is inverted to maximize setup and hold timing relative to data output. Thus, data will transition with the falling edge of the output clock and the output clock will be high when armed. However, if the Discontinuous Clock setting is true, the output clock is not inverted. Data will transition with the rising edge of the output clock and the output clock will be low when armed.

[11] This discussion assumes the DDROutput checkbox of the Module Setup page is not checked.

Figure 16 – PG Output Start Timing
(assumes data delays are zero)

When the PG has finished outputting the sequence, the last vector of the last block is held at the PG outputs until PGAppDotNet shuts off the output clock (up to ½ second later). Thus, it is important to ensure the last vector in your sequence is benign with respect to the SUT.

4.8 Importing and Exporting Data

Use the "Import Data" and "Export Data" file menu options to save/load block data to/from a file. These menu options become enabled only when a Listing window is active. (Note that an another load data alternative is to use the Fill-From-File context menu option to load vectors into a single group. See section 10 for more information.)

Exporting Data

To export data from PGAppDotNet, you can choose which columns of data to output by adding/hiding groups in the Listing window (using Edit menu or right-click context menu). Then select the "File->Export Data" menu option to output the block data to a file. Fill in the export range options, choosing from: all vectors, vectors between references, or a specified vector range. Select the output file format (either Tek TLA Data Exchange Format, or SynaptiCAD Test Vector Spreadsheet Format) and click the Save button. Example file formats below are shown below:

```
[vectors]
Sample[]        UserGrp1[15:0](Hex)     UserGrp2[15:0](Hex)     UserGrp3[15:0](Hex)
        UserGrp4[15:0](Hex)     Inhibits[7:0](Hex)      Strobes[3:0](Hex)       Timestamp[]
0       1111    5555    9999    DDDD    00      0       0
1       2222    6666    AAAA    EEEE    00      0       10.0000000 ns
2       3333    7777    BBBB    FFFF    00      0       10.0000000 ns
3       4444    8888    CCCC    0000    00      0       10.0000000 ns
```

Example file: Tek TLA Data Exchange Format

```
[Clocks]
Name                    Period  Offset  Duty    Invert
CLK1                    1 0     50      0
[Vectors]               End = 0
Absolute                Relative        UserGrp1[0:15]  UserGrp2[0:15]  UserGrp3[0:15]  UserGrp4[0:15]
0                       0 1111 5555     9999    DDDD    00      0
1                       1 2222 6666     AAAA    EEEE    00      0
2                       1 3333 7777     BBBB    FFFF    00      0
3                       1 4444 8888     CCCC    0000    00      0
[End]
```

Example file: SynaptiCAD Test Vector Spreadsheet Format

Importing Data

To import data into a block, use the File->Import Data menu option, which brings up the Import Data dialog. Browse for and select the file to import, select the file type (either

Tek TLA Data Exchange Format, SynaptiCAD Test Vector Spreadsheet Format, or Agilent Pattern Generator format), and click the Open button.

One common use for the import data function is to transfer data acquired from the Tek TLA. To export data from the TLA, the user can select the groups to export by shift-clicking on the column headers in a listing window (or leave all columns unhilighted to export all groups). Then, select the "File->Export Data" menu option to output the selected data to a file. Enter the file name, fill in the export range options, and save the file as type "Tektronix TLA Data Exchange Format".

If all group names and widths in the file match the current system configuration, loading proceeds normally. If the file contains a subset of groups in the current system configuration, only those groups defined in the file are overwritten. However, the length of the block is always changed to reflect the imported data block (thus truncating or extending groups that don't exist in the imported file). Note that the Sample and Timestamp groups in the Tek TLA Data Exchange Format are ignored.

If even one group name or width in the import file doesn't match the current system configuration, a warning will pop-up to notify you that:

"This operation will remove all the existing group information and create new groups. Do you want to proceed?"

It is important to note that this will affect all blocks not just the current block being imported into. This is equivalent to editing the channel setup after creating block data and means you must edit all extant blocks to accommodate the new group/channel configuration. If you are sure you want to continue, click Yes.

When creating new groups from an import file, channels are assigned automatically in ascending channel order to groups defined in the file. Currently, inhibit and strobe channels are never auto-assigned (since PGAppDotNet does not know for sure which groups are intended to be assigned to inhibits and strobes). Thus, in files that have inhibit and strobe groups defined, two potential error cases can occur:

1. If the inhibit and strobe groups are defined last in the import file, then all the preceding data groups will be successfully auto-assigned before a message appears indicating that too many channels are defined in the file. In this case, the inhibit and strobe groups are not imported.

2. If the inhibit and strobe groups are defined before other groups in the input file, then they will receive auto-assigned data channels and some of the last groups will not be defined and have their data imported.

In both cases, after importing, default groups of Inhibits and Strobes will be created with zero data.

In conclusion, if the auto-assignment behavior of PGAppDotNet is not what you want, go to the Channel Setup page and manually define your groups and channel assignments before importing. Then, no auto-assignment will occur during import.

4.9 Menus

This section documents the options available in the main menus. Note that the main menu options change depending on which window is active in PGAppDotNet.

File Menu:

Choose PG… – brings up the selection dialog to connect to a new PG instrument configuration. See section 4.5.1 for more information.
Default System – deletes setup and program data, restoring the default system configuration. The user is notified of this and given the option to cancel if he wants to save the current system configuration first.
Load System – loads a saved system file (.tpg extension) which contains all setup and program configuration. A system file may or may not contain data, but in any event any existing data is lost. Accordingly, the user is notified and asked whether to load the new system without saving the current system first (or he may cancel).
Save System – saves the current configuration automatically to the last system file loaded, assuming no intervening Default System or a connection to a different PG instrument configuration has occurred. Otherwise, this menu option behaves like the "Save System As…" menu option.
Save System As… – saves the current configuration to a designated file.
Export Data – brings up the export data dialog to export a block of data to a file. This menu option is only available if a Listing window is active. See section 4.8 for more information.
Import Data – brings up the import data dialog to import a block of data from a file. This menu option is only available if a Listing window is active. See section 4.8 for more information.
Page Setup… – brings up page setup dialog for setting print options such as printer, paper size, tray, orientation, and margins.
Print… – brings up print dialog for printing. Options for what to print depend on which window is active. If the system window is active
<most recently used system files> – loads the selected recent system file
Exit – exits PGAppDotNet. The user is asked whether to save system settings before exiting.

System Menu

Run/Stop System – runs the PG. Equivalent to clicking the Run/Stop button on the system toolbar.
Status Monitor – brings up the Status Monitor window to view PG state
System Properties – displays PG instrument information, i.e. number of channels, max clock speed, memory depth, serial numbers, firmware versions, and license options of PG modules.

Options… – brings up dialog to set PGAppDotNet color scheme, startup system, and other defaults such as radix, font size, channel value, whether to show status monitor automatically on run, and whether offline modules should inherit pod types from loaded system files.

View Menu

Toolbar – show/hide the main window toolbar
Statusbar – show/hide the main window statusbar
System – make the System window active
Output – make the compiler Output window active

If Listing window is active…
Listing Toolbar – show/hide the Listing window toolbar
Reference Toolbar – show/hide the Listing window reference toolbar
Smaller Font – decrease the Listing window font size
Larger Font – increase the Listing window font size
Waveform – bring up a Waveform window for the current block

If Waveform window is active…
Waveform Toolbar – show/hide the Waveform window toolbar
Reference Toolbar – show/hide the Waveform window reference toolbar
Measurement Toolbar – show/hide the Waveform measurement toolbar
Zoom In – decrease the Listing window font size
Zoom Out – increase the Listing window font size
Graticule – show/hide the Waveform window graticule
Listing – bring up a Listing window for the current block

Window Menu

Delete Window – deletes the active window. This menu option is only enabled when the Program, Setup, Listing, or Waveform windows are active.
System – makes the System window active
Next Setup – makes the Setup window active
Next Program – makes the Program window active
Cascade – arranges open windows in a stacked but staggered ordering
Tile Horizontally – arranges open windows in a horizontally tiled ordering
Tile Vertically – arranges open windows in a vertically tiled ordering
Minimize All – minimizes all open windows
<open window> – makes the specified window active

Help->Help Topics – brings up the on-line help
Help->Help On Window – brings up on-line help on the active window
Help->What's This ? – turns cursor into arrow-? for the user to click in windows and controls to bring up context specific help

Help->Technical Support – displays information on how to contact The Moving Pixel Company for questions and technical support
Help->Customer Feedback – information on how to provide customer feedback
Help->About PGApp – PGAppDotNet version information

4.10 Legacy Tektronix PG Comparison

The code for PGAppDotNet originated from the Tektronix Pattern Generator application "TLAPG", and indeed has an almost identical user interface and look-and-feel. However, there are salient differences between the two applications for several reasons:

- New specifications and capabilities of fully redesigned PG modules/probes
- Improvements and requested features from users, FAEs, and salespeople
- A more independent operating relationship between the PG and TLA
- Time constraints for initial implementation

It is important to note that PGAppDotNet will load legacy saved system and module files from the Tektronix Pattern Generator application. Simply use the File->Load System option in all cases. If the selection is a single-module system file or a module file, it will load automatically. If the selection is a multiple-module system file, a dialog will appear showing the module names within the file to allow you to select which module to load into the current system.

This following list compares TMPC's initial PG product offering (i.e. PGAppDotNet version 2.0 with the PG3A module) with the last release of the Tektronix Pattern Generator application (version 1.8) and their PatGen module (no longer available):

Maximum Clock Speed
TMPC: 300 MHz (with no half-channel mode)
TEK: 268 MHz in half-channel mode or 134 MHz in full-channel mode.

DDR Output
TMPC: vectors can be output in DDR mode (with license option)
TEK: no DDR output option

Memory Depth
TMPC: 32 MVectors
TEK: 256 KVectors or 1 MVectors with memory option

Module Data Outputs
TMPC: 64 data, 4 strobes, 8 inhibits (1 per 8 bits)
TEK: 64 data, 4 strobes, 4 inhibits (1 per 16 bits)

Strobe Shapes
TMPC: full-pulse, half-pulse (clk+), half-pulse (clk-), plus inverses
TEK: probe specific, RZ/R1 options with clock inversion

Inhibit/Strobe Progamming

TMPC: inhibits/strobes are programmed independently from data

TEK: inhibits/strobes were shared with probe D data

Inhibit by Event

TMPC: no support for a global data inhibit based on event

TEK: an event could be configured as a global data inhibit

Clock/Strobe Output

TMPC: clock and strobe are output simultaneously on probes

TEK: only one of clock or strobe could be enabled at a time on probes

Module Merging

TMPC: up to 4 modules may be merged to form 256-channel instrument

TEK: up to 5 modules could be merged to form 320-channel instrument

Multiple Module Control

TMPC: Multiple instances of PGAppDotNet per TLA/PC. One instance per instrument.

TEK: One instance of Tek PG application per TLA. .

Standalone Operation

TMPC: PG3A module may be plugged into TLA or into stand-alone cabinet.

TEK: Tek PG module needed to be plugged into TLA.

Minimum Block Size

TMPC: 4 vectors

TEK: 40 vectors

Sequence / Block Limits

TMPC: 3964 sequence lines / 3964 blocks

TEK: 4000 sequence lines / 4000 blocks

Block Looping Limits

TMPC: 1 M or infinite

TEK: 64 K or infinite

Clock / Event Filtering

TMPC: 4 clock filter settings up to 10 ns, 3 event filter settings up to 50 ns

TEK: no clock filter settings, 2 event filter settings up to 50 ns

Input / Output Signals

TMPC: ExtTrigIn, ExtTrigOut, RefClkIn, RefClkOut,
four TLA backplane signals if master plugged into TLA

TEK: four TLA backplane signals

Delay Adjustments

TMPC: input clock delay and eight output data delay adjustments (per byte)
TEK: input clock delay (skew adjustment), no data delay except in variable probe

Subsequences

TMPC: subsequences are not supported in PGAppDotNet initial release 2.0
TEK: supported subsequences, i.e. named groups of blocks that could be looped

Step Mode

TMPC: step mode is supported, allows stepping arbitrary number of clocks and provides total clocks stepped since start.
TEK: single step mode is supported

Run-time Control

TMPC: User can change clock/event threshold and filter settings, clock/data delays and probe output levels while running.
TEK: No dynamic control of hardware while running.

Run-time Status

TMPC: Clock frequency measurement and notification, event input signal and event register display in status window.
TEK: No clock/event signal status.

Offline Mode

TMPC: Offline modules are configurable in PGApp. Can mix offline modules with real hardware. Run simulation is not provided.
TEK: Offline modules are fully configurable via .ini files. Run simulation is provided.

5 Updating the PG Firmware

Field update of the PG3A firmware is achieved using the PGFWUpdate.exe utility. This utility is updated with every firmware version so be sure to download the latest version of it when you download the firmware. Its default location is in the "c:\Program Files\TMPC\PGAppDotNet" directory. Before running PGFWUpdate you will need the latest PG3A firmware file provided by TMPC, which can be downloaded on-line from TMPCs website at www.movingpixel.com/PGUpgrade.html. Note that the current PG firmware version used by your PG module can be determined by running PGAppDotNet and selecting the "**System->System Properties**" menu option in PGAppDotNet.

IMPORTANT: We strongly recommend that the computer being used to reload the PG firmware not be used for any other task during the firmware load process. If the firmware load process is interrupted, it is possible that the PG may have to be returned to the factory for firmware reload. The same is true if the process is interrupted due to a power failure.

Next, make sure no PGAppDotNet is connected to the module to be updated, and run PGFWUpdate.exe. A popup dialog will appear for you to select the serial number of the PG module to update. After this is done, the main window comes up (see Figure 17):

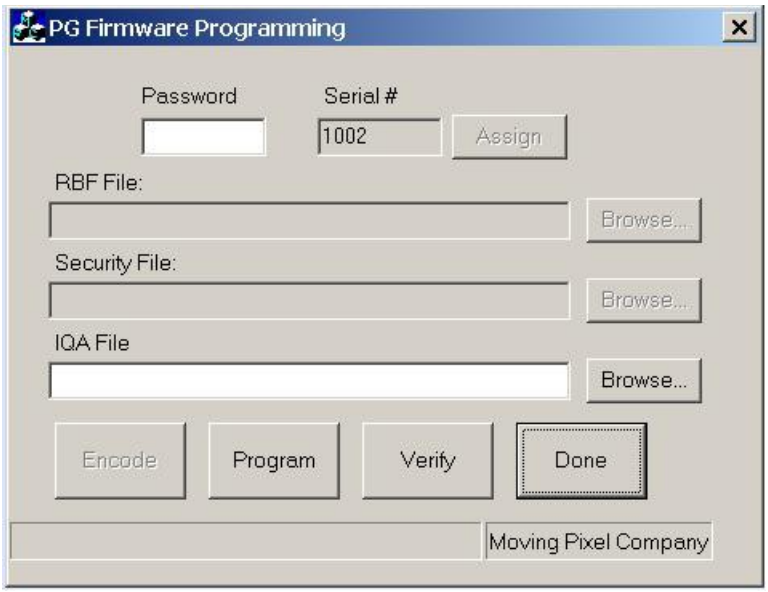


Figure 17 – PGFWUpdate.exe

Users do NOT need a password to upgrade! While this dialog supports password protected features such as programming serial numbers and encoding firmware files, the only option available to users is programming and verifying the PG3A firmware. To

update firmware, type in or browse for the .iqa firmware file and click the Program button. The status bar at the bottom of the window will update the progress of the firmware download (about 5 minutes). Click the 'Done' button to close the dialog when programming has completed.

You must now power cycle the TLA or cabinet (for stand-alone modules) before reconnecting via PGAppDotNet.

6 Probe Notes

For best operation, keep the cooling slots on the sides of the probe clear.

CAUTION:
The bottom of the probes have exposed electrical connections. If an electrically-conductive object were to touch one of these exposed connections, mis-operation or damage to the probe could occur. Operation on a smooth, hard surface is recommended.

Also note that all the probes except the P300 can be stacked on top of each other. Metal inserts are provided in each side of the probe for assisting in probe mounting. The screw thread is a UNC #4-40. Any screw thread inserted into this should not exceed 0.25 inches in length else the probe could be damaged.

6.1 P300 Inputs probe

The P300 Inputs probe provides a method for communicating user clock and user event information to the PG module. The P300 is not necessary for operation of the unit. Further, a user need not even connect the P300 to the PG module. Note that in the Quick Start (4.3 above), there is no mention of the P300.

The Inputs probe supports both a differential clock input and a single-ended clock input. The selection of the clock source is via a jumper accessible from the bottom of the probe.



Figure 18 P300 Inputs Probe

Each clock also has a selection of terminated or unterminated, again jumper-selectable from the bottom of the probe. The termination value for the differential probe is 100 ohms and the single-ended one is 50 ohms. However, these termination resistors are also accessible to the user from the bottom of the probe. They are in standard 0603 packages. The user can desolder the one present and replace the termination resistor with the desired value.

J1 1-2 connects the differential termination.
J2 1-2 connects the single-ended termination
J3 1-2 selects differential clock; 2-3 selects single-ended clock.

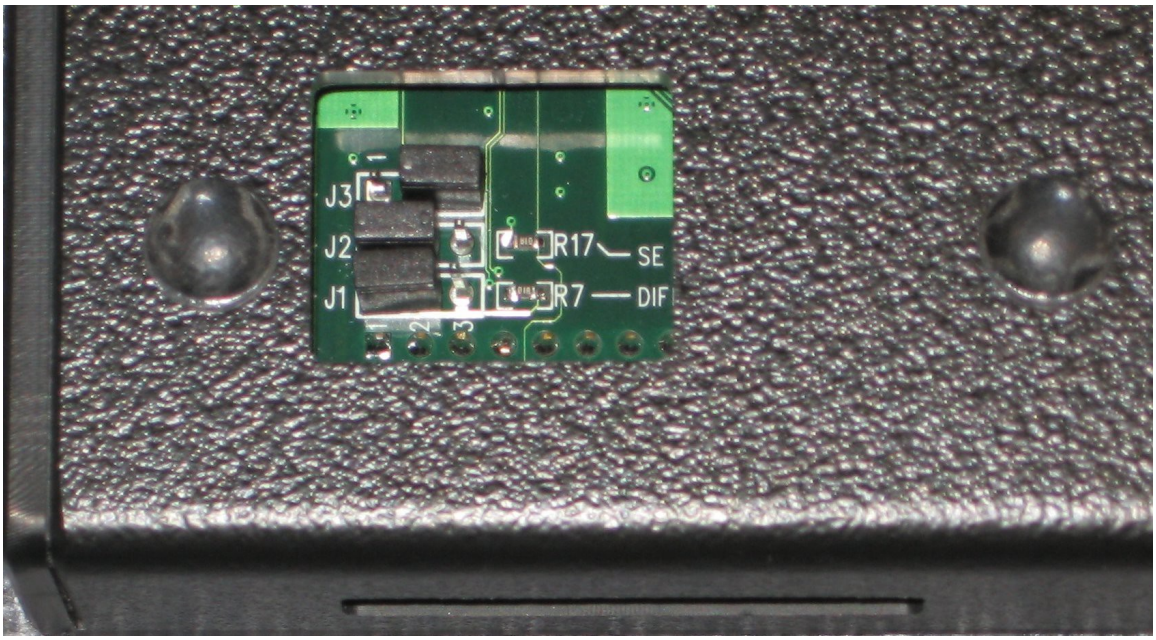


Figure 19 Bottom side of Inputs probe showing termination jumpers

Provided with the P300 is a small circuit board with many square pins. This accessory is the "shorting bar". The shorting bar has two functions: 1) to tie unused clock and event inputs to a known logic level, 2) to "contain" the unused signals in the P300 leadset so that they do not inadvertently touch active circuitry. In systems where the logic low level is ground, a user connects the unused clock and event inputs to the square pins that are tied together. In systems where the logic low level is not ground, the ground side of the leadset should NOT be connected to the shorting bar and an additional connection from the user system logic low level should be made to the shorting bar.

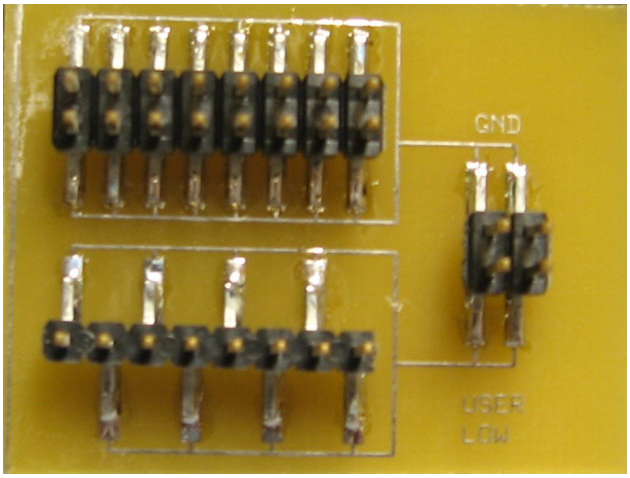


Figure 20 Shorting Bar

In the PGAppDotNet software, there are settings for the event and clock threshold voltages. If the user has selected the differential clock input for the P300 (via the jumper on the back of the probe), the threshold should be set to 0 volts. The threshold is not adjustable in the case of the differential input.

7 Troubleshooting notes

7.1 *"PG is locked by another user. Please select a different PG."*

In rare instances of a software failure or a computer crash, upon the next starting of the PG application, the user may get a dialog box:

"PG is locked by another user. Please select a different PG."

This is the result of a semaphore file not being deleted at application closure.

If you get this error AND you are sure that there are no other PGAppDotNets open and connected to this PG, then you can find and delete the semaphore file to correct the error.

The semaphore file is located in the same directory as the application PGAppDotNet.exe. Its name takes the form "**PGSema**xxxx" where xxxx is the last 4 digits of the serial number of the unit.

7.2 *Trouble with "CyUSB.sys"*

On some computers, the operating system seems to have trouble automatically finding our hardware driver file "CyUSB.sys". Our installer puts this file in c:\windows\system32\drivers.

If your computer needs help finding this file, it should be easy for you to "Browse" for it in the directory listed above.

8 Application Notes

8.1 Timing up a system

Often a user needs data from the PG to arrive at the system-under-test (SUT) at an exact moment in time relative to the clock. These notes may help the user understand the controls available and how to use them to accomplish this goal.

User-provided clock via the P300 Inputs Probe

In this case, the primary control for adjusting the data phase to the user clock is the in the Module setup window (see 4.5.3.1above). The user should select Clocking -> External. Once this selection is made, then the adjustments for the External clock source are available. The Delay control will allow the user to adjust the delay so that any phase is achievable (at clock rates greater than 1 MHz). The delay from the source User clock at the SUT through the PG and back to the Output Probe is unimportant in the continuous clock case – simply add more delay until the desired clock phase is achieved.

To set up the delay, use an oscilloscope or logic analyzer in real-time and, while running a pattern with changing data, observe a changing data bit from an output probe relative to the clock at the SUT and increasing the delay until the desired relationship is achieved. In this case, users would not normally use the clock output of the Output Probe. The user should then stop and restart the PG (perhaps saving the System first) to assure proper operation across all data channels after adjusting the delay (see "Data Delay" in section 4.5.3.3 for an explanation).

PG to FIFO connection

In many systems, an interface between to logical sections is connected via a FIFO. While there are many types of FIFO interfaces, we are referring to an interface that has Data, Clock, Write enable inputs and a "Ready" output signal at the FIFO. The most straightforward interface for use with the PG is:

1) tie WE active always
2) gate the clock with the Ready signal from the FIFO. Connect this gated clock to the P300 Inputs probe as the clock input and use the PG in discontinuous clock mode.
3) Connect the clock from an output probe to the FIFO input clock, adjusting the timing, if necessary.

In this manner, whenever the FIFO is ready, the PG will clock a piece of data into it. This avoids the use of events (and the round-trip delays incurred) and special programming to construct an interface.

9 Appendix A – PG3A Specifications

Characteristics and Specifications

Tables 1 through 4 list the characteristics and specifications for the pattern generator. Table 5 lists details about the shipping container.

All specifications are valid after the unit has been operating for 30 minutes in a temperature-stable environment.

Please note that any operation outside of any of these specifications could cause damage that might take many forms: immediate faulty operation, short-term mis-operation, out-of-spec operation, long-term mis-operation, extreme non-repairable damage, or hazard to the user. The fault may not be obvious after the damage occurs. Do not operate this equipment outside the specifications found below.

Table 1: PG operational modes and limits

Characteristic	Description
Operational mode	
Normal	Pattern data output is synchronized by the internal/external clock input
Step	Pattern data output is synchronized by software command or PGAppDotNet directive
Output pattern	
Maximum Data Output Rate	300Mbits per second per channel, option –P3DDR is 600Mbps/channel, 32 channels
Maximum Clock Output Frequency	300 MHz, 4 clock outputs
Maximum Operating Frequency	The maximum operating frequency of the module is a function of the output level, output pattern and the load condition, including the series termination resistor in the probe. Operating conditions exceeding this frequency may result in damage to the probe.
Pattern length	4 to 32,000,000 (32 MVectors)[12] Option -64M increases the maximum to 64,000,000.
Number of channels	64 data, 4 strobe, 8 byte enables
Sequences Maximum	3965
Number of blocks Maximum	3965
Repeat count	1 to 1048575 or infinite
PGAppDotNet software	Control software requirements and limits
Operating system	Microsoft XP, Windows 7-32, Windows 7-64

[12] Because of the way that the pattern compiler works, it is difficult for the user to predict actually how much memory is required for a given pattern. Issues like partial loop unrolling, small block replication, etc. greatly complicate the calculation. Of course, once a pattern is compiled, the memory footprint is exactly known. Please contact the factory for more information.

Table 2: PG Clocking and Reference Input/Output specifications

Characteristic	Description
Internal Clock	
Clock Period	10mS to 3.33nS
Clock Frequency	100 Hz to 300 MHz
Period resolution	
Frequency accuracy (internal timebase)	+/- 300ppm
External Clock Input	via Inputs Probe, both differential and single-ended, jumper-selectable on Inputs Probe
Clock Frequency	DC to 300 MHz
Threshold	-2.0V to +2.5V (single-ended input only) differential clock input is non-adjustable Diff Common Mode voltage +3V to -2V
Resolution	32 bits
Minimum pulse high, low	1.4 nS
Input Impedance	Jumper-selectable, on or off, resistor changeable by user, shipped as 100 ohm differential, 50 ohm single ended
Sensitivity	500mV p-p, single ended; 40mV (V+-V-) differential
Round-trip time, approximate	~80nS (30 MHz and greater), else ~120nS Clock at input probe to clock appearing at output probe output.
Clock delay adjustment range	
30 MHz and above clock	17.25 nS in 20pS steps[13]
Below 30 MHz	500 nS in 20pS steps
External Reference Input	via front panel
Frequency and connector type	10 MHz, SMA female connector
Sensitivity	500mV, ac-coupled
Termination	50 ohm
Operating Limits	500mV to 5V peak-peak, R-D limiter internal to PG
External Reference Output	via front panel
Frequency and connector type	10 MHz, SMA female connector
Amplitude (square wave)	
No load	3.3V p-p max, ground-referenced, DC coupled.
50 ohms	1.65Vp-pmax, ground-referenced, DC coupled
Outputs	
Inherent skew, any data output to any data output	+/- 200pS max includes probe cable, skew measured at probe input. Does not include probe skew, which is probe-type dependent.
Byte-skew adjustment range	
30 MHz and above clock	17.25 nS in 20pS steps[14]
Below 30 MHz	500 nS in 20pS steps

[13] An inversion control is also available that effectively doubles the range. This gives the user at least one clock cycle of delay adjustment at the lowest clock frequency.

[14] An inversion control is also available that effectively doubles the range. This gives the user at least one clock cycle of delay adjustment at the lowest clock frequency.

Table 3: PG Event Processing and Inputs Probe Mechanical

Characteristic	Description
Event Action	Advance or Branch
Number of Event Inputs	9 External: 8 from Inputs Probe, 1 from front panel "Trig In"
Number of Event definitions	8: up to 512 event input patterns can be OR'd to define an event
Event Filtering	0 ns, 25 ns, or 50 ns
Event threshold	-5V to +5V
Sensitivity, minimum	500mV p-p
Signal amplitude, limits	-5VDC to +5VDC
Variable Event Setup Latency (FREQLAT) (used below in Waitfor/Branch setup calculation)	300 MHz = 2 user clocks 250 MHz = 6 user clocks 200 MHz = 8 user clocks 150 MHz = 12 user clocks 100 MHz = 14 user clocks 50 MHz = 18 user clocks <= 10 MHz = 22 user clocks
WaitFor Setup Time	130 ns + <filter dly> + 10 user clocks + FREQLAT
Branch Setup Time	130 ns + <filter dly> + 21 user clocks + FREQLAT
External Trigger Input	via front panel , SMA female connector
Sensitivity, minimum	500mV
Threshold	AC-coupled, not adjustable, edge only
Termination	1.05 kOhm
Operational limits	500mV to 5V p-p
Risetime	No slower than 10V/uS
External Trigger Output	via front panel, , SMA female connector
No load	3.3V p-p typ, ground-referenced, DC coupled.
50 ohms	1.6V p-p typ, ground-referenced, DC coupled

Inputs Probe	Characteristic	Notes
Weight	120 grams	approximate
Dimensions	Length: 117mm, Width 77mm, Height: 27mm	approximate
Cable to module	Length: 1500mm	approximate

Table 4: PG Electrical and Mechanical

Characteristic	Description
PG3AMod	module version of pattern generator
Slot Width	Requires two mainframe slots
Weight, approximate	2.0 kg
Overall dimensions, approximate	Depth: 375mm, Width 275mm, Height: 65mm
Power, maximum	80 W, drawn from TLA mainframe
PG3ACab	cabinet (stand-alone) version of pattern generator
Weight, approximate	4kg
Overall Dimensions, approximate	Depth: 400mm, Width: 305mm, Height: 97mm
Power, maximum	100 W
Voltage	100 – 240 VAC, 50-60 Hz
Power Factor Correction	Active, meets EN61000-3-2
Fuse	
110V nominal operation	1 amp Slowblow, 250 V or equivalent, 1.25 inch x 0.25 inch : **F1A, 250VAC**
220V nominal operation	2 x 1 amp slow blow, 250V 5 mm x 20 mm: **F1A, 250VAC**
Misc	
P300 Dimensions	125mm x 75mm x 30mm approx without leadset
P300 weight	About 115 grams without leadset or input cable
Cable from module to output probe	Approximately 1500 mm

Table 5: PG Shipping Container Materials

Material	Amount
Cardboard (paper)	~1320 grams
Polyethylene (expanded foam)	~165 grams
Urethane foam pads	~115 grams
Shipping container dimensions	485mm x 420mm x 305mm

10 Appendix B – Filling Data From A File

One of the options in the context-menu of the Listing window is "Fill->From File". This function was introduced as support for the P321 serial probe to automate generation of serial-data protocol signals from data values stored in a file, making it more convenient and less tedious for the user to build complex serial-data patterns. In the future, this feature may be expanded to support more general functionality.

10.1 Fill-From-File Operation

As it is currently defined, the fill-from-file function allows a text file to be used as input to a data compiler to generate low-level serial bit patterns for RS232/RS422, I2C, and SPI. To activate the function:

1) Right-click in a group column header in a Listing window (or select a single-column range of cells and right-click)
2) Select the "Fill" context-menu option and the "From File…" sub-menu option to bring up a file dialog.
3) Browse for the text file to use as input and click "Open".

The data compiler will import data from the file, parse it according to embedded commands, and build a column of data vectors to enter into the selected group or column of cells. If the current block length is different than the length of the generated data from the file, the user is asked whether the block length should be lengthened or shortened accordingly. The user can click on Yes, No, or Cancel with the following results:

1) "Yes": lengthen or shorten block as necessary, padding with zeros or truncating other groups in the block respectively.
2) "No": leave block length and truncate or pad with zeros the generated data as necessary.
3) "Cancel": abort the operation.

10.2 Fill-From-File File Definition

The data pattern file is a text file and follows these rules:

- Each line is either a comment, command, or data line.
- White-space at beginning or end of lines is ignored.
- Comment lines can be blank or begin with "//"
- Commands and data lines can also end with comments. All text after "//" in the line is ignored.
- Command lines begin with "#" and are followed by the command plus arguments separated by white-space.
- Commands and arguments are case-insensitive. Numeric arguments are interpreted as decimal unless prefixed by "0x", where they are interpreted as hex.

- Data lines generally have a single hex, decimal, or binary number (interpreted according to the current radix). In the case of a multiple-field defined SPI protocol, a data line will have N numbers separated by white-space, where N is the number of SPI fields defined.

Currently, six commands are supported:

RADIX <Decimal | Hex | Binary>
Sets the current radix for data lines. Note, this setting does not affect numeric arguments which are interpreted as either decimal or hex by virtue of whether they are prefixed with "0x".
Ex: "RADIX Hex" (set the radix to hexadecimal)

DATA
Declares that data lines to follow are basic numeric values to import.

ASEQ <Start Val> <Incr> <Cnt>
Defines an arithmetic sequence with <Cnt> vectors, beginning with <Start Val> and incrementing by <Incr>. Values wrap and continue if the maximum value is reached. No data lines need follow this command.
Ex: "ASEQ 0x10 1 16"

RS232 <DataBits: 7 | 8> <Parity: Even | Odd | None> <StopBits: 1 | 2>
Declares the protocol for the data lines to follow as RS232/RS422. The three arguments define RS232 settings: number of data bits, parity and number of stop bits.
Ex: "RS232 8 Even 1"

I2C <DevAddr> <ExtFlag: 0 | 1> <Byte Count>
Declares the protocol for the data lines to follow as I2C. The three arguments define I2C settings: the device address (7-bit or 10-bit), a flag indicating whether the device address is an extended 10-bit address or normal 7-bit address, and the number of bytes to send.
Ex: "I2C 0x22 0 100"

SPI <Mode: 0-3> <1-8 field specifiers of the form [0:x] or [x:0]>
Declares the protocol for the data lines to follow as SPI. The arguments define SPI settings: mode indicating clock phase and polarity, and field specifiers that define data field lengths and endianism. A field specifier of the form [0:x] defines the field to have length (x+1) bits and little-endian output ordering. A field specifier of the form [x:0] defines the field to have length (x+1) bits and big-endian output ordering.
Ex: "SPI 2 [7:0] [23:0] [7:0]

SPI data lines for SPI contain values for each of the fields. Fields are output in order (left-to-right). Bits of each field are output according to the endian ordering indicated by the field specifier.
Ex: "a0 7f6600 40" (radix is hex)

The SPI mode indicates the (CPH, CPOL) setting. CPH determines the clock phase of a quiescent line: CPH = 0 means SCLK is low when quiescent and CPH = 1 means SCLK is high when quiescent. CPOL indicates which clock edge data is sampled by the slave. CPOL = 0 indicates sampling on the first clock edge and CPOL = 1 indicates sampling on the second clock edge. See Figure 21 for a graphical representation of SPI mode.

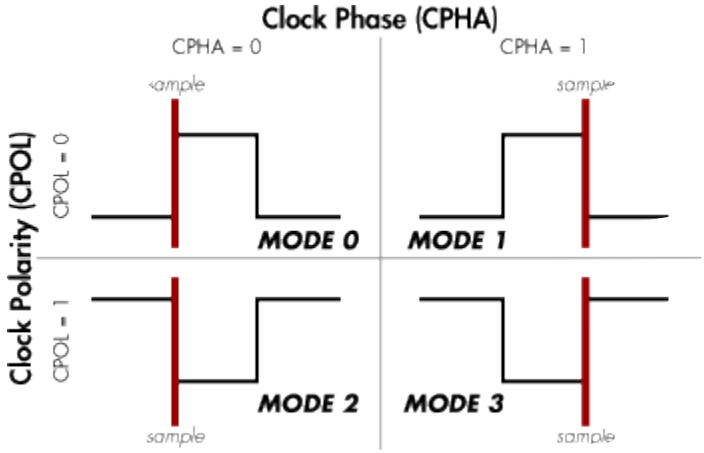


Figure 21 – SPI mode descriptions

11 Warranty

Standard Product Warranty
The *Moving Pixel* Company

The *Moving Pixel* Company (TMPC) warrants that the products that it manufactures and sells will be free from defects in materials and workmanship for a period of one (1) year from the date of shipment. If a product proves defective during this warranty period, TMPC, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify TMPC of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by TMPC, with shipping charges prepaid. TMPC shall pay for the shipping charges resulting from the return of the product to Customer. Customer shall be responsible for paying all duties, taxes, insurance, and any other charges for products returned.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. TMPC shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than TMPC representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; c) to repair any damage or malfunction caused by the use of non-TMPC supplies; or d) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

THIS WARRANTY IS GIVEN BY TMPC IN LIEU OF ANY OTHER WARRANTIES, EXPRESS OR IMPLIED. TMPC AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TMPCs' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TMPC AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE
OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. THE ABSOLUTE MAXIMUM LIMIT OF LIABILITY UNDER THIS WARRANTY IS TO THE AMOUNT PAID BY THE CUSTOMER FOR THE PRODUCT FOR WHICH A WARRANTY CLAIM IS MADE.

Contact information:
The *Moving Pixel* Company
4905 SW Griffith Drive, Suite 106
Beaverton, Oregon 97005 USA
+1.503.626.9663 Pacific Time Zone
+1.503.626.9653 fax
www.movingpixel.com
information@movingpixel.com

12 Software License Agreement

PLEASE READ THIS DOCUMENT CAREFULLY BEFORE USING THE SOFTWARE. BY USING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT OPEN, INSTALL, OR USE THE SOFTWARE. PROMPTLY RETURN WITHIN 15 DAYS THE SOFTWARE, ALL RELATED DOCUMENTATION, AND ACCOMPANYING ITEMS TO THE PLACE OF ACQUISITION FOR A FULL REFUND.

This is a legal agreement between you and **The *Moving Pixel* Company** (TMPC). This Agreement states the terms and conditions upon which TMPC offers to license the software sealed in the disk package, together with all related documentation and accompanying items including but not limited to, the executable programs, drivers, libraries, and data files associated with such programs (collectively, the Software).

LICENSE

1. Grant of License

The Software is licensed, not sold, to you for use only under the terms of this Agreement. You own the disk or other media on which the Software is originally or subsequently recorded or fixed as permitted by this Agreement. However, as between you and TMPC (and, to the extent applicable, its licensors), TMPC retains all right, title, and interest to the Software and all copyrights to the Software, and reserves all rights not expressly granted to you. This is a non-exclusive license.

2. Limitation on Using, Copying, and Modifying the Software

Except to the extent expressly permitted by this Agreement or by the laws of the jurisdiction where you acquired the Software, you may not modify the Software. Nor may you sub-license any of your rights under this Agreement.

3. Decompiling, Disassembling, or Reverse Engineering

You acknowledge that the Software contains trade secrets and other proprietary information of TMPC and its licensors. Except to the extent expressly permitted by this Agreement or by the laws of the jurisdiction where you are located, you may not decompile, disassemble, or otherwise reverse engineer the Software, or engage in any other activities to obtain underlying information that is not visible to the user in connection with normal use of the Software. In any event, you will notify TMPC of any information derived from reverse engineering or such other activities, and the results thereof will constitute the confidential information of TMPC that may be used only in connection with the Software.

TERMINATION

The license granted to you is effective until terminated. You may terminate the license at any time by returning the Software (including any portions or copies thereof) to TMPC at the address shown below. The license will also terminate automatically without any

notice from TMPC, if you fail to comply with any term or condition of this Agreement. You agree upon such termination to, except in the State of California, return the Software (including any portions or copies thereof) to TMPC. In the State of California, destroy the Software (including any portions or copies thereof and notify TMPC of such destruction in writing. Upon termination, TMPC may also enforce any rights provided by law. The provisions of this Agreement that protect the proprietary rights of TMPC will continue in force after termination.

LIMITED WARRANTY

TMPC warrants, as the sole and exclusive warranty, that the disks on which the Software is furnished will be free of defects for a period of ninety (90) days. In the event one or more of such disks is defective, TMPC will replace the defective disk(s) free of charge upon receiving the defective disk at the address set forth below.

No distributor, dealer, or any other entity or person is authorized to expand or alter this warranty or any other provisions of this Agreement. Any representation, other than this express limited warranty, will not bind TMPC.

EXCEPT AS STATED ABOVE IN THIS AGREEMENT, THE SOFTWARE IS PROVIDED AS-IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

TMPC does not warrant that the functions contained in the Software will meet your requirements, or that the operation of the Software will be uninterrupted or error-free. You assume full responsibility for the selection of the Software to achieve your intended results, and for the installation, use, and results obtained from the Software. You also assume the entire risk as it applies to the quality and performance of the Software. Should the Software prove defective you (and not TMPC, or its distributors or dealers) assume the entire cost of all necessary servicing, repair, or correction.

This warranty gives you specific legal rights, and you may also have other rights that vary from country/state to country/state. Some countries/states do not allow the exclusion of implied warranties, so the above exclusion may not apply to you. TMPC disclaims all warranties of any kind of the Software was customized, repackaged, or altered in any way by any third party other than TMPC.

LIMITATION OF REMEDIES AND DAMAGES

THE ONLY REMEDY FOR BREACH OF WARRANTY IS THE EXPRESS LIMITED WARRANTY SET FORTH ABOVE. IN NO EVENT WILL TMPC OR ITS LICENSORS BE LIABLE FOR ANY PUNITIVE, INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES, OR FOR ANY LOST PROFITS, LOST SAVINGS, LOST REVENUES, OR LOST DATA ARISING FROM OR RELATING TO THE SOFTWARE OR THIS AGREEMENT, EVEN IF TMPC OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL TMPCS LIABILITY OR DAMAGES TO YOU OR ANY OTHER PERSON EVER EXCEED THE AMOUNT PAID BY YOU TO USE THE SOFTWARE, REGARDLESS OF THE FORM OF THE CLAIM. Some countries/states do not allow the limitation or exclusion of liability for the

incidental or consequential damages, so the above limitation or exclusion may not apply to you.

PRODUCT RETURNS

If you must ship the Software to TMPC or an authorized TMPC distributor or dealer, you must prepay shipping and either insure the software or assume all risk of loss or damage in transit.

U.S. GOVERNMENT RESTRICTED RIGHTS

All Software and related documentation are provided with restricted rights. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable. If you are sub-licensing or using the Software outside of the United States, you will comply with the applicable local laws of your country, U.S. export control law, and the English version of this Agreement.

MANUFACTURER

The *Moving Pixel* Company
4905 SW Griffith Drive #106
Beaverton OR 97005
+1.503.626.9663 phone
+1.503.626.9653 fax
www.movingpixel.com

GENERAL

This Agreement is binding on you as well as your employees, employers, contractors and agents, and on any successors and assignees. Neither the Software nor any information derived therefrom may be exported except in accordance with the laws of the U.S. or other applicable provisions. This Agreement is governed by the laws of the State of Oregon (except to the extent federal law governs copyrights and federally registered trademarks). This Agreement is the entire agreement between us and supersedes any other understandings or agreements, including but not limited to, advertising of the Software. If any provision of this Agreement is deemed invalid or unenforceable by any country or government agency having jurisdiction, that particular provision will be deemed modified to the extent necessary to make the provision valid and enforceable, and the remaining provisions will remain in full force and effect. If any legal action is brought by you or TMPC regarding the Software or this Agreement, the prevailing party shall be entitled to recover, in addition to any other relief granted, reasonable attorney fees and expenses of litigation. Neither you nor TMPC will waive any rights under this Agreement, unless such waiver is in writing. For questions concerning the Software or this Agreement, please contact TMPC at the address stated above.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP