Operators Manual
For

Pulse Instruments

PI-2005 Pattern Generator and PI-PAT Software

January 2016

Pulse Instruments
1234 Francisco Street
Torrance, CA 90502
(310) 515-5330
http://www.pulseinstruments.com
support@pulseinstruments.com

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

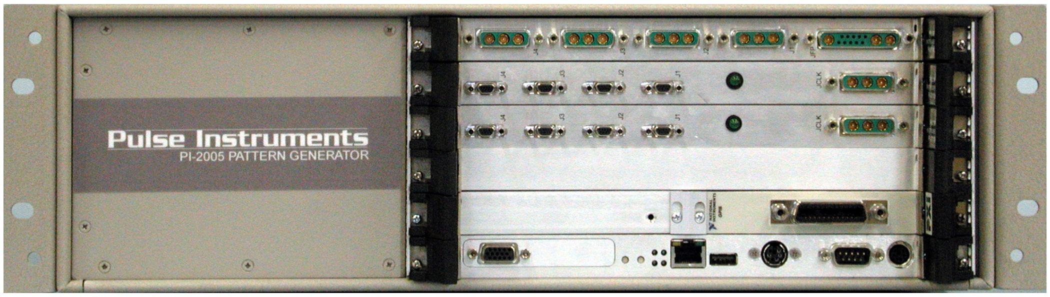


Figure 1: PI-2005 Pattern Generator

Pulse Instruments PI-2005 Pattern Generator can be used to generate a wide range of simple to complex digital patterns for any test application that requires a serial or parallel digital data stream. With almost infinite looping capability and up to 8 MBits per channel, the PI-2005 can create digital data patterns at speeds and complexities that fulfill the most demanding requirements.

The data generator provides master clock rates from 12 kHz to 250 MHz, with either LVTTL, TTL, or 5V/CMOS output drivers. The unit can be configured from 16 to 64 channels in 16 channel increments. Channel to channel skew is less than 600 picoseconds. Each channel of the PI-2005 can be individually programmed to be either RZ or NRZ format. The 250 MHz, phase-locked-loop clock gives you the capability of high resolution timing and picosecond edge control.

The unit is programmed by inputting pattern data, defining subpatterns, and defining programs to output those subpatterns. Pattern data consists of binary data entered into up to 8 MBits of pattern memory for each channel (depending on the cards installed), and subpatterns associates ranges of bits with unique subpattern numbers. A subpattern can be from 12 bits to 8 MBits long, and up to 256K subpatterns can be defined.

The output data stream is defined by instruction or program steps that control the execution sequence and repeating of subpattern data. This technique provides virtually endless data pattern depth. Multiple programs can be stored in a single file with all their associated pattern data and subpattern definitions.

The PI-2005 was designed around an x86-based single-board-computer, and includes a hard drive, network interface, and an optional GPIB/IEEE-488 interface. These features enable the PI-2005 to be used as the system controller in your test setup, as a slave instrument over GPIB, or as a standalone instrument.

The PI-2005 is programmed using Pulse Instruments' PI-PAT software operating under Windows 7. This provides a user-friendly interface for programming, editing and controlling the Pattern Generator. The PI-2005 can also be used as a remote instrument controlled over the GPIB/IEEE-488 bus by PI-PAT software or other existing software running on another computer. In this mode, the PI-2005 can emulate a PI-5800A, providing an easy upgrade path for existing test systems presently using the PI-5800A.

1.1. Before You Get Started

1.1.1. How to Read This Manual

This manual describes operation of CompactPCI-based pattern generator instruments from Pulse Instruments. This includes products marketed as the PI-2005, PI-2005A or PI-11000 Instrument Mainframe, or any CompactPCI chassis containing Pulse Instrument pattern generator cards (including PI-21000 Clock Generator card and PI-2110x or PI-2120x Pattern cards). The instrument chassis may also contain a variety of other CompactPCI instrument cards, either from Pulse Instruments or from a 3rd party.

Any of these configurations will be described in this manual as "PI-2005," and this manual will describe primarily operation of the clock generator cards and pattern cards.

If you are operating your pattern generator in conjunction with other equipment from Pulse Instruments (e.g. clock driver cards, low-noise DC bias cards, or data acquisition subsystems), then you will likely also be using PI-Controller software. PI-Controller can provide top-level control of the PI-2005 by making calls to PI-PAT.

This manual should be read first, followed by the PI-Controller manual for the PI-11008.

If you intend to write or use custom software to control the PI-2005, you should read this manual in its entirety, to become familiar with the features and controls, before attempting to program the instrument.

1.1.2. Packing List

Your PI-2005 Pattern Generator was packed with the following items:

- Chassis
- AC power cable
- CPU card installed in the CPU slot
- PI-21000 Clock Card installed in one of the remaining slots
- One or more PI-2110x or PI-2120x Pattern Cards installed in the remaining slots
- One clock input cable per pattern card
- One or more pattern card output cables per pattern card
- Windows 7 installation CD
- PI-DATS installation CD (including PI-PAT) with contents copied onto the hard drive in the C:\Drivers directory
- GPIB/IEEE-488 interface card (optional), with the National Instruments driver package in the C:\Drivers folder:
- This printed manual (also available in Portable Document Format in the C:\Documentation folder)

1.1.3. Configuration

The PI-PAT control software, Windows operating system, video drivers, and network drivers have been preloaded into the Pattern Generator. If you have ordered the optional GPIB/IEEE-488 interface, those drivers have also been pre-installed. All software has been provided in its original format on CD. Drivers for the video and network chipsets of the CPU board are included with Windows 7.

1.1.3.1. Local vs. Remote Operation

PI-PAT can control the PI-2005 locally, or it can control the PI-2005 remotely from another computer, over the IEEE-488 bus. In Pulse Instruments' terminology, "Local" always refers to the PI-2005, regardless of where the operator is sitting. "Hardware" refers to the clock cards and pattern cards in the local PI-2005.

In “Local” mode, PI-PAT runs on the single-board-computer inside the PI-2005. It stores pattern files on the local hard drive and writes directly to the pattern and instruction memory in hardware. If ordered without a GPIB interface the PI-2005 is configured by default as a local instrument, to be used with a user-supplied monitor, keyboard and mouse.

If the unit was ordered with the optional GPIB interface, it will have been configured either as a System Controller or as an Instrument.

In System Controller mode, the default mode is **Local Only**, and the PI-2005 can control other instrumentation, such as DVMs, oscilloscopes, etc.

In Instrument mode, the default address is 14, and the default mode is **Local GPIB**. These settings can be changed via the **Hardware** dialog box in Local mode.

Another copy of PI-PAT may run in **Remote GPIB** mode on a remote computer, handling graphical pattern editing and passing data and commands over the IEEE-488 bus to a PI-2005 in **Local GPIB** mode. PI-PAT on the local machine listens for those calls and passes them to hardware. Pattern and instruction files can be stored locally and run via a simple set of commands from the remote machine. This architecture allows a central computer to control multiple “local” PI-2005 Pattern Generators while minimizing bus traffic and distributing the processing load.

From a Remote computer, the local IEEE-488 address can be changed, but the IEEE-488 mode cannot be set to Controller. That setting must be changed from the local machine, which will require a monitor, keyboard and mouse to be attached.

1.1.3.2. Windows Logon (optional reading)

The Windows 7 “auto logon” function has been enabled so that the user does not have to logon to the computer every time it starts up. By default, the auto-logon feature logs the user into the local machine only, with local Administrator privileges.

If you plan to connect the PI-2005 to your company network or to the Internet, consult your network administrator for the appropriate network settings and permissions. This may also require you to modify or disable the Windows automatic logon features in order to comply with your company’s specific network configuration and security policies. See the following Microsoft Knowledge Base articles for details on modifying or bypassing the auto-logon settings.

http://support.microsoft.com/support/kb/articles/Q114/6/15.asp

http://support.microsoft.com/support/kb/articles/Q97/5/97.asp

Your PI-2005 has been configured with the following Windows account information:

- Computer name: PI2005_<serial>
- Username: Administrator
- Password: password
- Workgroup: WORKGROUP

where the “<serial>” is the 10-digit serial number of your PI-2005.

1.2. Known Issues and Significant Format Changes, PI-2005 and PI-DATS 2.708 (including PI-PAT)

1.2.1. PI-PAT Application (pipat.exe)

1.2.1.1. Instruction line comments are now imported/exported with text files

Instruction line comments are now preserved when PI-PAT files are exported as text files. The comments are delimited with a bang (!) character.

Older versions of PI-PAT (prior to PI-DATS 2.621) will not correctly parse instruction line comments and will generate syntax errors. Text files must be edited to remove instruction line comments before they are sent to instruments running older software.

1.2.1.2. Channel names are now imported/exported with text files.

Channel names are now exported using the NAME CHANNEL command.

Older versions of PI-PAT (prior to PI-DATS 2.627) will not correctly parse channel names and will generate syntax errors. Text files must be edited to remove channel names before they are sent to instruments running older software.

1.2.1.3. Instruction comments no longer terminate programs

Comments between instruction lines no longer terminate a program. This will break compatibility with earlier versions of PI-PAT (prior to PI-DATS 2.626) only in the case where an instruction comment was deliberately inserted to terminate a program. Blank lines or another BEGIN statement can still be used to terminate a program.

1.2.1.4. PI-PAT permits up to 256K of pattern depth for PI-2110x cards

PI-2110x cards have always had 256 Kbits of pattern memory installed per channel, but software support was limited to 64 Kbits until PI-DATS 2.627. PI-PAT now permits bit patterns up to 256K in depth for PI-2110x cards. This will break compatibility with instruments running earlier versions of PI-PAT software (prior to PI-DATS 2.627) that limited pattern memory to 64K bits.

Files in the PI-2005 Extended forms (.w25x) support up to 8 Mbits of pattern memory per channel, are compatible only with PI-2120x cards.

1.2.1.5. Commands beyond last installed channel may be incorrectly applied

ASCII commands for pattern channels beyond the last installed channel will get applied to the last installed channel unless your application explicitly checks for and halts on errors. Use NUMCHAN command to determine the number of installed channels before sending commands, or check STATUS after every GOTO and LOAD command to ensure the specified channel exists.

1.3. Specifications

Master Clock:	
Frequency:	12 kHz to 250 MHz (Internal Clock) <1 Hz to 250 MHz (External Clock)
Accuracy:	<1.0%
Resolution:	<0.625% of Programmed Value
Period Jitter:	<100 ps
External Clock Input:	TTL or ECL levels, software selectable, Conec-13C3 connector
External Clock Output:	LVTTL levels
External Clock Output to Pattern card output delay:	12 - 14 ns

Output Characteristics:	
Channel to channel skew: • Between similar type channels • Between standard and delayed channels	 ≤ 600 ps 3.8ns ± 600 ps
Output Levels: Into 1 MΩ impedance	PI-21x00-y-z Pattern Cards LVTTL compatible PI-21x01-y-z Pattern Cards TTL compatible PI-21x02-y-z Pattern Cards: 5V/CMOS
Rise/Fall time (10%-90%):	LVTTL Outputs: < 3.2 nanoseconds TTL Outputs: < 2.5 nanoseconds 5V/CMOS Outputs: TBD
Output Impedance:	50 Ω ±10%
Pulse Duration:	4.4 ns minimum

Channel Formats:	
Invert/Non-invert:	Independently selectable (all channels)
RZ and NRZ Modes:	Independently selectable (all channels)
Tri-state:	Independently selectable (all channels)
	Channels pairs (Ch 1-2, 3-4, etc, 15-16) can also be tri-stated “on-the-fly” for the duration of one or more instruction lines
Maximum Delay (Delay option only):	5.8 - 6.6ns ns programmable
Steps	64

Subpatterns:	
Minimum Length:	20 bits (PI-2110x cards) 12 bits (PI-2120x cards)
Maximum Length:	262,144 bits, (PI-2110x cards) 1 Mbits or 8 Mbits, (PI-2120x-y-1M or PI-2120x-y-8M cards, respectively)
Maximum Subpatterns:	3,276 (PI-2110x cards) 262,144 (PI-2120x cards)

Program Lines:	
Maximum Repeats/Line:	1,023 repeats per SUBPATTERN instruction (PI-2110x cards) 32,767 repeats per SUBPATTERN instruction (PI-2120x cards)
Maximum Repeats/Loop:	1,023 repeats per REPEAT instruction (PI-2110x cards) 32,767 repeats per REPEAT instruction (PI-2120x cards)
Looping Limitation:	Limited only by the total number of lines in the program.
Maximum Program Lines:	PI-2110x Cards: • 256 K (262,144) x 4 K (4,096) compiled lines for 4K FIFO • 256 K (262,144) x 64 K (65,536) compiled lines for 64K FIFO • 256 K (262,144) x 128 K (131,072) compiled lines for 128K FIFO PI-2120x Cards: • 262,144 compiled lines

Other:	
Software:	PI-PAT running under Windows 7
Power:	115/240 VAC @ 50/60Hz
LVTTL Pattern Card	3.3 A without delay, 4.2 A with delay
TTL Pattern Card	3 A without delay, 5.0 A with delay
Outside Dimensions:	5.5” H x 17.5” W x 12” D
Board Size:	Size 6U Eurocard
Cables:	4 x 4 mass terminated BNC
Board Slots:	6 or 8

Programming Capabilities:	
Step mode:	programmable execution length (up to 255) defined in steps (subpatterns)
Sequence mode:	programmable execution length (up to 255) defined in sequences (Begin - programs)

1.4. Software Overview

Each signal or channel installed in the instrument can be given a mnemonic name by the user. The user and the software then use this name to identify signals, instead of using the chassis, slot, and channel numbers. For example, the pattern channel you are using as a Frame Sync signal can be displayed on the screen labeled as "FRAMESYNC" instead of as "Channel 6”. This use of mnemonic names makes the programming process more intuitive and more convenient.

The display of the signals can be configured by the user. The default display shows the signals in the order they are present in hardware. The user can arrange the display to group the signals in a more functional or logical order and switch back and forth between the user-defined and hardware order.

When working with the user-defined display, the order in which the channels are displayed on the screen does not determine which line of pattern data is sent to which channel. The control software uses the Format Table to "route" the pattern data to the appropriate channels. For example, if FRAMESYNC is assigned to channel 6 in the Format Table, then the pattern data labeled FRAMESYNC will be sent to channel 6 regardless of where the FRAMESYNC channel appears on the screen. By using the Format Table, you can display the signals in groups by their functional association instead of by the order in which their cables were connected. This makes it easier to construct and edit your timing sequences.

The relationship between the physical location and channel name of each signal is stored with the format information, and can be edited by choosing **Edit Formats** from the **Edit** menu.

The data for your test setup is stored in a Waveform File. Waveform Files contain pattern data, subpattern data, program instructions, channel formatting and clock settings. Again, you can have many different Waveform Files stored on the disk under different names.

PI-PAT works by editing the contents of the Waveform Definition Files, and then writing or "downloading" that information to the Pattern Generator. Because PI-PAT writes to the instrument only when the user requests it, programming can be done at a remote computer and written to the instrument at a later time.

PI-PAT stores Waveform files in two formats:

- .w25, supporting PI-2110x and PI-2120x cards
  - .w25 files can be up-saved to .w25x files
- .w25x, supporting PI-2120x cards
  - .w25x files can be down-saved to .w25 files, but may truncate data

The PI-2005 can also read files generated for the PI-5800A.

2. HARDWARE

2.1. Mounting

The PI-2005 is contained in a standard 6U CompactPCI chassis with six card slots. The chassis is compatible with standard 19” racks, and is 5.5” high. There are two cooling fan in the unit—one vents to the left side panel and one vents to the back of the unit. Ensure that the vents are not obstructed when mounting the PI-2005.

2.2. Rear Panel

Insert the provided AC power cable into the power socket on the back of the unit. The internal power supplies on 6-slot mainframes are autoswitching and will accept 110-240 VAC, 50-60 Hz. Internal power supplies on 8-slot mainframes must be manually switched for the correct AC input voltage.

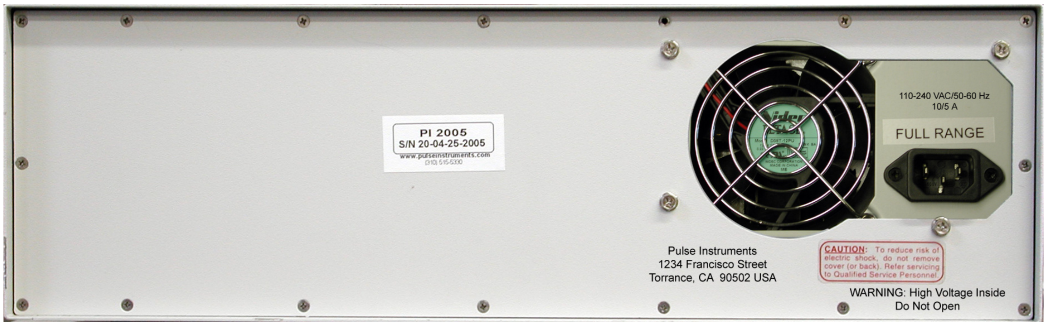


Figure 2: Rear panel of the PI-2005, 6-slot mainframe

On the rear panel is the Pulse Instruments Serial Number of the instrument. Please refer to this number when contacting the factory for service or technical support.

2.3. Front Panel

2.3.1. Power

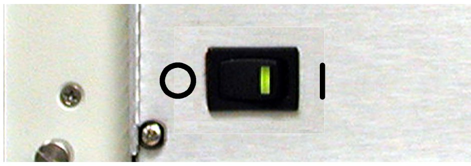


Figure 3: Power Switch

To turn on the PI-2005, press the right side of the rocker switch on the front panel. The LED will light up to indicate that the power is on.

> **WARNING: Do not power up the unit until all proper signal connections have been made. In particular, each PI-2110x Pattern Card must be connected to the PI-21000 Clock Generator Card before power up, otherwise the system will not boot properly or will lock up after booting. If the CLK input cable is connected to the Pattern Card after power up, the system should be rebooted.**

2.3.2. Removing/Inserting CompactPCI cards

Pulse Instruments CompactPCI cards are **not** Hot-Swap compliant. Always turn off power to a CompactPCI chassis before inserting or removing a Pulse Instruments CompactPCI instrument card.

To remove a CompactPCI card from the chassis, loosen the captive screws at the outside of each black ejector handle, then push the ejector handles outward until the card slides out approximately ½". Once the ejector handles have released, pull the card straight of out the chassis until it slides free.

If you are removing a 3U card (for example, the National Instruments IEEE-488 Interface Card),   there may be only one ejector handle. The other end of the card is secured by a small screw, which will have to be removed first

To insert a card, gently slide it into the chassis slot until the ejector handles touch the chassis. Push the locking tabs inward until the card is fully seated in the chassis slot, then secure the captive screws. If you are inserting a 3U card (for example, the National Instruments IEEE-488 Interface Card), there may be only one ejector handle. The other end of the card should be secured with a small screw.

Be sure to properly power down the chassis before removing or inserting a card, unless the card is Hot-Swap compliant. At present, Pulse Instrument Pattern Generator cards (PI-21000 Clock Card and PI-21100 Pattern Cards) are ***NOT*** Hot-Swap compliant.

2.3.3. The CPU board

The CPU card is a 6U, CompactPCI x86-based SBC with the following features:

- Single- or Dual-CPUs from 800 MHz to 2.0 GHz with multimedia extensions
- 256 MB – 4 GB RAM
- On-board video controller
- Interrupt and Reset buttons
- On-board 10/100BASE-T Fast Ethernet or Gigabit Ethernet port(s) with status LEDs
- One or more Universal Serial Bus (USB) ports
- PS/2 mouse and keyboard port(s) on selected models
- One or more serial ports (external or internal)
- CD-ROM, CD-RW or DVD-R drive (optional)

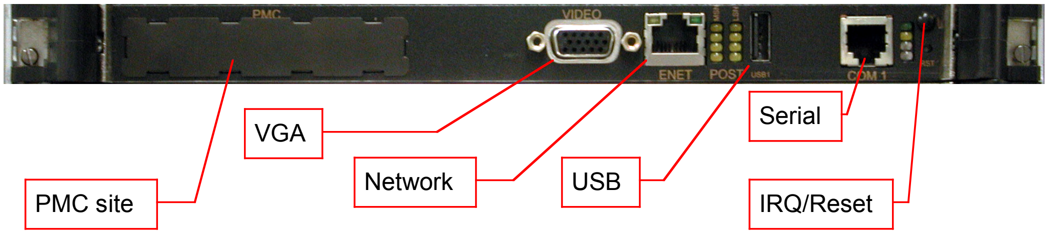


Figure 4: CPU Card (GMS c165 shown)

Your CPU card may look appear slightly differently from the model shown above, based on options you have ordered.

The PI-2005 can be controlled Locally, via a user-supplied monitor, keyboard and mouse, or Remotely, via the optional IEEE-488 interface. If the PI-2005 is being controlled Remotely, connection of the monitor, keyboard and mouse is optional.

2.3.4 GPIB Interface (Optional)

The optional GPIB interface may be supplied in 3 different form factors:

2.3.4.1. PXI/CompactPCI

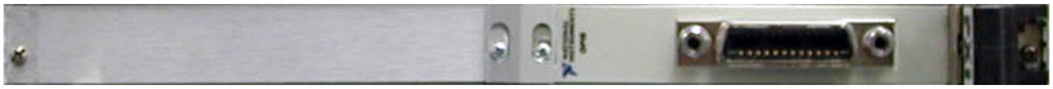


Figure 5: Front Panel of PI-31000 IEEE-488 Card (National Instruments PXI-GPIB)

This is the optional PXI-GPIB National Instruments Card (Pulse Instruments part number PI-31000), and it is used to communicate with the instrument over the IEEE-488 bus. All the required drivers have been installed and the original installation program has been copied to the hard drive in the C:\DRIVERS directory. Please see **Section 7.2.1. Configuring the GPIB Interface** for information on configuring this option.

This interface may be used in both System Controller mode and Instrument mode, under either Windows XP or Windows 7.

2.3.4.2. PMC

Units may also use a PMC plug-in module (PI-31000A). The functionality is identical, except that the PMC module requires a different cable, included with the PMC module.

This interface may be used in both System Controller mode and Instrument mode, under either Windows XP or Windows 7.

2.3.4.3. USB-GPIB

Units may also use a USB module (PI-31000B). On units running Windows XP or earlier this interface may only be used in System Controller mode. On units running Windows 7, this interface may be used for both System Controller mode and Instrument mode.

2.3.5. PI-21000 Clock Generator Card

The figure below shows the front panel of the PI-21000 Clock Generator Card.

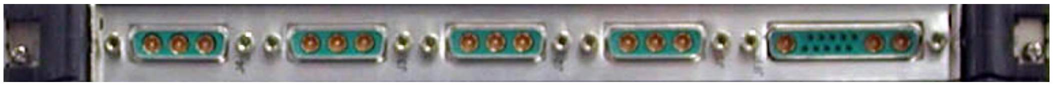


Figure 6: Front Panel of PI-2005 Clock Card (PI-21100)

The Clock card has 4 output connectors and one input/output connector. The four CLK outputs are equivalent, and any pattern card can be connected to any CLK output. Pattern card numbering (and, therefore, pattern channel numbering) is determined by slot order in the mainframe, not by CLK cable connection.

2.3.5.1. Clock Card Output Connectors

The Output connectors are 3C3-type connectors made by CONEC.

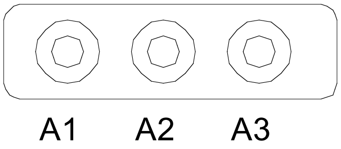


Figure 7: Clock Card Output Connector

Pin A1 – CLK out
Pin A2 – *CLK out
Pin A3 – START out

> **WARNING: The CLK outputs are differential PECL, and should never be connected directly to a ground-referenced input (such as an oscilloscope). This could permanently damage the outputs. To measure or monitor the CLK outputs, use an attenuating probe, a DC-block (such as a Pulse Research Lab PRL-SC-104), or a PECL translator (such as a Pulse Research Lab PRL-550-PQ4X**).

The master clock output can also be measured directly from the JFP connector (listed below) on pin A2, which outputs TTL levels and can safely be monitored directly on an oscilloscope.

2.3.5.2. Clock Card IN/OUT Connector

The IN/OUT (JFP) connector is a 13C3 type connector made by CONEC.

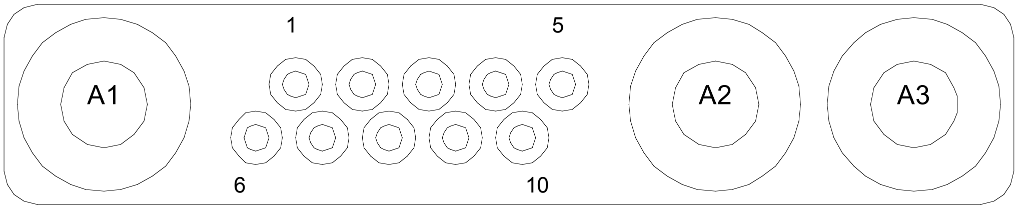


Figure 8: Clock Card IN/OUT connector

2.3.5.2.1. External Clock In

Pin A1 - The External clock in can be selected using the **Hardware** dialog box. TTL or ECL thresholds can be selected. See **Section 3.4.5.1. Clock**

2.3.5.2.2. Stop

Pin 4 - The Stop signal can be used to stop the timing sequence programmed in the PI-2005 from an external source. The **Hardware** dialog box is used to enable and disable this function.

Pin 9 - GND

See **Section 3.4.5.9. Start/Stop from External Signal** for detailed information about the functionality of this pin.

2.3.5.2.3. Start

Pin 5 - The Start signal can be used to start the timing sequence programmed in the PI-2005 from an external source. The **Hardware** dialog box is used to enable and disable this function.

Pin 10 - GND

See **Section 3.4.5.9. Start/Stop from External Signal** for detailed information about the functionality of this pin.

2.3.5.2.4. System Clock Out

Pin A2 - The System Clock Out is the system operating clock as programmed in the **Hardware** Function dialog box. See **Section 3.4.5.1. Clock** for detailed information about the functionality of this pin.

Pin A3 – NOT USED. Reserved for future use.

2.3.5.2.5. I^2C

Pin 1 – I^2C Data
Pin 6 – GND
Pin 2 – I^2C Clock
Pin 7 – GND

2.3.6. PI-21x0y Pattern Card

The PI-2005 supports two families of pattern cards, PI-2110x and PI-2120x. All cards installed in a PI-2005 must be from the same family:

Pattern cards have additional specifiers as follows:

- X specifies the output voltage, where:
  - 0 = LVTTL compatible
  - 1= TTL compatible
  - 2 = 5V/CMOS compatible
- Y specifies the presence of delay:
  - 0 = No programmable delay
  - 1 = Programmable delay per channel
- For PI-2110x cards, Z specifies the size of the looping FIFO memory:
  - 4 = 4096 lines (4K)
  - 64 = 65,536 lines (64K)
  - 128 = 131,072 lines (128K)
- For PI-2120x cards, Z specifies the size of the shared pattern/instruction memory:
  - 1M = 1 MBit
  - 8M = 8 MBit
- All installed cards must have the same FIFO or memory size, but they can have any combination of output voltage and delay.

The figure below shows the front panel of the PI-2110x Pattern Generator Card.

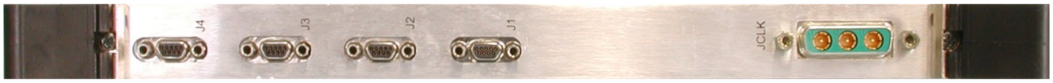
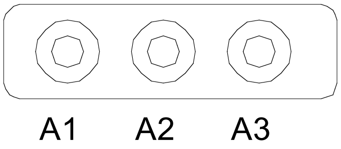


Figure 9: Front Panel of PI-2110x Pattern Card

2.3.6.1. Pattern Card Clock Input Connector (JCLK)

JCLK is the CLK input connector. It is a 3C3 Conec connector designed for high-speed signals.

Each Pattern Card installed in the PI-2005 or in a CompactPCI chassis **must** be connected to a Pulse Instruments PI-21000 Clock Generator Card via its JCLK connector. Failure to connect each Pattern Card to the Clock Generator card can cause the system to become unresponsive. Pattern cards that will not be connected should be physically removed from the system.



Pin A1 – CLK in
Pin A2 – *CLK in
Pin A3 – START in

WARNING: Do not power up the unit until all proper signal connections have been made. In particular, each PI-2110x Pattern Card must be connected to the PI-21000 Clock Generator Card before power up, otherwise the system will not boot properly. If the CLK input cable is connected to the Pattern Card after power up, the system should be rebooted.

2.3.6.2. Status Indicator LED

The green LED on each Pattern Card is a status indicator showing when PI-PAT is running on the PI-2005. When the PI-2005 is being operated in Remote mode over GPIB, this LED will indicate when the PI-2005 is ready to receive commands. When the PI-2005 is first powered up, the LED will remain off for approximately 45-60 seconds until the unit is ready.

When issuing a **Shut Down PI-2005** command from a remote computer, the LEDs will indicate when it is safe to remove power from the instruments. It is safe to remove power from the PI-2005 thirty seconds after the LEDs on the Pattern Cards turn off.

2.3.6.3. Pattern Card Output Connectors

Connectors J1 through J4 are the Output connectors.

The output connector on the pattern card is a 9 pin MOLEX 1.27mm (.050") Pitch Ulti-Mate Commercial Micro D. The MOLEX part number is - Board Mount, 9 Pin Right Angle Plug, with Pin Contacts, 83611-9006.

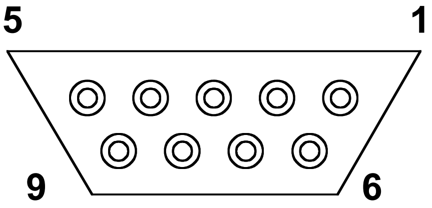


Figure 10: Pattern Card Output Connector, as viewed from the front of the card

Pin 1 – channel 3 signal
Pin 2 – channel 1 GND
Pin 3 – N/C
Pin 4 – channel 4 signal
Pin 5 – channel 2 GND

Pin 6 – channel 3 GND
Pin 7 – channel 1 signal
Pin 8 – channel 4 GND
Pin 9 – channel 2 signal

J1 – Channels 1-4
J2 – Channels 5-8
J3 – Channels 9-12
J4 – Channels 13-16

When used with a Pulse Instruments output cable 88001030-x or 88001035-x, the numbered coaxial outputs will correspond to channels 1-4. When used with Pulse Instruments output cable 88001120-x, the cable is the same on both ends (e.g. wired straight through), and pin 1 is wired to pin 1.

For Pulse Instruments output cables 88001030-x and 88001035-x, the GND pins in the Micro-D are wired to the corresponding BNC/SMA shields, but are not tied together within the cable. The GND signals are tied together on the PCB.

The following tables maps PI-5800 channel assignments to their equivalents in PI-PAT and the PI-2005:

PI-5800	PI-PAT	PI-2005		
Cable	Ch#	Card	J#	Ch#
A1	1	1	1	1
B1	2	1	1	2
A2	3	1	1	3
B2	4	1	1	4
A3	5	1	2	1
B3	6	1	2	2
A4	7	1	2	3
B4	8	1	2	4
A5	9	1	3	1
B5	10	1	3	2
A6	11	1	3	3
B6	12	1	3	4
A7	13	1	4	1
B7	14	1	4	2
A8	15	1	4	3
B8	16	1	4	4

PI-5800	PI-PAT	PI-2005		
Cable	Ch#	Card	J#	Ch#
A9	17	2	1	1
B9	18	2	1	2
A10	19	2	1	3
B10	20	2	1	4
A11	21	2	2	1
B11	22	2	2	2
A12	23	2	2	3
B12	24	2	2	4
A13	25	2	3	1
B13	26	2	3	2
A14	27	2	3	3
B14	28	2	3	4
A15	29	2	4	1
B15	30	2	4	2
A16	31	2	4	3
B16	32	2	4	4

2.4. Slot Numbering

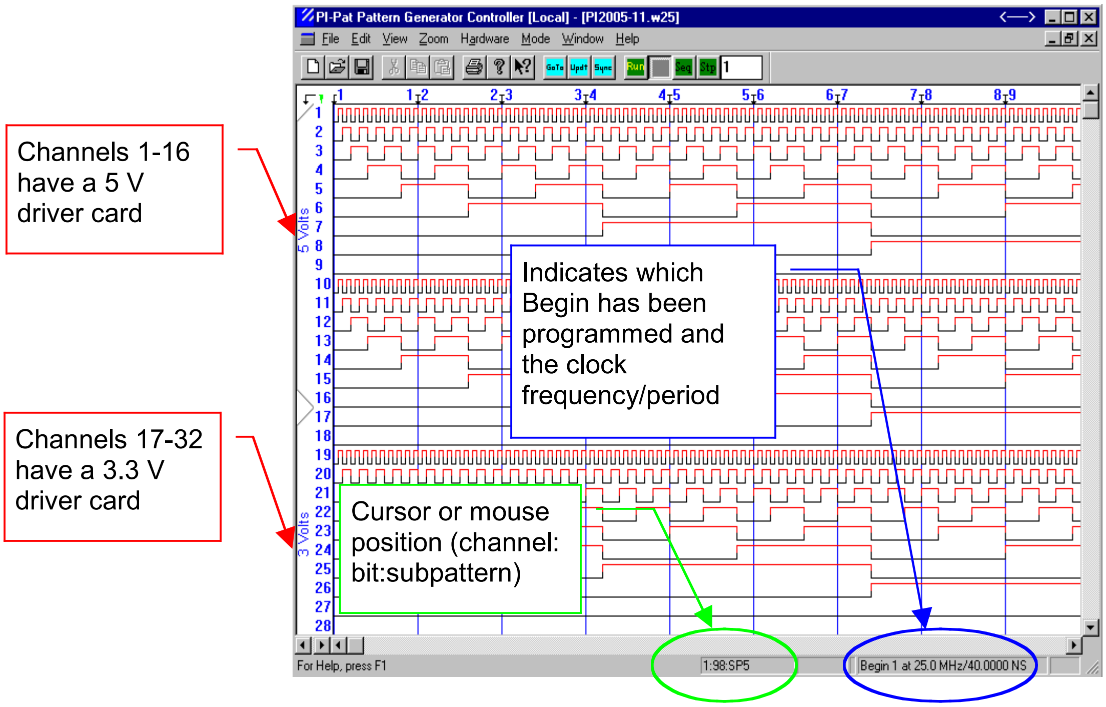
The card slots in a CompactPCI chassis are numbered in descending order, counting away from the system slot, where the CPU card or bridge card is installed. Therefore, for instruments with a system slot at the bottom, Pattern Card #1 will be the top-most pattern card. For instruments with a system slot at the top, Pattern Card #1 will be the bottom-most pattern card.

3. USING PI-PAT

This will be a brief explanation of basic functionality and available menus, followed by a comprehensive section for each menu item and a short tutorial. In this document, menu items or buttons to be selected will be displayed in boldface type, such as the **OK** button, which appears in most dialog boxes.

By default, your PI-2005 has been configured to start PI-PAT automatically every time it is turned on. If you want to disable this feature, you can delete the shortcut to PI-PAT (pipat.exe) in the Windows Startup directory.

If you are running PI-PAT on a remote computer the first dialog box allows you to create a new PI-PAT .w25 or .w25x document. Select the document type either by double-clicking on the appropriate title or by highlighting the title and clicking **OK**. If you are using the PI-2005 in the Local mode this dialog box will not be shown, and a new PI-2005 document corresponding to the installed cards will be created.


PI-Pat Pattern Generator Controller [Local] - [PI2005-11.w25]
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp 1
1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
5 Volts
3 Volts
For Help, press F1
1:98:SP5
Begin 1 at 25.0 MHz/40.0000 NS

Channels 1-16 have a 5 V driver card

Channels 17-32 have a 3.3 V driver card

Indicates which Begin has been programmed and the clock frequency/period

Cursor or mouse position (channel: bit:subpattern)

Figure 11: Main Window with Waveform File Open

From the Main Window you can access all the functions of PI-PAT.

3.1. PI-PAT Main Window

PI-PAT will start up displaying the Main Window and a blank Pattern Window. **Figure 11** shows the Main Window with an existing PI-PAT file open.

The main portion of the window contains pattern data/waveforms for each channel. Channel names and numbers appear in blue down the left side of the window, and the waveform for each channel extends out to the right. Each channel has 256 Kb, 1 Mbit or 8 Mbit of pattern depth available, so only a small portion can be displayed at any time.

Pattern cards can have a 3.3 V or a 5 V output module. The type of output module is indicated on the left-hand side of the screen, when **Hardware Order** is checked (see **Section 3.4.3. View**). If the file is being edited remotely, or if there is no card present in the unit, then output module will be shown as “0 Volts.”

At the bottom of the main window, the Status Bar contains several useful indicators:

- The Channel number, Bit position and Subpattern number at the cursor/mouse position.
- The Begin number (program number) and the clock frequency/period that have been programmed to the hardware. If no instructions have been written to hardware, the Begin number will not appear.
- The Insert/Over-write mode. In Insert mode (default), new pattern data typed, entered, or pasted at the cursor will displace existing data, which will be moved to the right. In Over-write mode, the legend “**OVR**” will appear in the lower right corner of the Main Window, and typed, entered, or pasted data will replace existing data. Use the **Insert** key to toggle between these two modes.

At the top of the main window, the Title bar will show the name of the current pattern file and a legend showing the synchronization state. There are five types of pattern file information that can be synchronized with the hardware:

- Subpattern definitions (s)
- Pattern data (p)
- Format information for channels (f)
- Instruction set (i)
- Clock settings (c)

The Title bar will always show the current synchronization status using the letters above. For example the legend **<spfic>** indicates that none of the settings have been synchronized with the hardware. As each type of setting is written to hardware, the status letter will change to a dash. In the fully synchronized state, the legend will read **<----->**.

3.2. Graphical Pattern Editing

The Main display can be used to edit the pattern data, define signal names and define subpatterns. The pattern data is displayed as waveforms with high logic levels shown in red and low logic levels shown in black.

3.2.1. Inputting Pattern Data

3.2.1.1. Drawing

To create or edit patterns, draw in the Pattern Window with the mouse and the left mouse button. As you drag the mouse, new waveform data will appear in the selected channel.

3.2.1.2. Typing

You can also insert data by typing “1”, “0”, “L” or “H”. High or low-levels will appear at the cursor position. If you are in **Insert** mode (default), existing pattern data will be displaced and moved to the right. In **Over-write** mode, the legend “**OVR**” will appear in the lower right corner of the Main Window, and new pattern data will replace existing data. Use the **Insert** key to toggle between these two modes.

Use the **Backspace** or **Delete** keys on the keyboard to delete data to the left or right of the cursor, respectively.

You can also move the cursor with keyboard arrow keys (← ↑ → ↓).

3.2.1.3. Cut/Copy/Paste

You can also Cut, Copy and Paste sections of pattern data. To select a section of pattern data, drag with the right mouse button (*note that this behavior is slightly different than in other Windows applications*). Once your selection has been highlighted, choose **Edit:Cut** or **Edit:Copy** to make a copy of the selection. Click at another location and then choose **Edit:Paste** to insert the cut/copied data.

3.2.1.4. Inverted Channels

If you are editing an inverted channel (indicated by the legend “INV” next to the channel name or number), the display will reflect the actual output. For example, dragging the mouse low in an inverted channel will draw a high-level output. Similarly, typing “0” in an inverted channel will also draw a high-level output. Changing the output format does not change the underlying data.

3.2.1.5. RZ Channels

If you are editing a channel with return-to-zero (RZ) output format (indicated by the legend “RZ” next to the channel name or number), the display will reflect the actual output.

3.2.1.6. HIZ Channels

If you are editing a channel with Tri-State (HIZ) output format, the channel will be indicated by a grayed-out waveform. Data can still be entered or drawn in the HIZ channel.

3.2.2. Signal Names

By default, signal names are numbers indicating the hardware configuration detected at startup. You can also assign mnemonic names to each signal and display them in any order.

For example, if you are using channel 6 as your FRAMESYNC signal and channel 12 as your PIXELCLK, you could assign these names and then rearrange the display so that they are displayed next to each other. By using this method you can display the signals in groups by their functional association instead of by the order in which their cables were connected. This makes it easier to construct and edit your timing sequences. The signal names are shown on the left side of the display. To change a signal name, click on it and type the new name.

To rearrange the channel order, choose **Edit:Edit Formats**. To temporarily restore the default order, choose **View:Hardware Order**.

3.2.3. Subpatterns

The subpattern start bits and end bits are shown along the top of the display. To add a new subpattern click on and drag the down arrow ⮧ from the upper left corner of the display across the top border of the Pattern Window. As you drag the arrow, a new subpattern definition will appear with solid blue lines, down arrows, and the next available subpattern number.

The down arrow with the number to the right ⮧ is the start of the subpattern and the down arrow with the number to the left ↴ is the end of the subpattern. The default subpattern length for a new subpattern is 12 bits for PI-2120x cards and 20 bits for PI-2110x cards.

You can also edit the subpattern definitions with the mouse. Drag the start arrow ⬐ to move the entire subpattern. Drag the end ⬎ arrow to change the length of the subpattern. To remove a subpattern, click on and drag the start arrow to the upper left corner of the display. If two subpatterns are adjacent to each other, the end and start arrows may overlap in the display. Clicking on the overlapping arrows will select the lower-numbered subpattern arrow. If you need to change the position of the higher-numbered subpattern arrow, you can temporarily move the lower-numbered arrow first, or else use the **Edit Subpattern** window.

Subpattern definitions may not overlap; a bit position can belong to only one subpattern at a time.

3.2.4. Markers

Markers can be added to the display to help measure time intervals in your waveform pattern. To add a marker to the display, drag the green arrow from the upper left corner across the top of the display. You can have as many markers as you need. To remove a marker, drag its green arrow to the upper left corner.

3.3. Updating Pattern Generator Settings

Pulse Instruments pattern generators operate using several sets of settings:

- **Patterns**—the bit patterns (waveforms) and subpattern definitions
- **Instructions**—the output, looping, and repeating instructions
- **Formats**—controls for inverting, RZ-ing, and tri-stating outputs, as well as for setting delays
- **Clock**—internal/external clock source settings and the internal clock period

All these settings must be sent to the pattern generator in order for it to operate. This data can be created in PI-PAT, running on the Local instrument or on your remote PC, or it can be created in your custom application and sent to the instrument over GPIB.

After the pattern generator receives the settings, they must be “written” to the hardware before they will take effect. This permits the user to experiment with changes and discard errors before affecting the output.

All settings must be written to hardware before the initial command to **Run**, but if subsequent changes to the operating settings are made, only the changed data needs to be written to hardware.

3.3.1. Local Operation

In Local mode, all settings are written to the hardware over the CompactPCI bus. This mode of communication is very fast, even with large amounts of data. The **Sync** button is used to write all settings to hardware.

You should always **Sync** your data at least once before attempting to **Run** your pattern.

3.3.2. Remote Operation

The Sync button operates equivalently in Remote operation. It causes all settings to be sent over the GPIB to the PI-2005 and then causes the PI-2005 to write those settings to hardware. You can always use **Sync** from your remote PC to synchronize your pattern generator settings.

In applications involving very large patterns, the user may wish to use several other buttons provided for sending settings to the instruments. Remote operation relies on GPIB for communication with the PI-2005, and GPIB is several orders of magnitude slower than the PCI bus. If you have already **Sync**'ed all the pattern generator settings, and you subsequently change only the clock period, for example, you need not re-send the entire set of pattern data. The following buttons can be used to send only a subset of settings to the PI-2005:

- **Update**—sends and writes pattern data and subpattern definitions only. The **Update** button is available in the PI-PAT main window.
- **Compile**—sends and writes instructions only. The **Compile** button is available in the **Hardware** dialog box and in the **Edit Instructions** window.
- **Write Formats**—writes channel formats only. The **Write Formats** is available in the **Hardware** dialog box and in the **Edit Formats** window.
- **Write Clock**—writes clock settings only. The **Write Clock** button is available in the **Hardware** dialog box.

If you have any doubt as to which settings you have modified, you should use the Sync button to ensure that all settings are updated on the pattern generator.

3.4. Menu Items

3.4.1. File

The **File** menu will allow you to specify a **Mnemonic Table** File; create a **New** Waveform File; **Open** or **Close** an existing Waveform File; **Save (Save As…)** the current Waveform File; **Load**, **Store** and **Delete Hardware Files**; **Print** data; examine or edit the **Properties** of the current waveform file; **Exit** PI-PAT or **Shut Down 2005…** to power down the PI-2005. The **New**, **Open, Save** and **Print** functions are also available on the tool bar.

3.4.1.1. Load Mnemonic Table

The command is provided for compatibility with PI-DATS, Pulse Instruments' Data Acquisition and Test Software suite. This feature is not used in PI-PAT as a stand-alone application.

3.4.1.2. New

To create a new Waveform File, select **New** and then choose the type of document to create (e.g. PI-2005 or PI-2005x document). This will create another pattern window. The new file will be displayed with the default channel labels (1, 2, etc.) until you edit them. To close a pattern window select **File:Close** or click the Close Window ("**X**") button. If you have generated or edited a bit pattern in that window you will be prompted to save it.

3.4.1.3. Open...

To open an existing Waveform File, select **Open...**. A dialog box will be displayed allowing you to select any waveform file with the following extensions:

Extension	File Type
.w25	PI-2005
.w25x	PI-2005x

This will open an existing Waveform file in a separate pattern window. To close a pattern window select **File:Close** or click the Close Window ("**X**") button. If you have generated or edited a bit pattern in that window you will be prompted to save it.

3.4.1.4. Close

This will close the close the current window. If you have generated or edited a bit pattern in that window you will be prompted to save it.

3.4.1.5. Save/Save As...

To save the changes you have made to a file, select **Save**. If you wish to keep the original file intact, you may save the current setup in a different file by choosing **Save As....** The software will display a dialog box allowing you to specify a path and filename. The software will assign the extension .w25 or .w25x to your file unless you specify a different extension. If you have opened a .w58 (PI-5800) or a .w20 (PI-2000) file and **Save As...** with a .w25 extension, the file will be converted to a PI-2005 formatted file. See **Section 6. PI-5800A Compatibilities** for details about converting files.

3.4.1.6. Import . . .

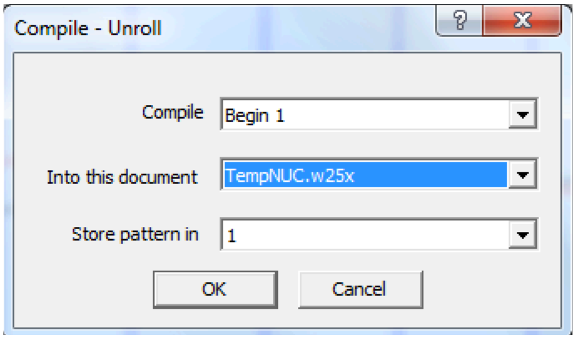
This function can be used to import a text file containing commands and pattern data in a simple text format for the PI-2005. For details on the format and structure of text commands, see **Section 7.4. Programming Methodology**.

3.4.1.7. Export . . .

This function can be used to export the current PI-2005 commands and pattern data to a text file. See **Section 7.4. Programming Methodology** for details.

3.4.1.8. Compile/Unroll . . .

This function can be used to compile and unroll a repetitive pattern program from one pattern file (the “Source”) into a single subpattern of a destination pattern file (the “Target”):



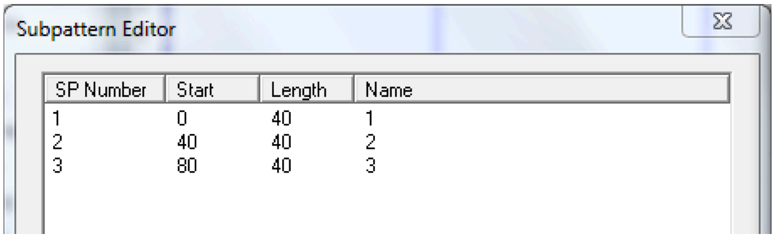
Compile: Begin 1

Into this document: TempNUC.w25x

Store pattern in: 1

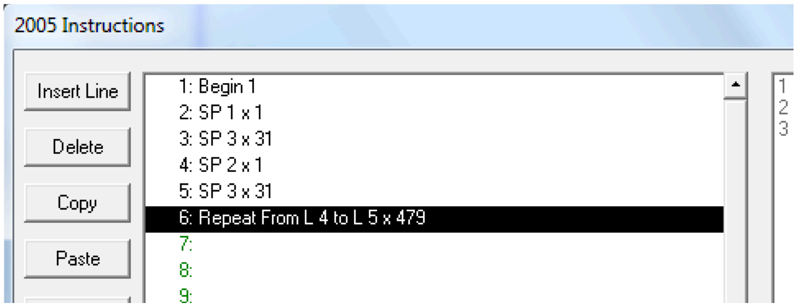
Figure 12: Compile/Unroll . . . Dialog Box

The target file must already have a subpattern defined that is the same length or longer than the resulting unrolled program. For example, consider the following subpatterns and the following instruction set:



SP Number	Start	Length	Name
1	0	40	1
2	40	40	2
3	80	40	3

Figure 13: Sample Subpattern Definitions



2005 Instructions

1: Begin 1
2: SP 1 x 1
3: SP 3 x 31
4: SP 2 x 1
5: SP 3 x 31
6: Repeat From L 4 to L 5 x 479
7:
8:
9:

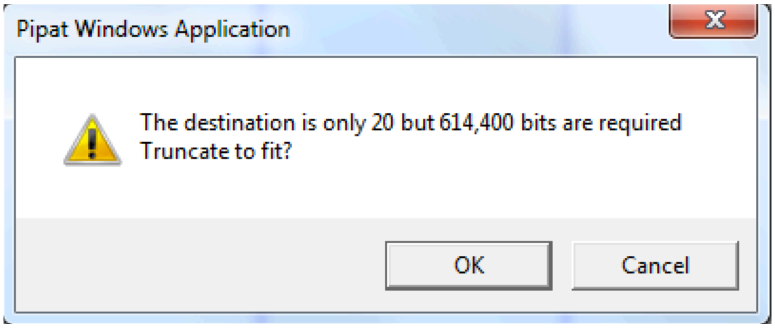
Figure 14: Sample Instruction Set

Compilling and unrolling Begin 1 would require:

(40 x 1) + (40 x 31) +

(((40 x 1) + (40 x 31)) * 479) = 614,400 bits of pattern memory in the target subpattern.

If the target subpattern does not have sufficient length, PI-Pat will ask if the data should be truncated. The user has the option of truncating or canceling and then re-defining the subpattern with the appropriate length.



Pipat Windows Application

The destination is only 20 but 614,400 bits are required
Truncate to fit?

Figure 15: Compile/Unroll Error Dialog

3.4.1.9. Printing

The software can print some or all of the contents of the open Waveform File. Select **Print...** to print the pattern data. A dialog box will appear, allowing you to define the print range. Select **Print Instructions...**, **Print Subpatterns...**, **Print Formats...** to print out the other Pattern Generator parameters. **Print Preview** and **Print Setup...** are used to view the output before it is printed and to set up the printer.

3.4.1.10. Properties

Allows the user to view the file type, date created and date modified, author, and to attach notes to the file.

3.4.1.11. Exit

Select **Exit** to leave PI-PAT and return to Windows. If you have not yet saved the Waveform File currently on the screen, you will be prompted to do so. When PI-PAT is closed on the local machine, the PI-2005 will stop and all pattern output will cease.

Closing PI-PAT on a remote computer will not stop pattern output on a connected PI-2005 unless a **Stop** command or **Shut Down PI-2005** command is issued first from the remote computer.

3.4.1.12. Shut Down PI-2005…

This selection will prepare the PI-2005 for powering off. A dialog box will ask you to confirm that you want to shut down. When the system is shut down Windows will display a message that it is safe to turn off the computer. This will take approximately 15 seconds and allows the system software and operating system to close.

If you are controlling a PI-2005 from a remote computer, the **Shut Down 2005 . . .** command will cause the local PI-2005 to prepare for powering down. When PI-PAT on the local computer has shut down, the green LED on the front panel of each Pattern Card will turn off. Once those LEDs have turned off, allow approximately 30 seconds for the internal operating system of the PI-2005 to finish closing before turning off the power.

NEVER turn the power off without first performing a Shut Down 2005…

3.4.2. Edit

The **Edit** menu allows you to edit the displayed waveforms and switch to other available editing modes. You can select **Undo** to reverse a change to the pattern information. You can use the **Copy**, **Cut**, **Paste, Multiple Paste…** and **Delete** functions to move blocks of data much as you would blocks of text in a word processor. The **Multiple Paste…** function provides the ability to paste multiple times by entering a repeat value. The **Copy**, **Cut** and **Paste** functions are also available on the tool bar. You can select the **Subpatterns…**, **Instructions…**, or **Formats…** menus to edit any characteristic of a signal. The **Pattern Special...** option allows you to select a predefined pattern of **CLK/2** or **NOT CLK/2** for a selected channel. The **Go To…** menu provides an easy way to move the cursor to a desired location. This menu can also be accessed through the **Go To** button on the toolbar.

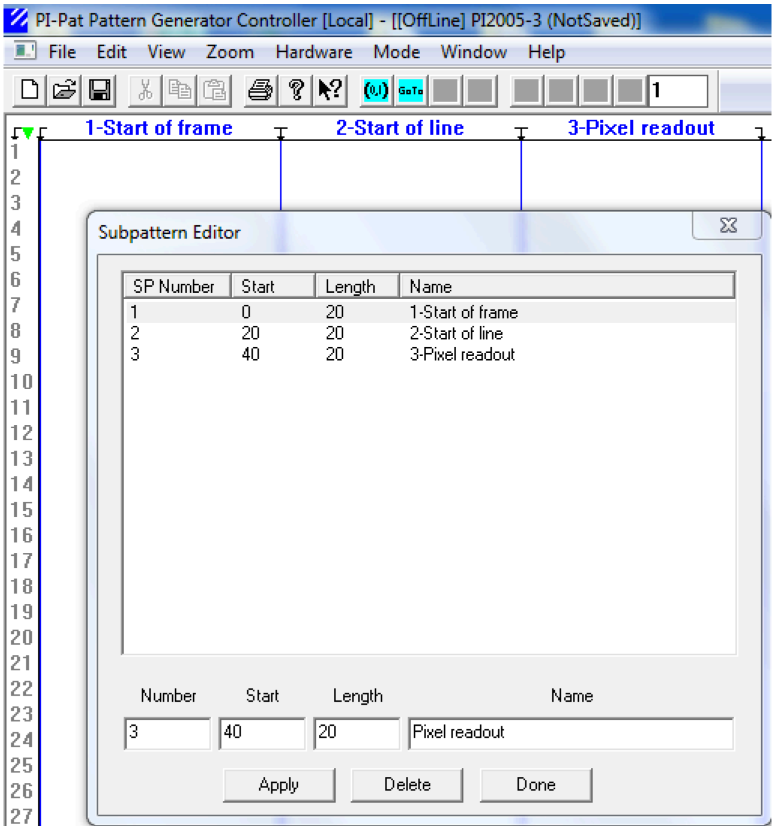
3.4.2.1. Edit Subpatterns ...

Select **Edit Subpatterns ...** to display the **Subpattern Window**, allowing you to edit the start bit, length, and name of any subpattern.

To edit a subpattern, select it by clicking on its definition in the Subpattern list. You can then type its new start bit and length in the boxes at the bottom of the window. To enter the subpattern, click **Apply**.

To create a new subpattern, type any unused number in Number field at the bottom of the window. By default the Start bit will be populated with the bit position after the last defined Subpattern, with a length equal to the previous Subpattern’s length, and with its Subpattern number as its default name. Enter a length and/or edit any of the other settings, then click **Apply**.

To delete a subpattern, select it and click the **Delete** button.



SP Number	Start	Length	Name
1	0	20	1-Start of frame
2	20	20	2-Start of line
3	40	20	3-Pixel readout

Number	Start	Length	Name
3	40	20	Pixel readout

Figure 16: Edit Subpattern Window

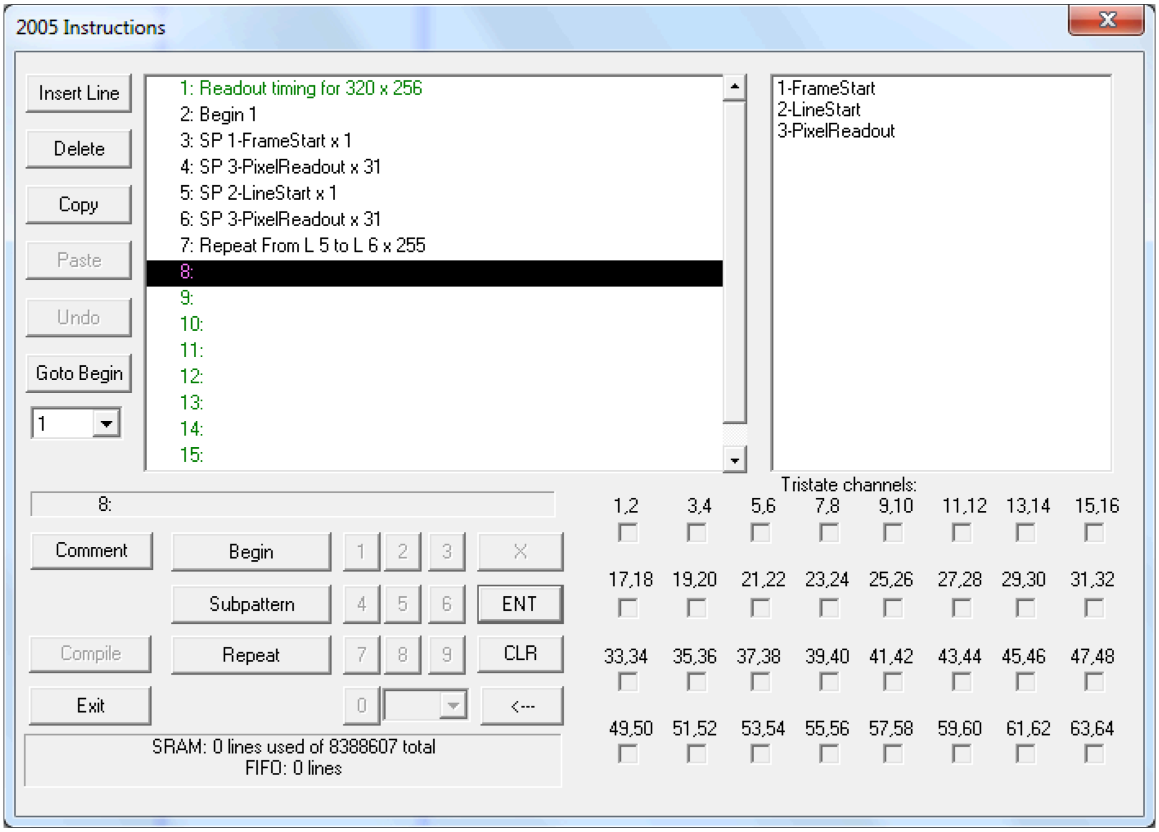
When you are finished click **Done** to close the Subpattern Window and return to the Main Window. If you have made changes to any of the Subpattern definitions, be sure to click **Apply** before clicking **Done**.

3.4.2.2. Edit Instructions ...

Select **Edit Instructions ...** from the **Edit** menu to view the Instruction Window containing the programs for the current waveform. Once pattern data and subpatterns have been defined, the Instruction window is used for establishing the sequence of subpatterns that will be output for a specific pattern application or program.

PI-PAT’s instruction format allows the user to make the tradeoff between complexity and flexibility. In the simplest case, the user can define one long subpattern and run it continuously. In a more complex case, the user may define as many as 262,144 different subpatterns to be intermixed as required. Multi-level looping, re-usable subpatterns, multiple program sets, and features like on-the-fly Tri-Stating let you create highly complex patterns with just a few lines of code.

Although PI-PAT’s instruction format is simple and intuitive, you may want to read through **Section 4. Tutorial – How to create and run a program** for an example of how to create PI-PAT programs.



2005 Instructions

1: Readout timing for 320 x 256
2: Begin 1
3: SP 1-FrameStart x 1
4: SP 3-PixelReadout x 31
5: SP 2-LineStart x 1
6: SP 3-PixelReadout x 31
7: Repeat From L 5 to L 6 x 255
8:
9:
10:
11:
12:
13:
14:
15:

1-FrameStart
2-LineStart
3-PixelReadout

8:

SRAM: 0 lines used of 8388607 total
FIFO: 0 lines

Figure 17: Instruction Window

Select an instruction line by clicking on it with the left mouse button. You can select multiple lines by holding the SHIFT key down and clicking the left mouse button on the end of the group, or else highlight multiple lines by dragging with the right mouse button. As you enter or edit instruction lines, command buttons will be enabled or disabled as appropriate. If you enter an instruction incorrectly, you will receive an error message. Use **CLR** to correct errors, then re-enter the instruction. When you have finished entering or editing instructions, click **OK** to close the Instruction Window and return to the Main Window.

3.4.2.2.1. Begin statement

Every PI-PAT program begins with a BEGIN statement, and so each program is identified by its “Begin number.” Multiple programs can be stored in the same file simply by entering multiple BEGIN statements.

To enter a new Begin statement on the selected line, click **Begin**, and then enter a Begin number. Click **ENT** to complete the entry. The Begin number can be any number that is not already in use. The first use of the begin statement is typically Begin 1.

You can also enter a Begin statement by typing “**b**”, a number, and **Enter**.

A **Begin** statement ends with the first empty line or when another **Begin** statement is encountered.

3.4.2.2.2. Subpattern statement

To enter a Subpattern statement on the selected line, click **Subpattern**. Enter a Subpattern number (or select it from the right portion of the window), click **X**, click the number of times it is to be repeated, then click **ENT** to complete the entry. The maximum Subpattern number is 262,144 (the maximum number of Subpatterns that can be defined) and the maximum repetition factor on a given instruction line is 32,767. If additional repetitions are required, use a Repeat statement.

You can also enter a Subpattern statement by typing "**s**", then a subpattern number, "**x**", the repeat factor, and **Enter**.

3.4.2.2.3. Repeat statement

A Repeat statement causes PI-PAT to repeat (loop) a contiguous block of instructions immediately preceding the Repeat statement. For example, a Repeat statement on line 10 can cause PI-PAT to repeat lines 2 through 9, or 6 through 9, etc.

To enter a Repeat statement on the selected line, click **Repeat**. Enter the starting instruction line of the repeat loop, click **X**, enter the number of times the loop is to be repeated, then click **ENT** to complete the entry. The statement "Repeat from line 2 to line 3 x 7" means: Execute lines 2 and 3, and then repeat 7 times.

You can also enter a Repeat statement by typing "**r**", the instruction line of the loop start, "**x**", the repeat factor, and **Enter**.

The **Repeat** button allows any group of subpatterns to be repeated in a single statement up to 32,767 times. If additional repetitions are required, Repeat loops can be "nested." For example, the following short program will output subpattern 1 over 100 million times.

```
1 Begin 1
2 Subpattern 1 x 100                ┐           ┐
3 Repeat From L 2 to L 2 x 1023    ┘Inner loop │Outer loop
4 Repeat From L 2 to L 3 x 1023                ┘
```

If you are controlling the PI-2005 via PI-DATS, the pull down menu next to the **<-** button, can be used to vary the number of repeats per line based on a variable. This feature is not available separately in PI-PAT.

3.4.2.2.4. Inserting and Deleting Instruction Lines

Click **Insert Line** or type the **Ins** key to add a line or lines. The selected line or lines will be moved down, and all line numbers will be updated.

Click on a line to select it, then click **Delete** or type the **Del** key to delete the selected line from the program. All subsequent lines will be moved up and renumbered accordingly, as will line references in repeat loops. You cannot delete all the lines referenced by a **Repeat** instruction without first deleting the Repeat instruction. Care must be taken with this command as it can affect looping parameters.

Click **Copy** [Ctrl-C] to select a line or group of lines to be copied. If you are copying a group of lines that include a **Repeat** command, all lines referenced by the **Repeat** command must be included in the group. After moving the cursor and clicking the left mouse button to select a line in the program, click **Paste** [Ctrl-V] to paste the previously copied instruction or instructions to that location. All subsequent lines will be moved down and renumbered accordingly, as will line references in repeat loops. Care must be taken with this command as it can affect looping parameters. If a **Begin** line is included in a copy and paste, the Begin number will be updated to the next available begin number.

3.4.2.2.5. Tri-State On-The-Fly (TSOTF)

The PI-2005 has a new feature that allows the user to TRI-STATE channel pairs on-the-fly, during instruction execution.

While the output format of a channel can be tri-stated statically in the **Edit Formats** window, Tri-State on-the-fly allows a channel pair to be tri-stated dynamically, for the duration of a subpattern instruction. A user can use the same subpattern definition repeatedly and change the output format of each channel pair at run-time, instead of at program time. This can conserve channel assignments and allows for much more compact pattern definition.

After entering a subpattern statement in the **Edit Instructions** window, you can select multiple channels for tri-stating by holding the left mouse button down and dragging it over multiple check-boxes. Click on any selected checkbox to check or un-check all selected channels. After checking the TRI-STATE box(es) click on ENT to apply the change.

The program instructions will display the first channel in each channel pair that will be tri-stated during program execution. For example, if line 2 outputs subpattern 10 twelve times, and channels 1 and 2 are to be tri-stated during that time, line 2 would read:

```
2 Subpattern 10 x 12 T1
```

See **Section 5. Tri-State On-The-Fly Notes** for important notes about using this feature.

3.4.2.2.6. Comment

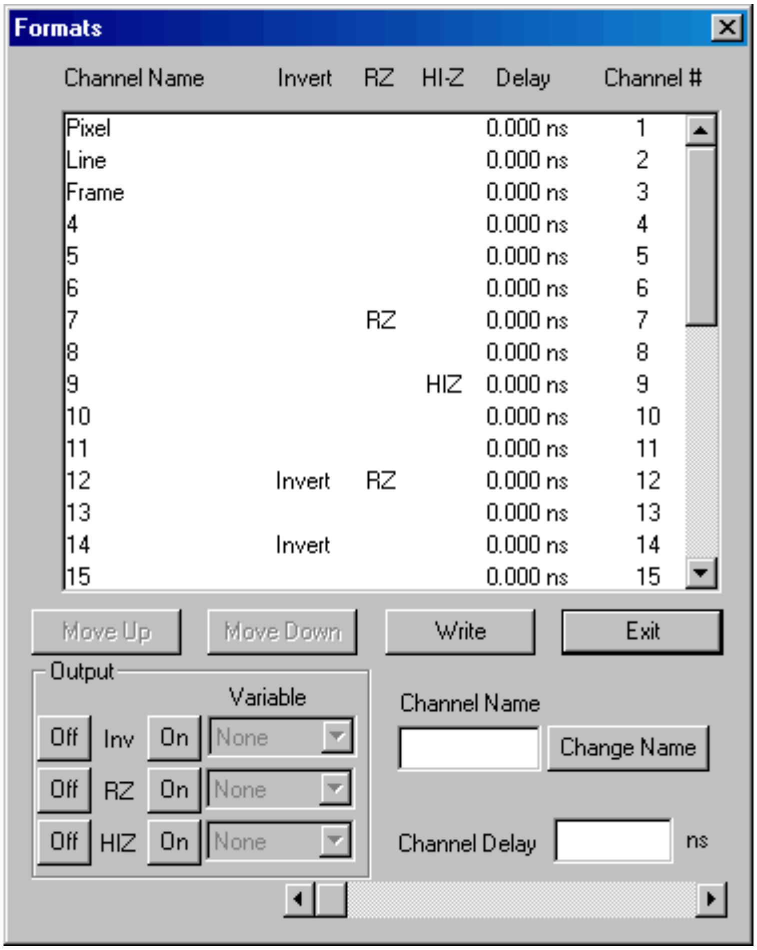
Place the cursor (click) on an empty line, and click on **Comment**, to insert a comment. This line will be ignored by the compiler. A comment can also be inserted by double clicking on an empty line.

3.4.2.2.7. Compile x

When editing instructions, click on **Compile x** to program the changes made in this window to the hardware. It has the same effect as the **Compile** button in the **Hardware** window. The Begin programmed to the hardware is the one where the cursor (the highlighted line) is

3.4.2.3. Edit Formats ...

Choosing **Edit Formats ...** from the **Edit** menu will display the **Formats Window**, which allows you to change the name and output format of each channel.



Formats

Channel Name	Invert	RZ	HI-Z	Delay	Channel #
Pixel				0.000 ns	1
Line				0.000 ns	2
Frame				0.000 ns	3
4				0.000 ns	4
5				0.000 ns	5
6				0.000 ns	6
7		RZ		0.000 ns	7
8				0.000 ns	8
9			HIZ	0.000 ns	9
10				0.000 ns	10
11				0.000 ns	11
12	Invert	RZ		0.000 ns	12
13				0.000 ns	13
14	Invert			0.000 ns	14
15				0.000 ns	15

Figure 18: Edit Formats - Invert, RZ, HI-Z, Delay

To modify a channel, select it by clicking anywhere on its row.

The channel's name and its other format characteristics will be displayed at the bottom of the window.

The default channel names are displayed if they have not been edited in the main window. Change the channel name by typing a new name in the **Channel Name** box. Click the **Change Name** button to complete the entry.

Clear the **Channel Name** entry and click **Change Name** to reset the channel name to the default numeric setting (1, 2, 3, etc).

Set the Invert/Non-Invert, Tri-State, and RZ/NRZ (Return-to-zero/Not-return-to-zero) parameters for the selected channel by using the **ON** and **OFF** buttons next to the **Invert**, **RZ** and/or **HIZ** indicators.

If the programmable delay option is present on a card, its channels will display the programmed delay in nanoseconds. To change the delay of a channel, select the corresponding channel, and either type in a value in the **Channel Delay** box between 0 and 6.6 ns, or use the scroll bar under that box. Multiple channels can be selected at once by using the **Shift** key.

If the delay option is not present, the delay column will say **No Delay**, and when a channel is selected, the delay scroll bar and the delay window will be grayed out.

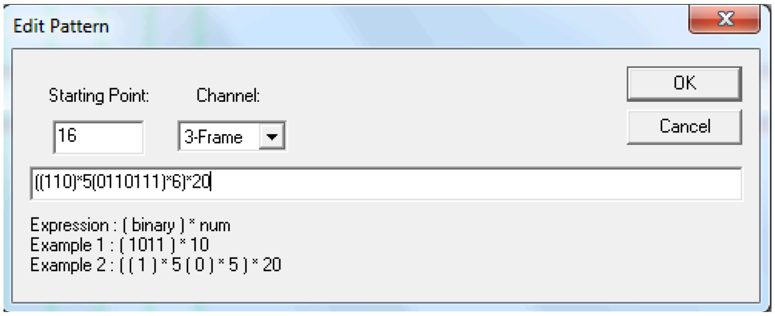
The location of the channel in the main menu display can be modified by selecting a channel and clicking on **Move Up** or **Move Down**. The last column in the format display shows the physical location of the channel. This information is used to restore the hardware order in the main menu display using the **View** option (**Section 3.4.3. View**).

Once you have finished modifying a channel, you can modify a different channel by selecting it, or click **OK** to close the **Format Window** and return to the **Main Window**. In order for the changes made in the **Edit Formats** window to take effect, you must

1. click **Write**, or
2. go to the **Hardware** window and click **Formats**, or
3. go to the **Hardware** window and click **Sync**.

3.4.2.4. Enter Pattern ...

Displays a dialog box allowing algorithmic entry of a repetitive sequence of bits, inserted at the present cursor location:



Starting Point: 16

Channel: 3-Frame

((110)*5(0110111)*6)*20

Expression : (binary) * num
Example 1 : (1011) * 10
Example 2 : ((1) * 5 (0) * 5) * 20

Figure 19: Enter Pattern dialog box

The basic syntax is a bit pattern of 1 and 0 characters, enclosed in parentheses, followed by an asterisk and a repeat factor, e.g.

(110)*5

Bit patterns can be concatenated by appending them, such as:

(110)*5 (0110111)*6

The Enter Pattern feature permits “nesting,” so a series of bit patterns can be enclosed in parentheses and repeated, such as:

((110)*5(0110111)*6)*20

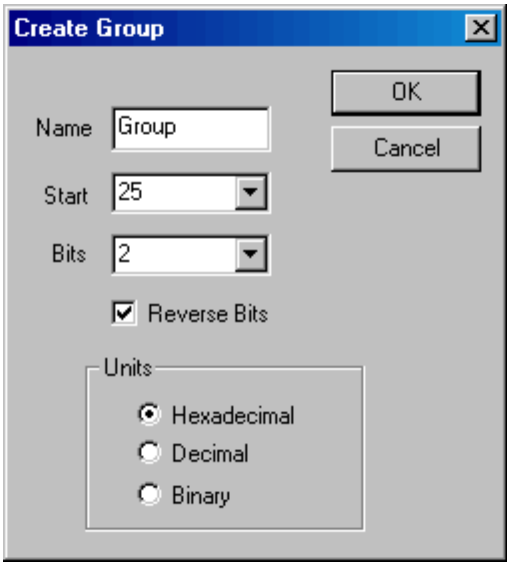
3.4.2.5. Pattern Special ...

Allows you to insert a **CLK/2** or **NOT CLK/2** pattern in the highlighted area. To highlight or select a section of the bit pattern, drag over the desired area with the *right* mouse button.

3.4.2.6. Create/Modify Group

Channels can be grouped together to create arbitrary waveforms and function outputs. The outputs can be displayed as an n-bit waveform, or as decimal, hex or binary data.

Click on any channel in the desired group, then choose **Edit:Create Group** or **Edit:Modify Group**. The Create and Modify group windows are similar:


Create Group

Name: Group

Start: 25

Bits: 2

☑ Reverse Bits

Units
● Hexadecimal
○ Decimal
○ Binary

Figure 20: Create or Modify Group

Enter a group name, the number of the starting channel, and the number of bits (channels) to include in the group.

Instead of the user-defined channel names, the Group name will be displayed in the Pattern Window along with the bit number of each channel.

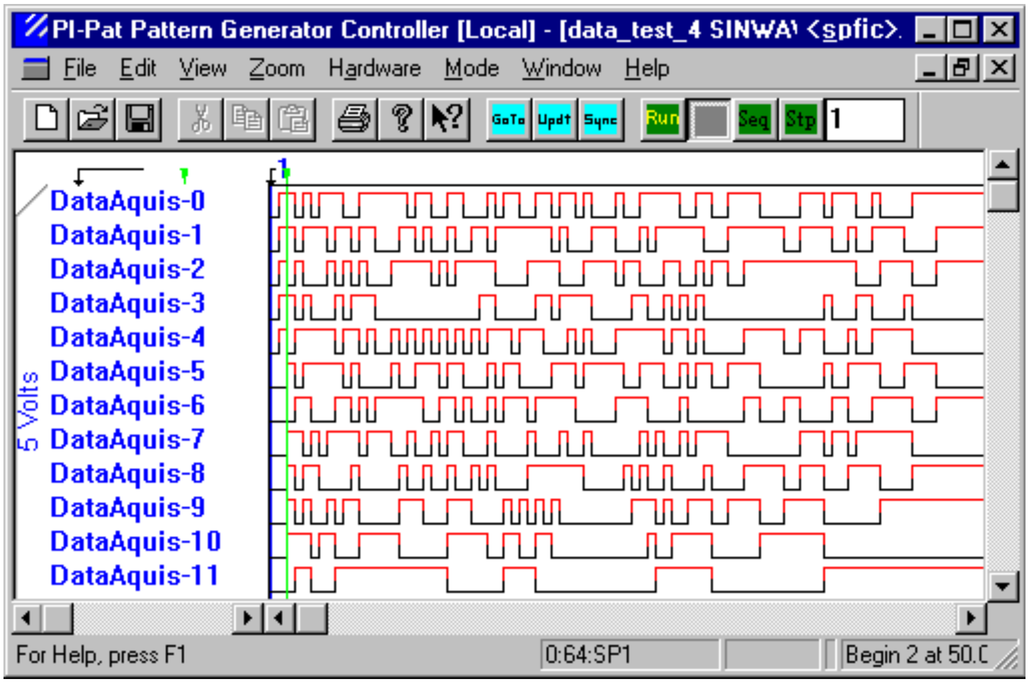


Figure 21: Grouped Channel Names

For example, if an 8-bit group named “DataAcquis” is defined for Channels 1-8, the Pattern Windows will display them as DataAcquis-7, DataAcquis-6, DataAcquis-5, etc. By default, bit numbers are assigned in descending order.

By checking the Reverse Bits box in the Create or Modify Group box, you can reverse the bit number assignment.

Bit number assignments are used when displaying multi-bit output as decimal, hex, or binary data. The largest numbered bit is the most-significant bit (MSB) and the lowest numbered bit is the least-significant bit (LSB). See **Section 4.5. Creating Arbitrary Waveforms** for a detailed description of groups.

3.4.2.7. Delete Group

This will delete the group definition of the group where the cursor is positioned. The pattern data will remain unchanged. PI-PAT will restore any user-defined channel names that were entered before the Group was created.

3.4.3. View

The **View** menu allows you to select whether the **Toolbar, Status bar,** and **Subpattern Lines** are displayed and to choose between the user-defined display order and the **Hardware Order**. When **Hardware Order** is checked, the channels will be displayed in the actual order they are present in the PI-2005. If **Hardware Order** is un-checked, then the order in which the channels will be displayed can be changed by moving the channels up or down in the **Edit Formats** window (See **Section**

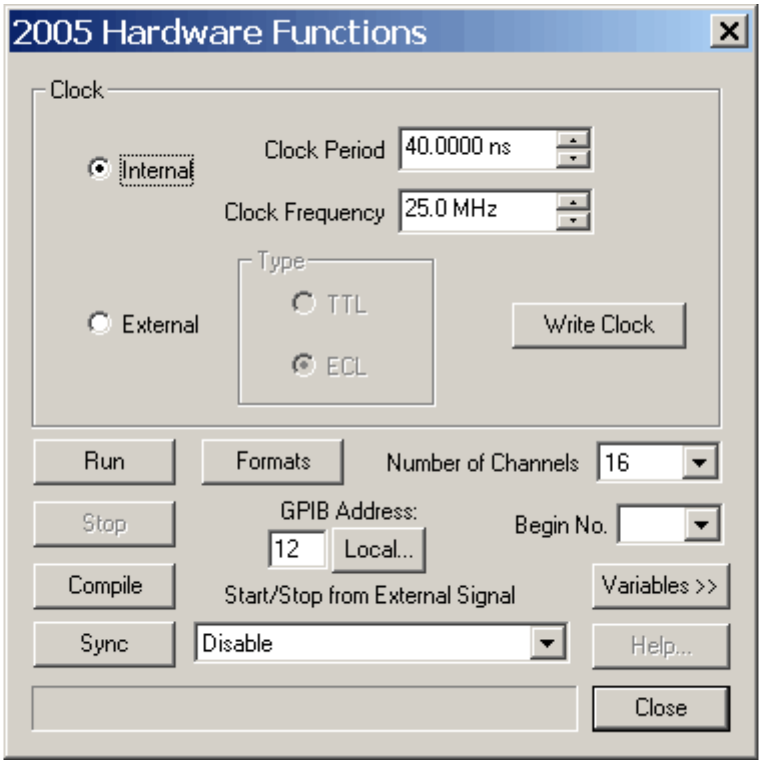
3.4.2.3. Edit Formats ...).

3.4.4. Zoom

From the **Zoom** menu you can select a horizontal magnification of the waveform display of **1X, 2X, 4X** and **8X**.

3.4.5. Hardware

The **Hardware** dialog box allows you to **Sync** parameter settings to the hardware. The Pattern Generator's System Settings such as the **Internal Clock Frequency/Period**; **External Clock Type** (**ECL** or **TTL**); **GPIB Address**; **Number of Channels** and **Start/Stop from External Signal** parameters are all set in this dialog box. This dialog box also contains buttons allowing the user to **Run**, **Stop**, or **Compile** the programs. For advanced users, who use PI-DATS test automation software, the **Variables** button allows the creation of test programs with variable Repeat statements.



● Internal
Clock Period 40.0000 ns
Clock Frequency 25.0 MHz
○ External
Type ○ TTL ● ECL
Number of Channels 16
GPIB Address: 12
Begin No.
Start/Stop from External Signal Disable

Figure 22: PI-2005 Hardware dialog box

3.4.5.1. Clock

This area of the PI-2005 **Hardware** dialog box is used to define the master operating clock. You can select either the **Internal** or **External** radio button to define the clock source.

For instructions on where to connect the external clock see **Section 2.3.5.2.1. External Clock In**. If the **Internal** radio button is selected you can set the master clock period and frequency for the system by typing either in the **Clock Period** or **Clock Frequency** box. Entering a value for the period will update the frequency displayed and entering a frequency will update the period displayed. The maximum clock frequency is 225 MHz (4.44nsec. period). Click the **Write Clock** button to complete the entry and change the clock period/frequency.

If the **External** radio button is selected, the levels of input clock must be set. This is done by selecting either the **TTL** (for TTL thresholds) or **ECL** (for ECL thresholds) radio button. If the external clock is selected, a clock **must** be connected and have a frequency > 0.1Hz for the PI-2005 to operate. No program will run without a clock present.

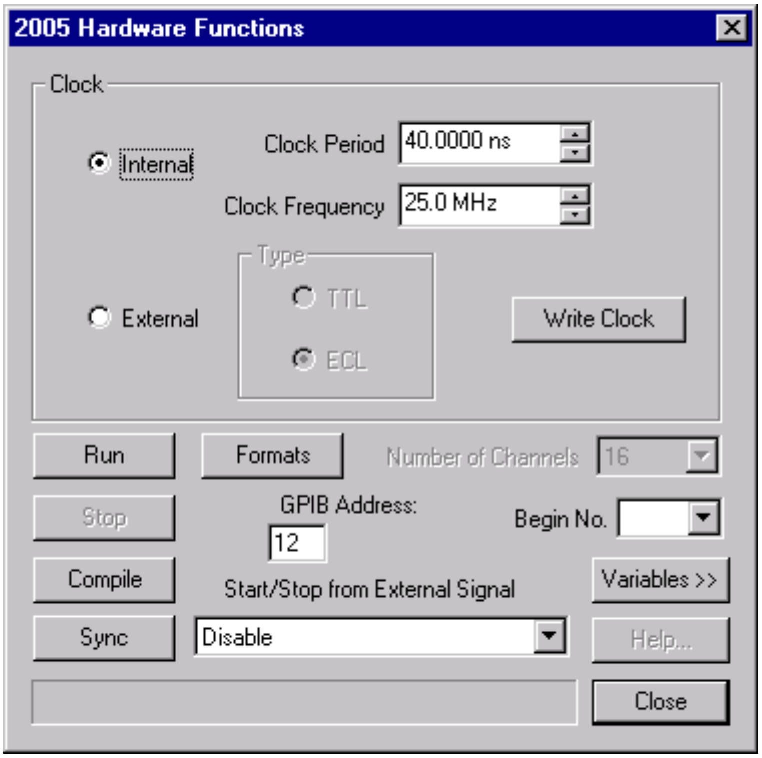
3.4.5.2. GPIB Address

By default, each PI-2005 with the optional IEEE-488 (GPIB) interface is configured with a default GPIB address of 14, and with the System Controller mode set to Off. This enables you to connect the PI-2005 to an established IEEE-488 bus and operate it from a remote computer.

The GPIB address of a PI-2005 can be changed either in Local operation or from Remote operation via the **Hardware** menu.

3.4.5.2.1. Changing GPIB address Locally

To change the GPIB address in Local operation, click the Hardware menu to bring up the Hardware dialog box:



2005 Hardware Functions

Clock

● Internal

Clock Period: 40.0000 ns

Clock Frequency: 25.0 MHz

○ External

Type: ○ TTL ● ECL

Number of Channels: 16

GPIB Address: 12

Begin No.

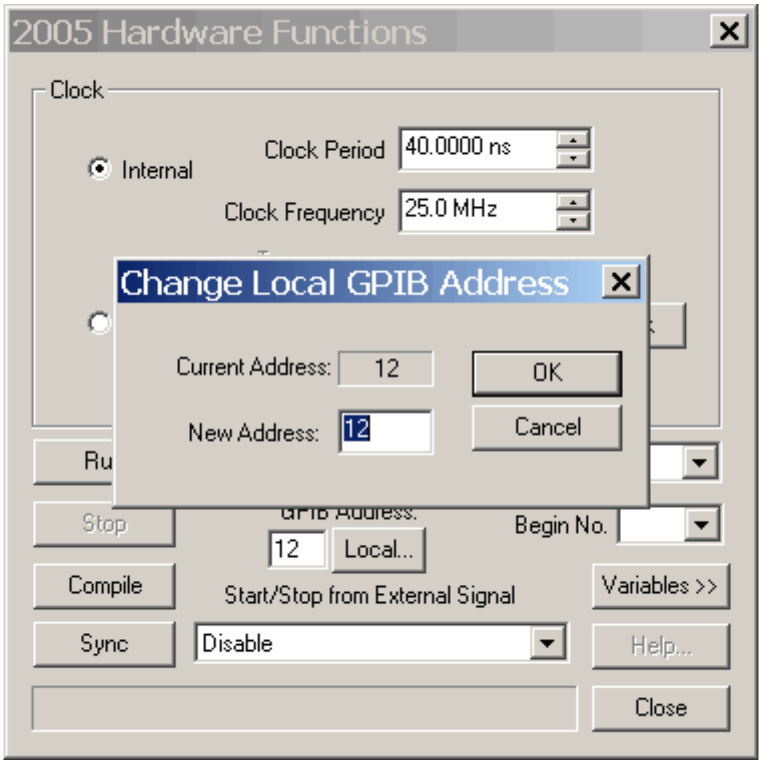
Start/Stop from External Signal: Disable

Figure 23: PI-2005 Hardware dialog box

Type the desired GPIB address in the **GPIB Address** box and click **Close**. The PI-2005 will now respond to commands sent to the new address.

3.4.5.2.2. Changing GPIB address Remotely

To change the GPIB address of the PI-2005 from your remote controller, click the Hardware menu to bring up the Hardware dialog box, then click on the **Local . . .** button to bring up the **Change Local GPIB Address** dialog box:



2005 Hardware Functions

Clock

● Internal

Clock Period: 40.0000 ns

Clock Frequency: 25.0 MHz

Change Local GPIB Address

Current Address: 12

New Address: 12

GPIB Address: 12

Start/Stop from External Signal: Disable

Figure 24: PI-2005 Hardware dialog box

Type the desired GPIB address in the **GPIB Address** box and click OK. The PI-2005 will now respond to commands sent to the new address.

3.4.5.3. Number of Channels

This displays the **Number of Channels** in the PI-2005. If you are using PI-PAT to control the PI-2005 from another computer or create timing files offline, the number of channels can be modified. The number of channels entered must be equal to or less than the number of channels physically available. PI-PAT will only define data for the channels defined in the software. If more channels are defined in PI-PAT than are available in the hardware, the data for those channels will not be written to the hardware. The channels that are removed are the higher number ones.

3.4.5.4. Sync

Clicking on **Sync** updates all hardware settings and timing information in the PI-2005. This should be clicked after any changes are made. **Sync** preserves the current run-mode, so if the PI-2005 is in **Run** when **Sync** is clicked, it will stay in **Run** mode. The **Sync** button is also available on the toolbar of the **Main Window**.

3.4.5.5. Compile

Clicking **Compile** causes the selected program to be downloaded to hardware and run. By default, the number in the drop-box will be the number of the most recently run program. To change the program to be run, use the drop-down button and select a different number from the list of available programs. This command does NOT program the subpatterns or the patterns; it only programs the instructions. Also, if the program was running when the **Compile** button was clicked, the new program will be re-started automatically after the new instructions are programmed.

3.4.5.6. Formats

Clicking on **Formats** updates the information present in the **Edit Formats** window: **Invert/NonInvert, RZ/NRZ, HI-Z, Delay**. If the program was running when the **Formats** button was clicked, the program will be re-started automatically after the new formats are programmed.

3.4.5.7. Run

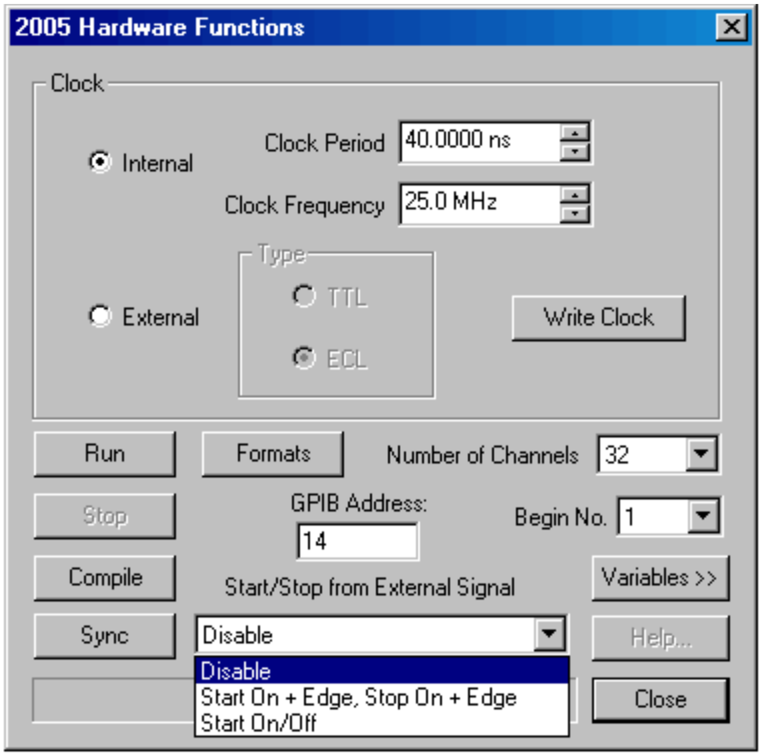
Clicking **Run** puts the Pattern Generator in the Run mode unless the **Start from External Signal** is selected (see **Section 3.4.5.9. Start/Stop from External Signal**).

3.4.5.8. Stop

Clicking **Stop** puts the Pattern Generator in the Stop mode and resets the program counter.

3.4.5.9. Start/Stop from External Signal

This pull down menu is used to enable the external Start/Stop parameters. The Start and Stop signals can be connected to the External Clock connector. The default is to have this function disabled. If enabled, the PI-2005 will start when a TTL high is detected on the Start signal after clicking on the Run button. A display window will indicate the system is waiting for a Start signal.



2005 Hardware Functions

Clock

● Internal

Clock Period 40.0000 ns

Clock Frequency 25.0 MHz

○ External

Type: ○ TTL ● ECL

Number of Channels 32

GPIB Address: 14

Begin No. 1

Start/Stop from External Signal: Disable

- Disable
- Start On + Edge, Stop On + Edge
- Start On/Off

Figure 25: Start/Stop from External Signal

The options in the **Start/Stop from External Signal** window are:

- **Disable** — the Start comes from the local source (e.g. **Start** button or Start command over GPIB)

- **Start on + Edge, Stop on + Edge** — a 1 CLOCK pulse on the EXTERNAL START input will start the program, and a 1 CLOCK pulse on the EXTERNAL STOP input will stop it. It does not matter what the status of the EXTERNAL START signal is; once the pulse has started the program a pulse on STOP will stop the program.
- **Start On/Off —** a HIGH level on the EXTERNAL START input will start the program, and a LOW level on the same input will stop it. In this mode the EXTERNAL START signal acts like a GATING signal for the program: when it is HIGH the program runs, and when it is LOW, the program stops.

The latency between the external start signal and the output of the pattern generator is approximately 40 master clock periods, +/- one clock period.

After the Pattern Generator receives a Stop (either a Stop pulse or a LOW on the Start line), the unit will be in Stop mode, and another Run command must be sent to “rearm” it for the next external Start.

3.4.6. Mode

PI-PAT can operate in several different modes:

- **Remote GPIB**—For running on a remote computer, controlling a Local PI-2005 over GPIB. You can also run PI-PAT in Remote GPIB mode if you are working “offline” (e.g. not connected to a PI-2005) and creating pattern files for later use. This requires the computer running PI-PAT to have a GPIB interface installed, running the standard 488.2 driver.
- **Local GPIB**—For running on a Local PI-2005, to be controlled by a Remote computer. This is the default mode for PI-PAT installed on your PI-2005. This requires the computer running PI-PAT to have a GPIB interface installed, running the standard 488.2 driver (PI-PAT versions x.xx or higher) or NI-Device (PI-PAT versions x.xx or lower).
- **Local Only**—For running on a Local PI-2005 with an attached monitor, keyboard and mouse. This mode also “decouples” PI-PAT from the GPIB interface, allowing the GPIB interface to be used by other software (including PI-DATS) to control other hardware.

To choose the PI-PAT operating mode, select from the **Mode** window. **Remote GPIB** and **Local GPIB** modes may require configuration of the GPIB interface, as detailed below.

3.4.6.1. Remote IEEE-488 Interface Setup

3.4.6.1.1. Version x.xx and Later

In later versions of PI-PAT the same National Instruments 488.2 driver is used for all modes.

3.4.6.1.2. Version x.xx and Prior

In earlier version of PI-PAT there are two different drivers used for Local GPIB and Remote GPIB modes.

The standard NI 488.2 driver is used when the PI-2005 is used as a system controller.

For operation as a GPIB instrument, the PI-2005 uses a driver package named NI-Device.

The two driver packages are mutually incompatible, so one must be completely uninstalled before installing the other. After uninstalling one driver, reboot the system before installing the other driver, even if the uninstaller does not prompt you to reboot. If both drivers are inadvertently installed simultaneously, both must be uninstalled, one at a time, with a reboot after each installation, before the desired driver is reinstalled again.

3.4.7. Window

This window provides standard Windows layout options such as **Cascade, Tile** and **Arrange Icons** as well as **New**, **Duplicate** and **Split** to make editing bit patterns easier.

3.4.8. Help

The help file is not presently available, although this Operators Manual is available in Portable Document Format (.pdf) on your PI-PAT CD, and is also available for download from:

http://www.pulseinstruments.com

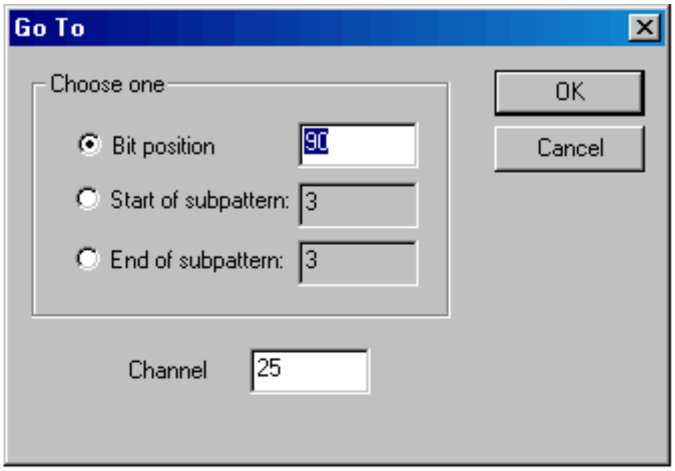
The **Upgrade Application** option on this menu is available to add access to additional Pulse Instruments software such as PI-CONTROLLER, PI-CONTROLLER PLUS and PI-DATS by entering passwords provided by Pulse Instruments.

3.5. Toolbar Functions

Standard functions such as **New, Open, Save, Cut, Copy, Paste, Print**, and **Help** are available in the toolbar. A **Go To** button is available to move the cursor to a specific location in pattern memory. The remaining buttons in the toolbar control the operation of the PI-2005.

3.5.1. Go To

The **Go To** button is used to move the cursor to a specific location in the bit pattern. Clicking on the **Go To** opens the Go To dialog box.



● Bit position 90

○ Start of subpattern: 3

○ End of subpattern: 3

Channel 25

Figure 26: Go To Dialog Box

The position of the cursor can be specified as a bit postion, the start of subpattern or the end of a subpattern for any channel specified. The default value for the Bit position is the present position of the cursor. The default value for the Start and End of subpattern is the last subpattern.

3.5.2. Sync

Clicking the **Sync** sends all data and settings to the PI-2005 hardware and preserves the current run-mode. If you are currently running a pattern, you can edit bits and subpattern definitions in the **Main Window** and quickly see the effects of your changes in the output by clicking the **Sync** button. The **Sync** button is also available in the **Hardware** dialog box.

3.5.3. Run

Clicking the **Run** button will start the Pattern Generator running the program specified by the “Begin No” in the **Hardware** window. If the Pattern Generator is already running this button is grayed out.

3.5.4. Stop

Clicking the **Stop** button will stop the Pattern Generator and reset the Pattern Generator to the beginning of the timing sequence. If the Pattern Generator is in **Seq** or **Step** mode **Stop** will reset the program counter.

3.5.5. Seq

Clicking the **Seq** button will cause the Pattern Generator to sequence through the entire program specified by the Begin No as many times as indicated in the box next to the button. If the Pattern Generator is running this menu is grayed out.

3.5.6. Stp

Clicking the **Stp** button allows the user to step through the number of subpatterns indicated in the box next to the button. If the Pattern Generator is running this menu is grayed out.

4. TUTORIAL – HOW TO CREATE AND RUN A PROGRAM

The PI-2005 is programmed by inputting pattern data, defining subpatterns, and defining programs to output those subpatterns. Pattern data consists of binary data entered into the 64 Kbits of pattern memory for each channel. A subpattern associates a range of bits with a unique subpattern number. A subpattern can be from 20 bits to 64 Kbits long, and up to 3,276 subpatterns can be defined.

The output data stream is defined by instruction or program steps that control the execution sequence and repeating of subpattern data. This technique provides virtually endless data pattern depth. Each program starts with a "BEGIN" statement and a program number, and thus each program is identified by its "BEGIN number." Multiple programs can be stored in a single file with all their associated pattern data and subpattern definitions.

4.1. Defining subpatterns

The first step in creating a program is defining the subpatterns. The subpatterns are groups of bits that can be arranged to be executed (generated) in a certain order, by creating Instructions.

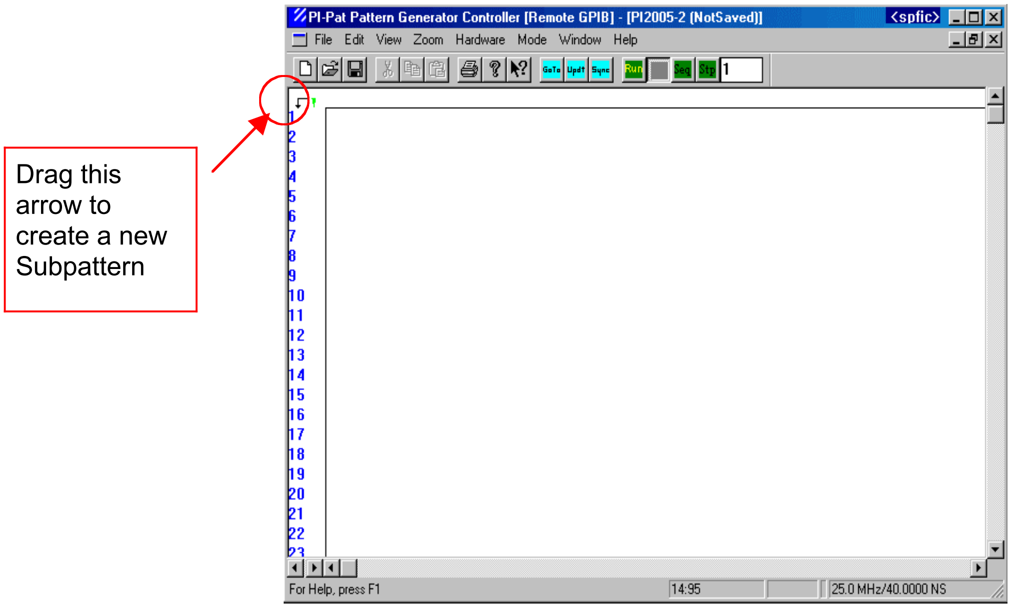
First the size of the subpattern must be defined. This can be achieved in 2 ways:

4.1.1. Numerical editing of Subpatterns

Section 3.4.2.1. Edit Subpatterns ... shows how to define the limits of a subpattern by editing the starting bit of the subpattern and the length of it.

4.1.2. Graphical editing of Subpatterns

For a quicker and sometimes more intuitive editing, the graphical interface can be used. In the main PI-PAT window, drag the right-angled black arrow in the upper left corner onto the top border of the Pattern Window:



Drag this arrow to create a new Subpattern

Figure 27: Graphical Editing of Subpatterns

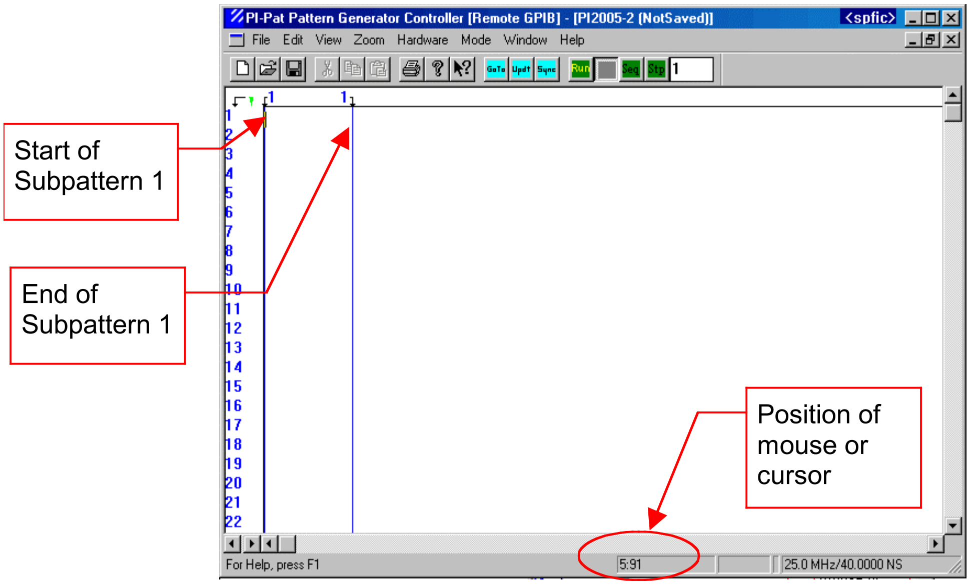

PI-Pat Pattern Generator Controller [Remote GPIB] - [PI2005-2 (NotSaved)]
<spfic>
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp 1
Start of Subpattern 1
End of Subpattern 1
Position of mouse or cursor
For Help, press F1
5:91
25.0 MHz/40.0000 NS

Figure 28: New Subpattern Created

PI-PAT will create a new subpattern with a default length of 12 bits or 20 bits, the minimum length allowed, depending on the installed pattern cards.

To move the subpattern to a different location, one can either:

a. Go to the **Edit Subpatterns** window (**Section 3.4.2.1. Edit Subpatterns ...**) and edit the **Start** of the subpattern, or

b. Drag the black arrow ⬐ delimiting the BEGINING of the subpattern to the new starting bit location

To modify the size of the subpattern, one can either:

a. Go to the **Edit Subpatterns** window (**Section 3.4.2.1. Edit Subpatterns ...**) and edit the **Length** of the subpattern, or

b. Drag the black arrow ⬎ delimiting the END of the subpattern to the new ending bit location

When the arrow delimiting the Beginning/End of the Subpattern is dragged (either for repositioning or for resizing) the status bar will display the exact position of the arrow being dragged. While this approach to editing the subpatterns is very intuitive, it is not always the most efficient for large numbers of subpatterns.

4.2. Drawing/Inputting Patterns and Waveforms

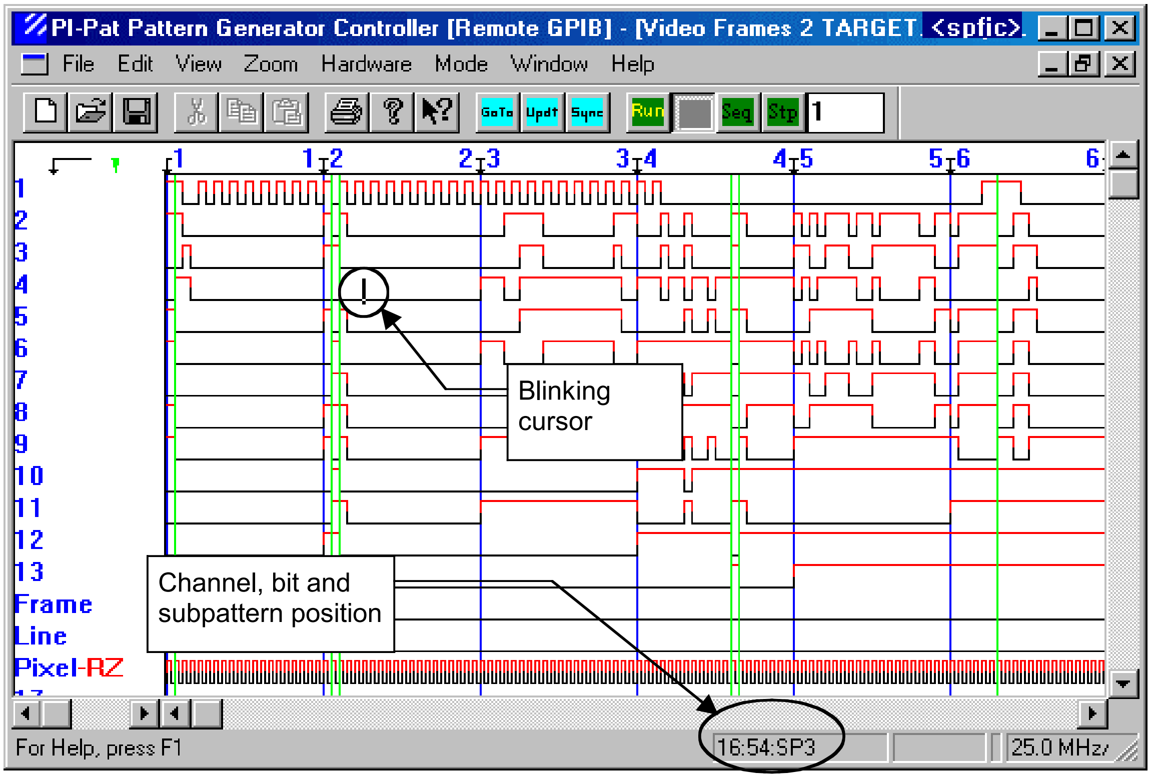
Once the limits of the subpatterns have been defined, they can be filled with waveforms. (You do not necessarily have to define subpatterns first. Waveforms can also be created first, and then subpatterns defined.)

To create a waveform on a particular channel, start by clicking the cursor on that channel, at the bit position where you want the waveform to start. If your pattern is large or if you have many channels, use the scroll arrows to make the desired bit position visible. You can also use the **Go To** button (see **Section 3.5.1. Go To**) to position the cursor at the desired position. Once at the desired position, the waveforms can be created:

a. From the keyboard by typing 1's and 0's or H's and L's (for High and Low levels). To move between and across channels, use the keyboard arrow keys (← ↑ → ↓).

b. By dragging with the left mouse button

When a waveform is drawn, the high levels are colored in **red**, while the low levels are **black**.



PI-Pat Pattern Generator Controller [Remote GPIB] - [Video Frames 2 TARGET <spfic>
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp 1
1 1 2 2 3 3 4 4 5 5 6 6
1
2
3
4
5
6
7
8
9
10
11
12
13
Frame
Line
Pixel-RZ
Blinking cursor
Channel, bit and subpattern position
For Help, press F1
16:54:SP3
25.0 MHz

Figure 29: Waveforms with channel, bit position and subpattern indicator

The position indicator will show the position of the last editing tool used, i.e., if the mouse is moved, the position of the tip of the mouse will be shown, and if the arrows are used to move the blinking cursor, the position of the cursor will be shown.

When editing a channel formatted in inverted mode, the channel name will be followed by the legend "**-Inv**" and the waveform will be changed to reflect the actual behavior of the channel when run. When typing or dragging in an inverted channel, the behavior will be reversed to remind the user that the channel is inverted.

When a channel is formatted in RZ mode, the waveform will be changed to reflect the actual behavior of the channel when run, and the channel name will be followed by "**-RZ**."

When a channel is formatted in Tri-State (HIZ) mode, the waveform display will be grayed-out.

The following example shows the displayed behavior when channels are formatted in HIZ, RZ and Inverted mode:

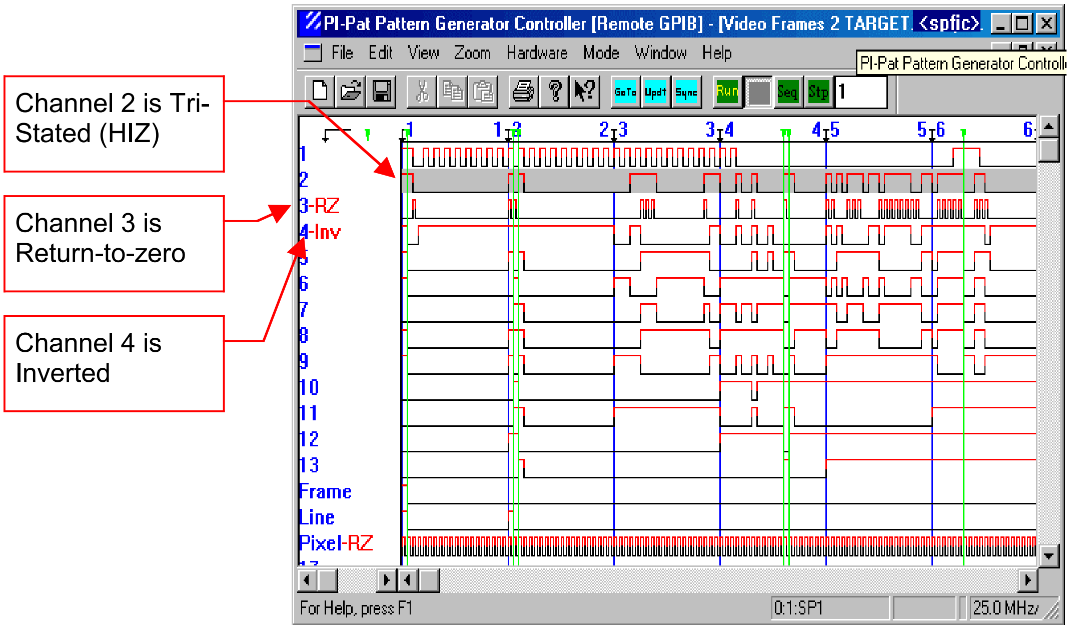

Channel 2 is Tri-Stated (HIZ)
Channel 3 is Return-to-zero
Channel 4 is Inverted
PI-Pat Pattern Generator Controller [Remote GPIB] - [Video Frames 2 TARGET. <spfic>.
File Edit View Zoom Hardware Mode Window Help
PI-Pat Pattern Generator Controll
GoTo Updt Sync Run Seq Stp 1
1 1 2 2 3 3 4 4 5 5 6 6
1
2
3-RZ
4-Inv
5
6
7
8
9
10
11
12
13
Frame
Line
Pixel-RZ
For Help, press F1
0:1:SP1
25.0 MHz

Figure 30: Waveform showing channel formats

4.3. Instructions

Once the subpatterns and the waveforms have been created, a program must be created by entering instructions. This is done in the **Edit Instructions** window (see **Section 3.4.2.2. Edit Instructions ...**).

Every program must start with a **Begin** instruction. The **Begin** instruction delimits programs, and without it a program will not be recognized. There is no END instruction; a program ends at the last non-empty line before the next Begin statement.

Each **Begin** statement has a number, and there can be multiple **Begin** statements.

4.4. Programming the hardware

After creating the patterns, instructions and formats, the hardware must be programmed. This is done in the **Hardware** dialog box. (see **Section 3.4.5. Hardware**).

4.5. Creating Arbitrary Waveforms

The PI-2005 can be used to generate digital representations of arbitrary waveforms. The arbitrary waveform function relies on binary combinations of bits, which requires that the channels be shown in their default **Hardware Order**. Turn this option on in the **View** menu.

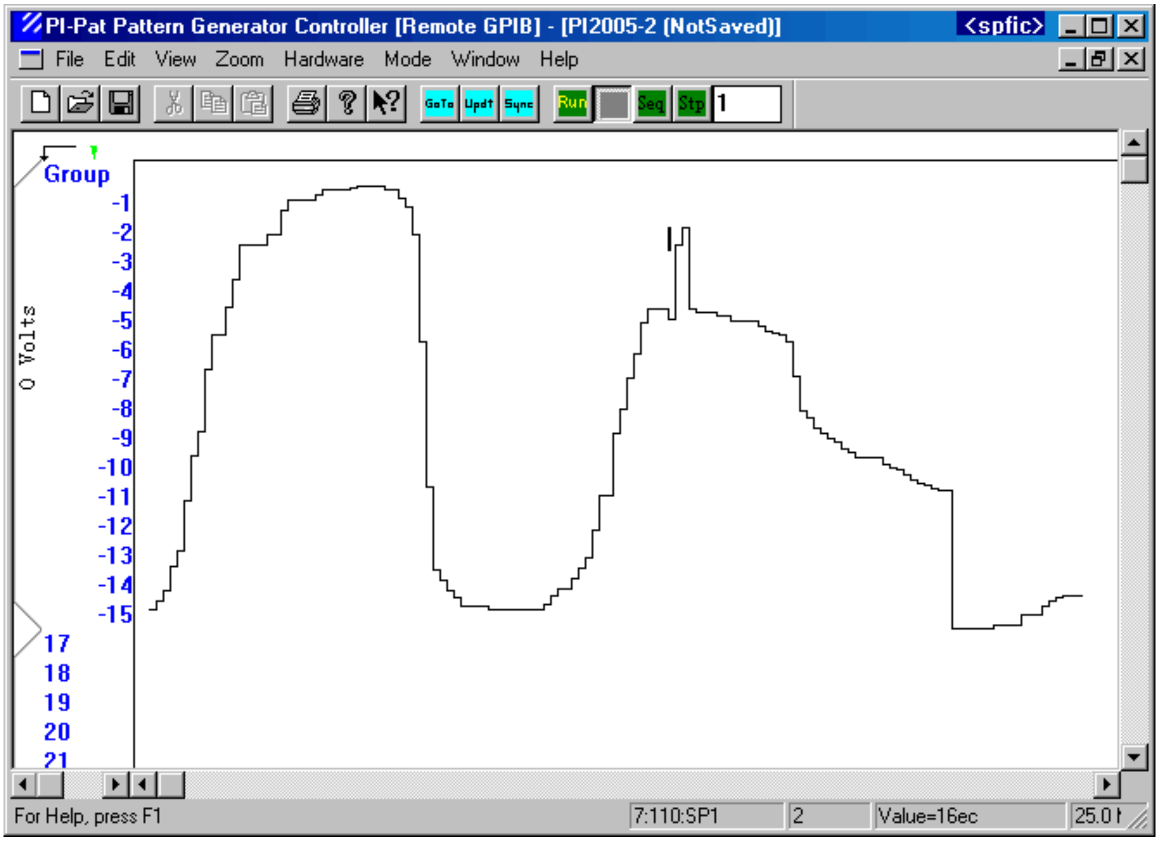

PI-Pat Pattern Generator Controller [Remote GPIB] - [PI2005-2 (NotSaved)]
<spfic>
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp 1
Group
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
17
18
19
20
21
0 Volts
For Help, press F1
7:110:SP1
2
Value=16ec
25.0

Figure 31: Creating Waveforms

To create an arbitrary waveform, follow these steps:

a. Select the channels that you want to use for the waveform generation, and group them (see **Section** 3.4.2.6. Create/Modify Group)

b. Choose to view the **Hardware Order** of the channels (see **Section 3.4.3. View**)

c. Position the mouse on one of the channels in the group, right click, and choose “Combine Waveforms” from the list

d. Hold the **Shift** key down and draw a waveform by dragging with the left mouse button

While showing the **Hardware Order** of a Group, right-clicking anywhere in the pattern data of that group will display the following menu:

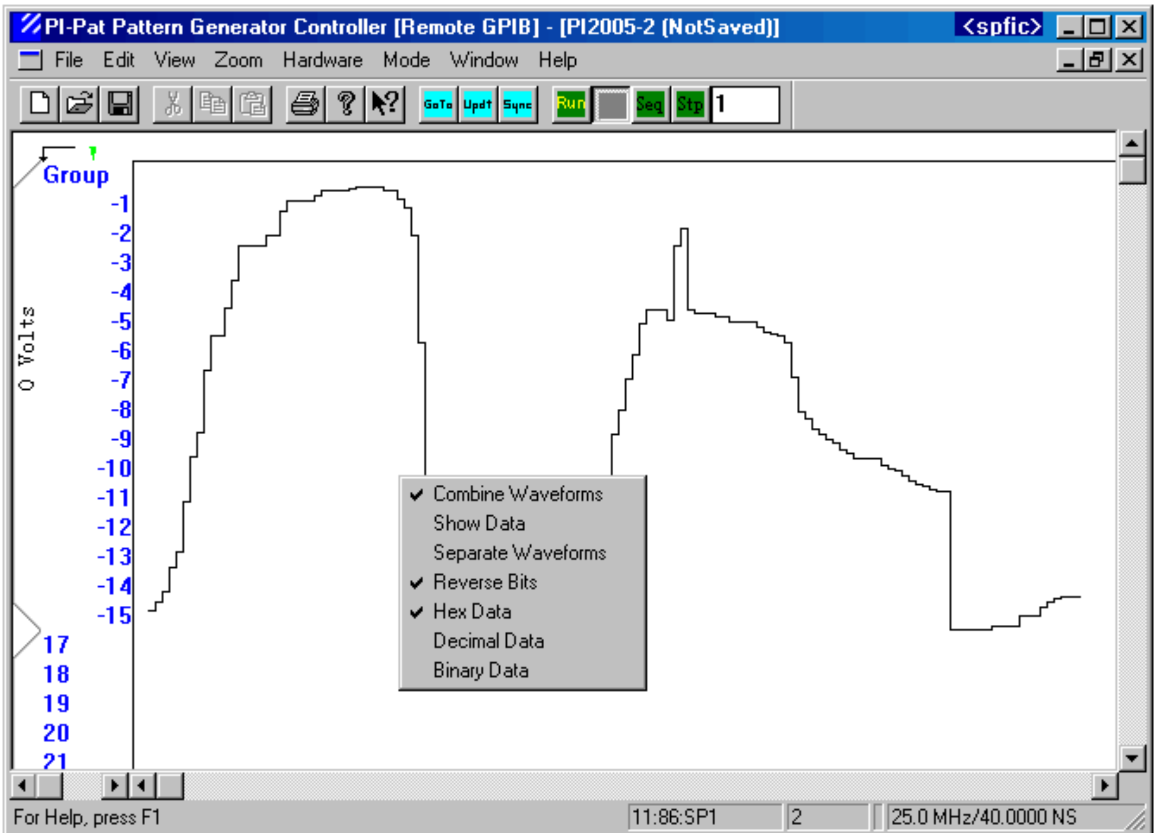


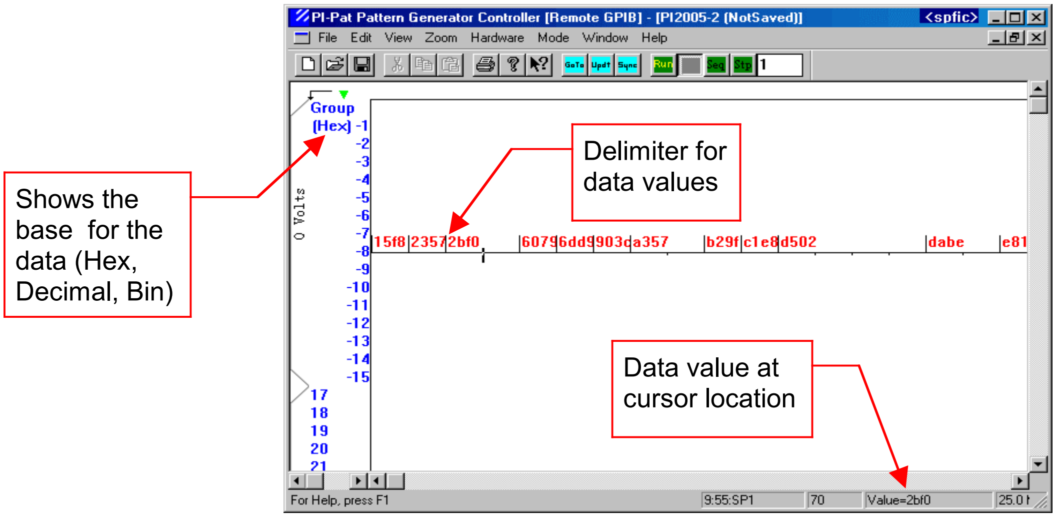
Figure 32: Waveform Display and Customization

4.5.0.1. Combine Waveforms

Will display all the channels as a single waveform (as shown in **Figure 31: Creating Waveforms**)

4.5.0.2. Show Data

Will display the actual numerical value of the Group data, based on a binary combination of the bits.


PI-Pat Pattern Generator Controller [Remote GPIB] - [PI2005-2 (NotSaved)]
<spfic>
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp 1
Group
(Hex) -1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
17
18
19
20
21
0 Volts
15f8 2357 2bf0 6079 6dd9 903d a357 b29f c1e8 d502 dabe e81
For Help, press F1
9:55:SP1
70
Value=2bf0
25.0

Shows the base for the data (Hex, Decimal, Bin)

Delimiter for data values

Data value at cursor location

Figure 33: Waveform showing Data in Hex

When **Show Data** is selected, the data value is shown for each clock period. If a data value repeats for more than one clock period, the data value is displayed once, and subsequent clocks with that value are indicated by hash marks. In **Figure 33**, the first three clock periods have the hex value “c2dd”.

You may need to use the **Zoom** menu to make the data values more readable.

4.5.0.3. Separate Waveforms

Will display each channel as an individual waveform. For example, the same waveform from **Figure 31: Creating Waveforms**, when **Separate Waveforms** is checked, will become:

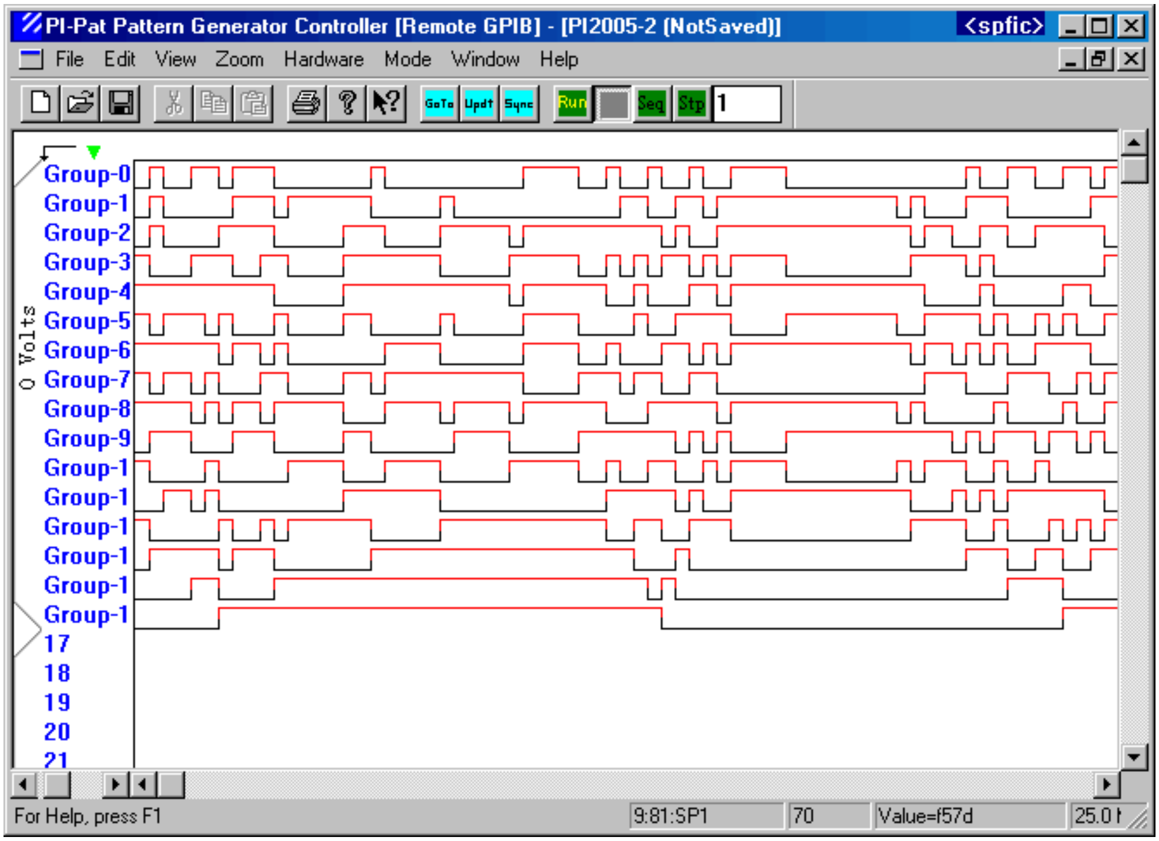


Figure 34: Waveform showing Separated Waveforms

4.5.0.4. Reverse Bits

Will change the order in which the bits of the group are calculated when combined by **Combine Waveforms** or **Show Data**. By default, the top channel is the most-significant bit (MSB), and bottom channel of the group is the least-significant bit (LSB). When **Reverse Bits** is checked, the bottom channel is the MSB and top channel is the LSB.

Note that when **Reverse Bits** is checked, the waveform (shown when **Combine Waveforms** is checked) will have a different shape too. For example, the same waveform from **Figure 31: Creating Waveforms**, when **Reverse Bits** is checked, will become:

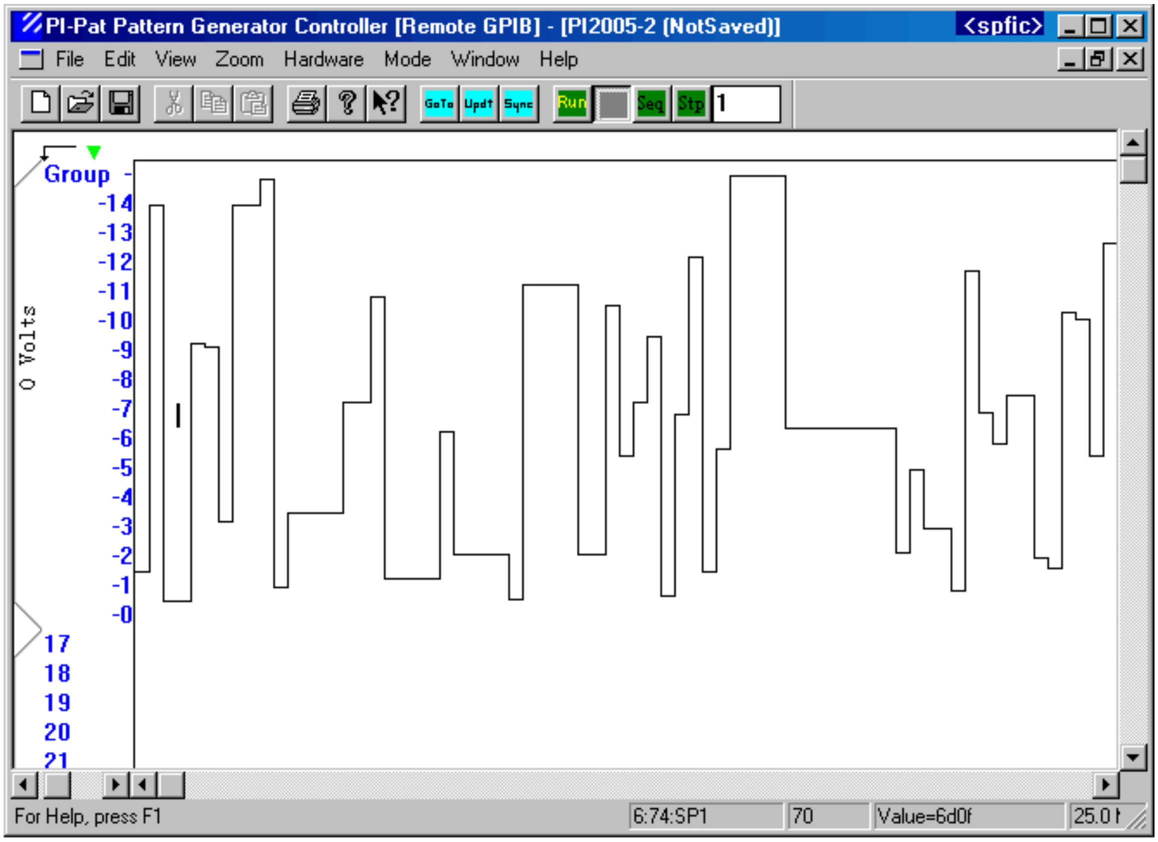

PI-Pat Pattern Generator Controller [Remote GPIB] - [PI2005-2 (NotSaved)]
<spfic>
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp 1
Group
-14
-13
-12
-11
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
-0
0 Volts
17
18
19
20
21
For Help, press F1
6:74:SP1
70
Value=6d0f
25.0 t

Figure 35: Waveform showing Reverse Bits

Note that the top channel is now the LSB (bit 0 of the Group), and the bottom channel is now the MSB (bit 15 of the Group).

The same result can be achieved by checking **Reverse Bits** in the Modify Group window (see **Section** 3.4.2.6. Create/Modify Group)

4.5.0.5. Hex, Decimal, Binary data

When the Group is displayed as **Show Data**, as shown in **Figure 33: Waveform showing Data in Hex** the value of the data can be displayed as a **Hex**, **Decimal**, or **Binary** value. The same result can be achieved by checking the appropriate check box in the Modify Group window (see **Section** 3.4.2.6. Create/Modify Group).

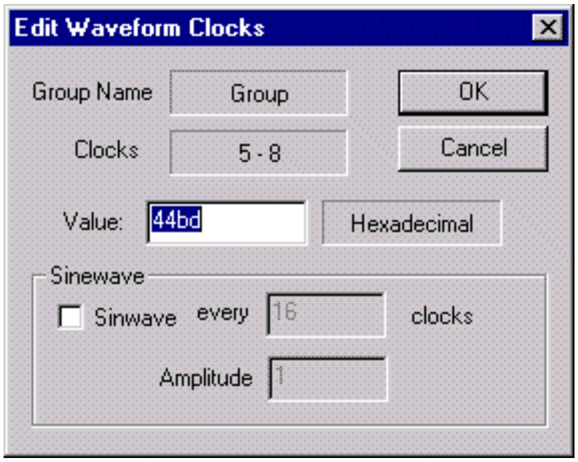
Note that, except for drawing a new waveform, none of the **Group** operations actually changes the pattern data or the output format of the channels.

4.6. Edit Waveform Clocks

When the Group is displayed as **Show Data**, the value of the data can be easily modified.

When the mouse is placed on the vertical lines that delimit the data values (**see Figure 33: Waveform showing Data in Hex**), a horizontal “scaling arrow” (↔) shows up. It can be used to extend the left data value to right, or the right data value to the left, by dragging with the left mouse button. Dragging to extend the data value will over-write existing data.

When a value is clicked above the horizontal line, the following dialog box appears:



Group Name: Group

Clocks: 5 - 8

Value: 44bd Hexadecimal

Sinewave

☐ Sinwave every 16 clocks

Amplitude 1

Figure 36: Edit Waveform Clocks

This dialog box allows the user to change the value of the bit positions shown in the “Clocks” window, by typing a value in the **Value** box.

If the user clicks below the horizontal line, a single bit position is selected. This allows the user to change the value of individual bits.

The **Sinewave** function is not yet implemented, and may be provided in a future update to PI-PAT.

5. TRI-STATE ON-THE-FLY NOTES

5.1. Overview

Tri-state “on-the-fly” (TSOTF) is a powerful new feature in the PI-2005. It allows a channel to be put in Tri-State or “Hi-Z” mode for the duration of a single program instruction. This permits the re-use of pattern data and subpattern definitions that would otherwise have to be duplicated.

For example, a user might want to test a device by driving 16 channels with a subpattern, and then selectively tri-stating the input to one channel at a time. Without TSOTF, this would require sixteen different program files with different output formats. With TSOTF, this becomes a simple program and sixteen Subpattern statements.

5.2. The No-Ringing Condition

Although TSOTF is a powerful feature, care must be taken when using it in programs, as tri-stating a non-terminated output can cause non-critically-damped ringing.

The outputs of the PI-2005 are back-matched with a 50 Ω termination. The back-matched termination can prevent reflections at the load even if the output cable is not terminated into 50 Ω, because a wave reflected by the end of the cable will be absorbed on the board. However, if an output is placed in a TRI-STATE (HI-Z, or High Impedance state), by definition this is a state where the impedance of the output is HIGH. The 50 Ω termination is not present, and thus no back-termination exists anymore.

In this situation, if the cable is not terminated properly, (for example it is terminated into 1 MΩ) the reflected wave will not be absorbed on the board and it will reflect back towards the end of the cable, which will reflect it back again, and so on, producing a non-critically damped ringing. The oscillation will eventually die out as the energy dissipates, but the ringing can produce undesirable effects in your test plan.

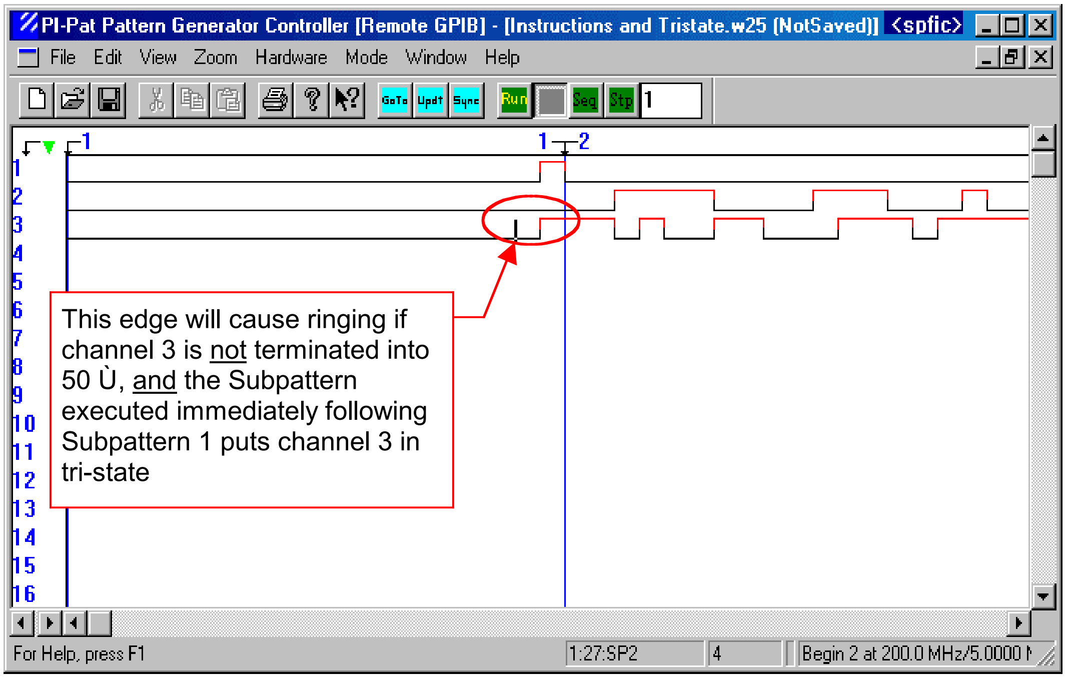
The simple rule to prevent ringing behavior is this: do not make any “transitions” (low → high or high → low) within T_{prop} before the channels are put into TRI-STATE. T_{prop} is the time it takes a wave to propagate from the output of the PI-2005 to the end of cable and back. Assuming a 1.5ns/foot propagation time through the output cable, and a 6 foot cable, this gives a

$$T_{prop} = 1.5 * 6 * 2 = 18 \text{ ns.}$$

Figure 37: The NO-RINGING condition

T_{prop} varies as a function of the output cable length, but will always be the double transit time of the cable. Channels are put in Tri-State mode at the beginning of the Subpattern command and remain in Tri-State through the Subpattern duration.

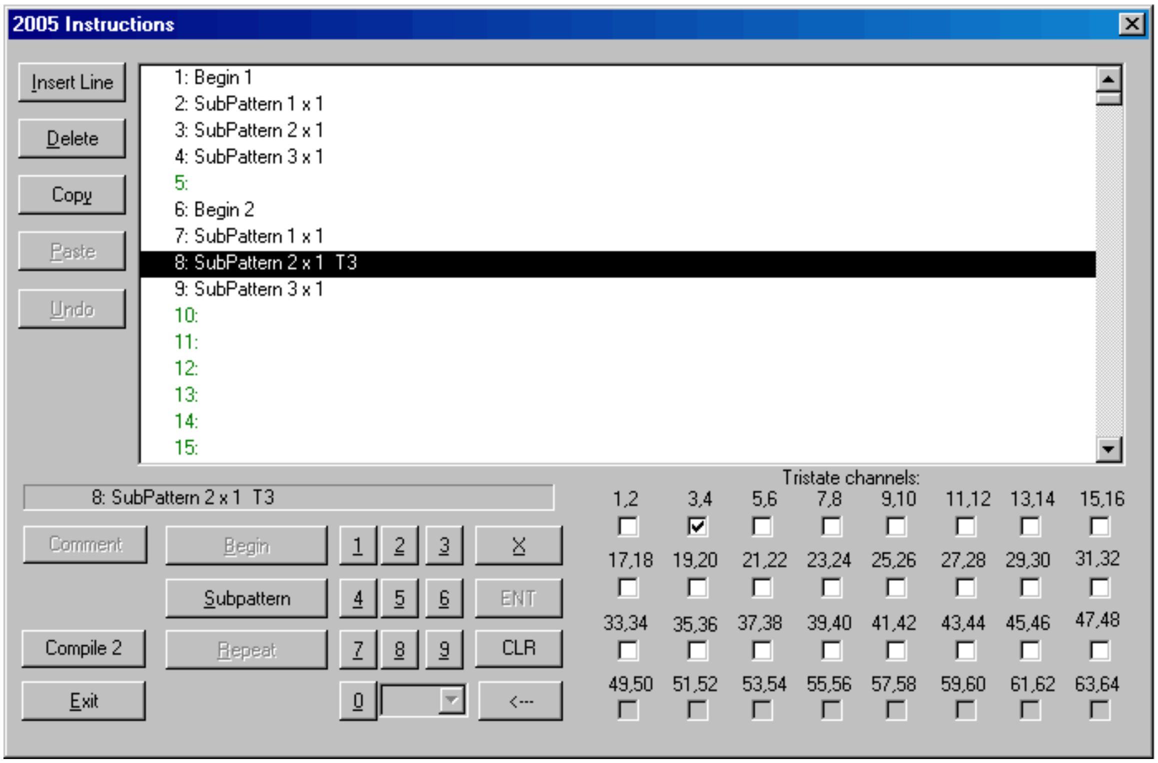
Here is an example to help illustrate this behavior.


PI-Pat Pattern Generator Controller [Remote GPIB] - [Instructions and Tristate.w25 (NotSaved)] <spfic>
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp 1
1
1
2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
For Help, press F1
1:27:SP2
4
Begin 2 at 200.0 MHz/5.0000 N

This edge will cause ringing if channel 3 is not terminated into 50 Ù, and the Subpattern executed immediately following Subpattern 1 puts channel 3 in tri-state

Figure 38: TSOTF Pattern Data and Subpattern Definitions

This program has 2 Begin statements:



1: Begin 1
2: SubPattern 1 x 1
3: SubPattern 2 x 1
4: SubPattern 3 x 1
5:
6: Begin 2
7: SubPattern 1 x 1
8: SubPattern 2 x 1 T3
9: SubPattern 3 x 1
10:
11:
12:
13:
14:
15:

8: SubPattern 2 x 1 T3

Tristate channels:

1,2	3,4	5,6	7,8	9,10	11,12	13,14	15,16
□	☑	□	□	□	□	□	□
17,18	19,20	21,22	23,24	25,26	27,28	29,30	31,32
□	□	□	□	□	□	□	□
33,34	35,36	37,38	39,40	41,42	43,44	45,46	47,48
□	□	□	□	□	□	□	□
49,50	51,52	53,54	55,56	57,58	59,60	61,62	63,64
□	□	□	□	□	□	□	□

Figure 39: TSOTF Program Instructions

In Begin 1 there is no TSOTF instruction. When run, Begin 1 will produce the following result on a scope:

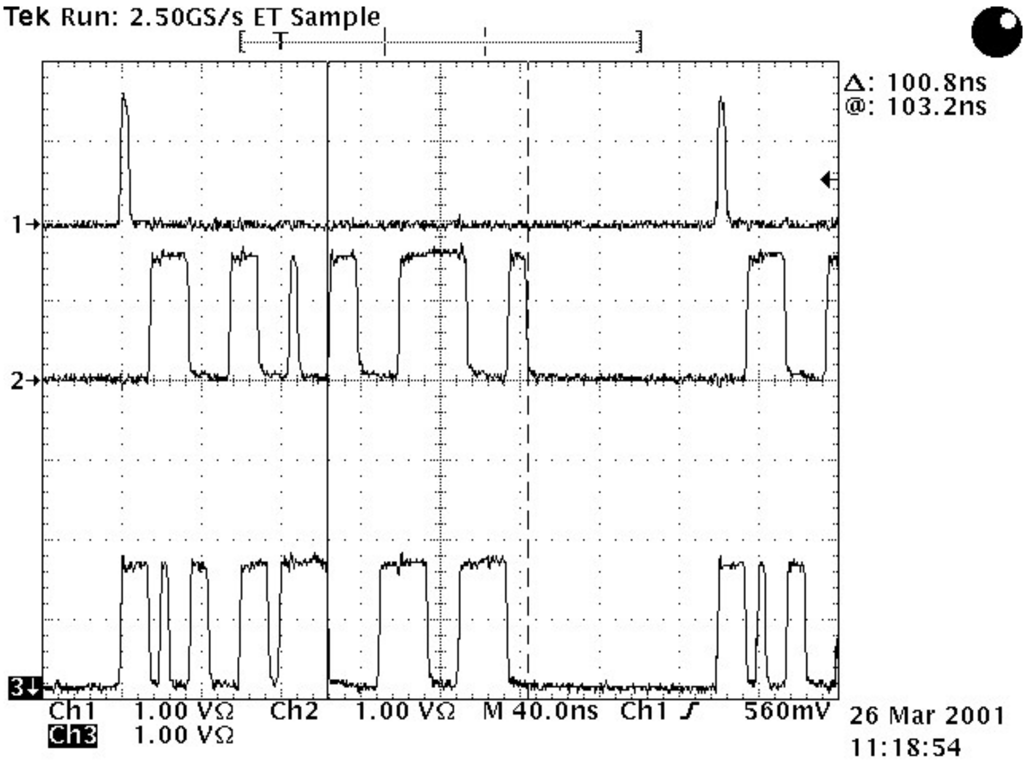

Tek Run: 2.50GS/s ET Sample
Δ: 100.8ns
@: 103.2ns
1
2
3
Ch1 1.00 VΩ Ch2 1.00 VΩ M 40.0ns Ch1 ⨍ 560mV
Ch3 1.00 VΩ
26 Mar 2001
11:18:54

Figure 40: TSOTF Output With No Tri-State

In Begin 2, channels 3 and 4 are placed in Tri-State during the execution of line 8, and are in NORMAL mode during lines 7 and 9. When run, Begin 2 will produce the following result on a scope, only if channel 3 is terminated **into 50Ω**:

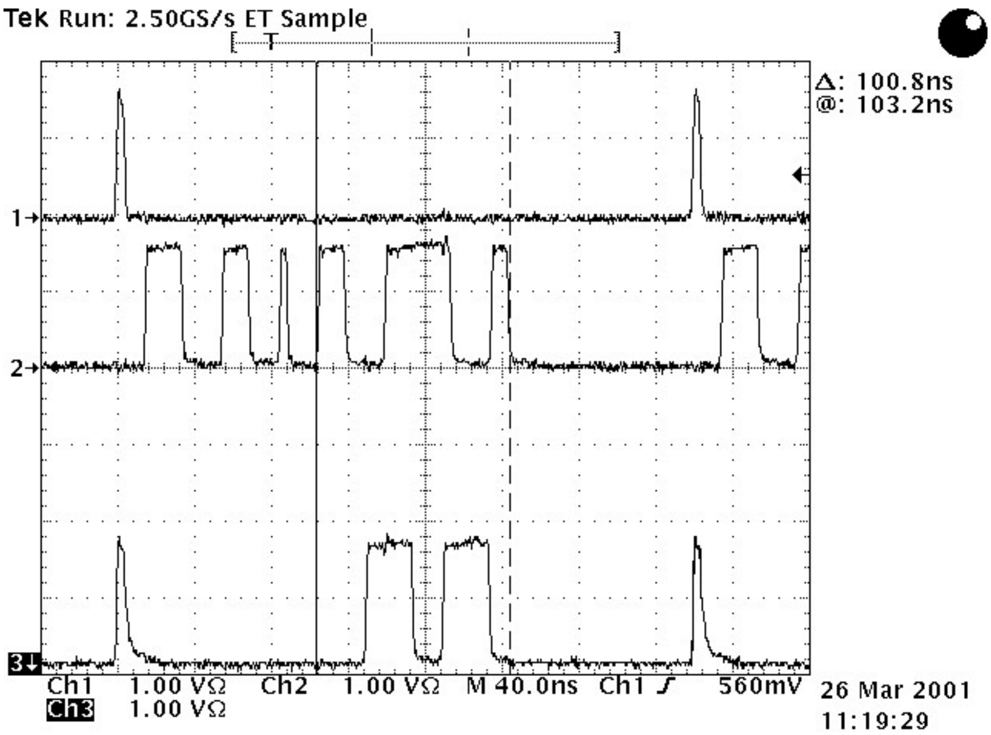
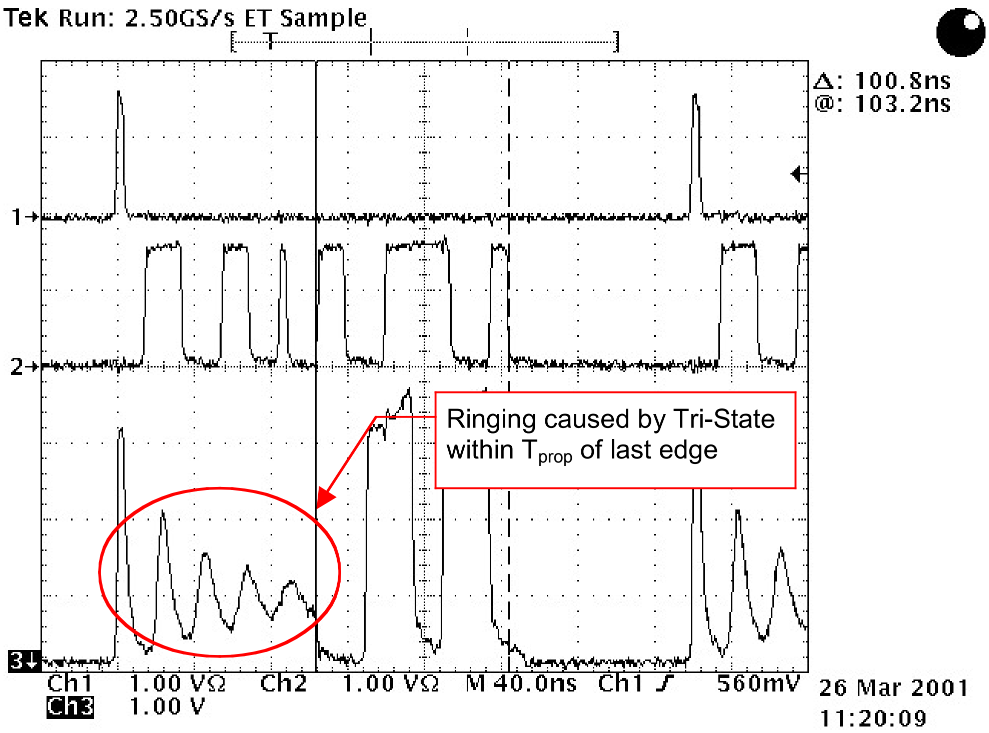


Figure 41: TSOTF No Ringing Into 50 Ω

The same Begin 2, when run with channel 3 terminated **into 1 MΩ**, will produce the following result:



Ringing caused by Tri-State within T_{prop} of last edge

Figure 42: TSOTF Ringing into 1 MΩ When Run at 200 MHz

Channel 3 is placed in Tri-State at the end of subpattern 1, and the rising edge on channel 3 occurs 5 ns before the Tri-State condition (when run at 200MHz). Because the wave does not have time to be absorbed, it reflects back and forth causing the ringing.

Note that the same program with the same patterns, if run at a lower frequency (20 MHz) works as expected, because the time interval between the rising edge and Tri-State condition is much longer:

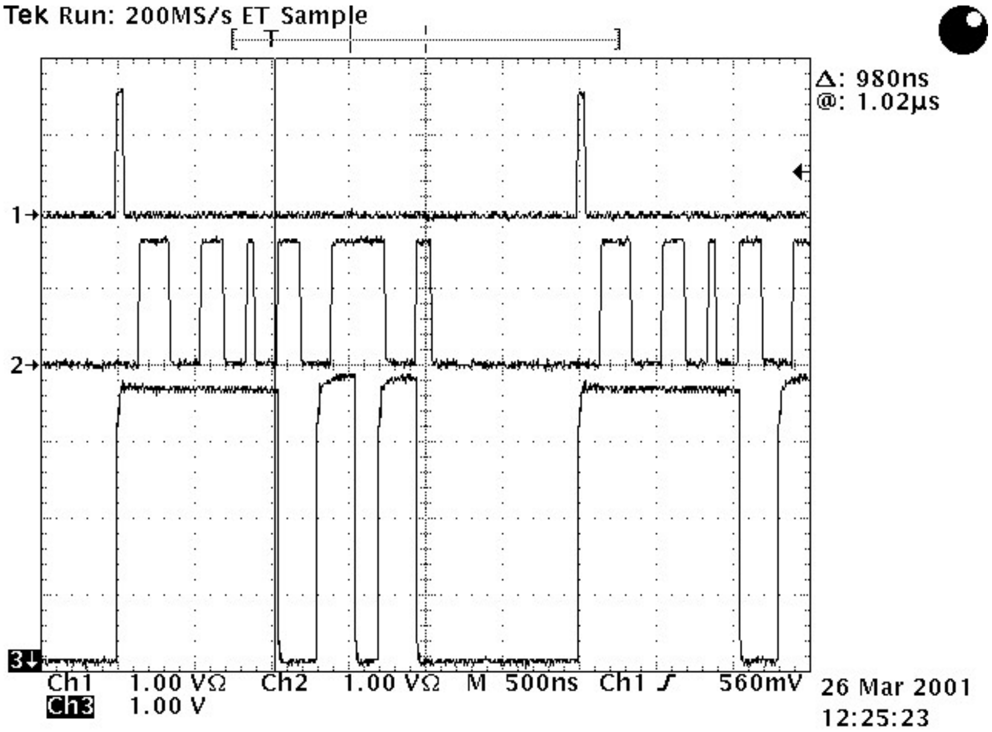
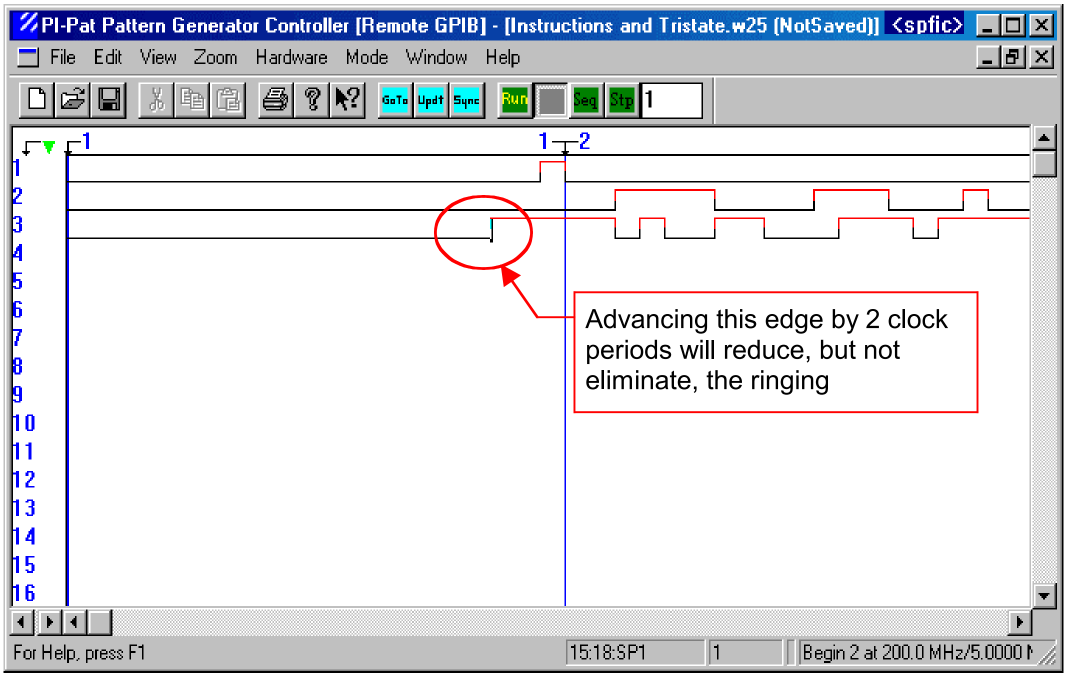


Figure 43: TSOTF No Ringing into 1 MΩ When Run at 20 MHz

If the clock period is longer then T_{prop} there is no danger of reflections. The edge reflections will always be absorbed before the Tri-State condition is invoked. For periods smaller then T_{prop}, the programmer must ensure that the NO-RINGING condition (see **Figure 37: The NO-RINGING condition**) is met.

Now here is the same pattern file from **Figure 38:** , but with the last edge in Subpattern 1 advanced by a two clock periods:



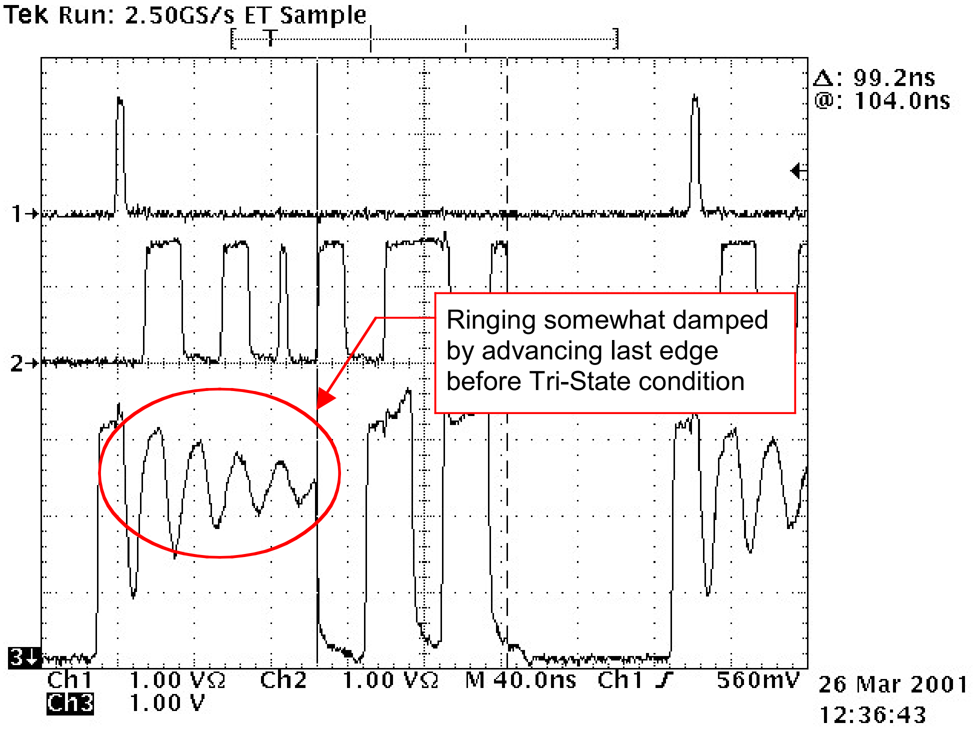
PI-Pat Pattern Generator Controller [Remote GPIB] - [Instructions and Tristate.w25 (NotSaved)] <spfic>
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp 1
1
1 2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
15:18:SP1
1
Begin 2 at 200.0 MHz/5.0000

For Help, press F1

Advancing this edge by 2 clock periods will reduce, but not eliminate, the ringing

Figure 44: TSOTF Pattern Data With Edge Moved Earlier by 2 CLKs

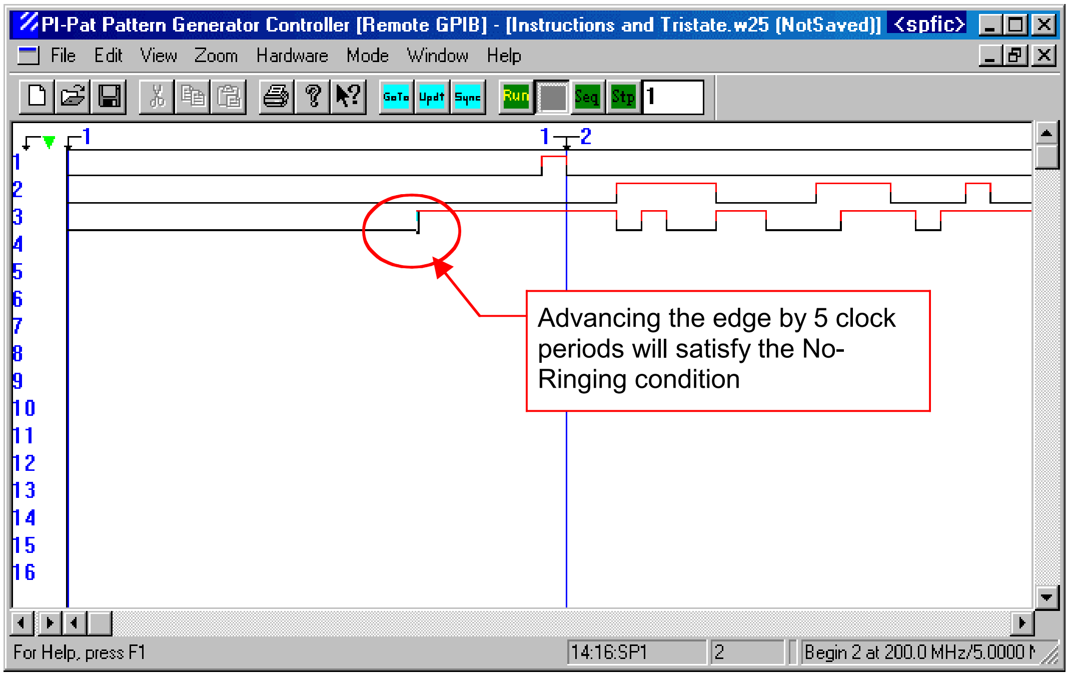
This will improve the behavior, because now the reflected wave has more time to dissipate. On the scope it will look like this:



Ringing somewhat damped by advancing last edge before Tri-State condition

Figure 45: TSOTF Ringing Damped When Run at 200 MHz

However, this will not completely eliminate the ringing. This requires moving the last edge further back, until it satisfies the NO-RINGING condition. Here is the same pattern with the position of the last edge advanced by 5 clocks:


PI-Pat Pattern Generator Controller [Remote GPIB] - [Instructions and Tristate.w25 (NotSaved)] <spfic>
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seg Stp 1
1
1 2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
For Help, press F1
14:16:SP1
2
Begin 2 at 200.0 MHz/5.0000 I

Advancing the edge by 5 clock periods will satisfy the No-Ringing condition

Figure 46: TSOTF Pattern Data With Edge Moved Earlier By 5 CLKs

This time there is no more ringing, just some benign reflections. There has been enough time for the open load reflection to be absorbed before the Tri-State condition was invoked.

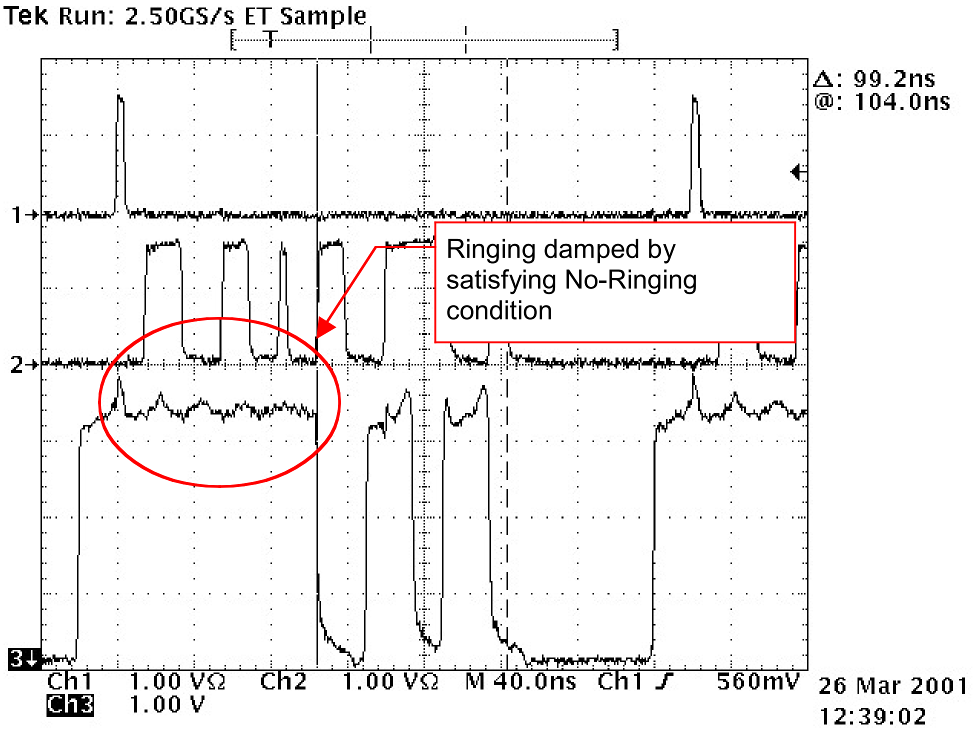

Tek Run: 2.50GS/s ET Sample
Δ: 99.2ns
@: 104.0ns
1
2
3
Ringing damped by satisfying No-Ringing condition
Ch1 1.00 VΩ Ch2 1.00 VΩ M 40.0ns Ch1 ∫ 560mV
Ch3 1.00 V
26 Mar 2001
12:39:02

Figure 47: TSOTF With Ringing Critically Damped When Run At 200MHz

6. PI-5800A COMPATIBILITIES

Although the PI-2005 was designed with the intent of providing maximum backward compatibility with the PI-5800A, there are a few command and parameter adjustments that may be required in order to allow the PI-2005 to run programs written for the PI-5800A.

6.1. Minimum subpattern length

The minimum allowable subpattern length for the PI-5800A and for PI-2120x cards is 12 bits. The minimum subpattern length for a PI-2110x cards is 20 bits.

Existing PI-5800A programs that violate the minimum subpattern length for PI-2110x cards can be used to program a PI-2005 if they are converted using PI-CONVERT, available from Pulse Instruments.

Conversion for minimum subpattern length is not required for PI-5800A programs used with PI-2120x cards.

PI-CONVERT avoids the minimum subpattern length violation by bit-multiplication and clock-multiplication. PI-CONVERT replaces each bit in the old program with two bits in the new program while doubling the programmed clock frequency.

6.1.1. Minimum subpattern length conversion, NRZ example

In this example, the "Unconverted" PI-5800 program runs at 50 MHz and has the following "illegal" 12-bit subpattern definition:

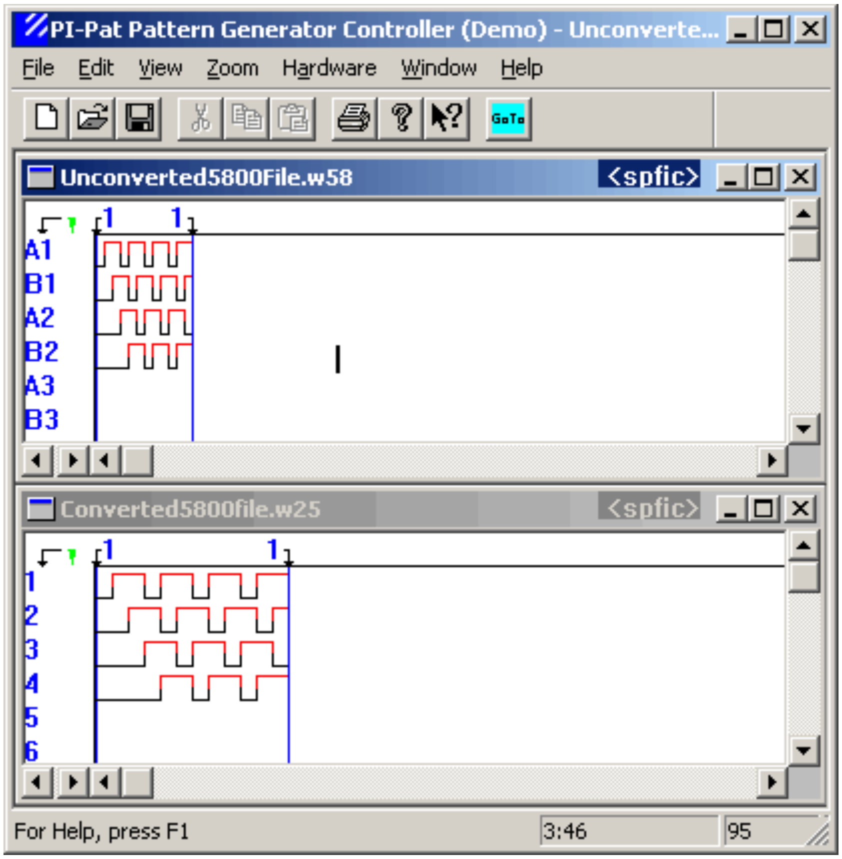


Figure 48: Converted PI-5800A file with "illegal" 12-bit subpattern definition in NRZ format

The "Converted" file has each bit replaced by two identical bits, and if run at 100 MHz, will have identical output to the original program run at 50 MHz on a PI-5800A. Note that this can also be verified by examining the original program at 2x zoom.

6.1.2. Minimum subpattern length conversion, RZ example

The "Unconverted" PI-5800 program shown below had channels set to RZ format:

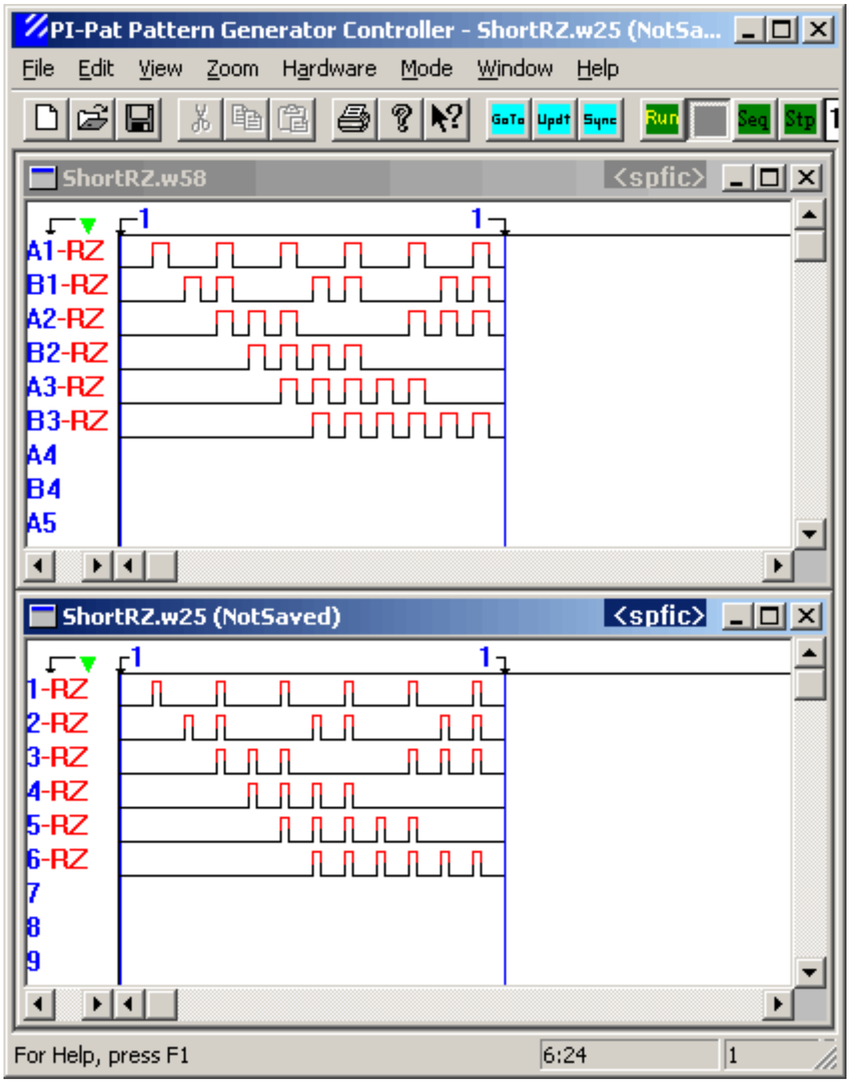

PI-Pat Pattern Generator Controller - ShortRZ.w25 (NotSa...
File Edit View Zoom Hardware Mode Window Help
GoTo Updt Sync Run Seq Stp
ShortRZ.w58 <spfic>
1 1
A1-RZ
B1-RZ
A2-RZ
B2-RZ
A3-RZ
B3-RZ
A4
B4
A5
ShortRZ.w25 (NotSaved) <spfic>
1 1
1-RZ
2-RZ
3-RZ
4-RZ
5-RZ
6-RZ
7
8
9
For Help, press F1
6:24
1

Figure 49: Converted PI-5800A file with "illegal" 12-bit subpattern definition in RZ format

Therefore the "Converted" program has a new zero bit after every original bit. If this program is run at 100 MHz, it will have identical output to the original program run at 50 MHz on a PI-5800A. Note that this can also be verified by examining the original program at 2x zoom.

This conversion can accomplished several ways, depending on the user's preference:

- Using PI-CONVERT, available from Pulse Instruments, for batch-converting large numbers of PI-PAT or GPIB files

- In PI-PAT, by loading a PI-5800 pattern file and performing a “**Save As**” command into PI-2005 format
- In PI-PAT, by manually editing the bit patterns and clock settings
- Via string-manipulation routines in custom software

6.2. Overlapping subpattern definitions

While the PI-5800A supported overlapping subpattern definitions, the PI-2005 does not. Each bit in a PI-2005 file must belong to one and only one subpattern.

Existing programs that violate the overlapping subpattern constraint can be used to program a PI-2005 if they are converted using PI-CONVERT, available from Pulse Instruments.

PI-CONVERT avoids violating the overlapping subpattern constraint by duplicating the pattern data of the second subpattern into a free area of memory, and then re-defining the second subpattern to point to the new data.

6.2.1. Overlapping subpattern conversion example

In the unconverted PI-5800A program shown below, bits 15-20 are common to SP 1 and SP 2:

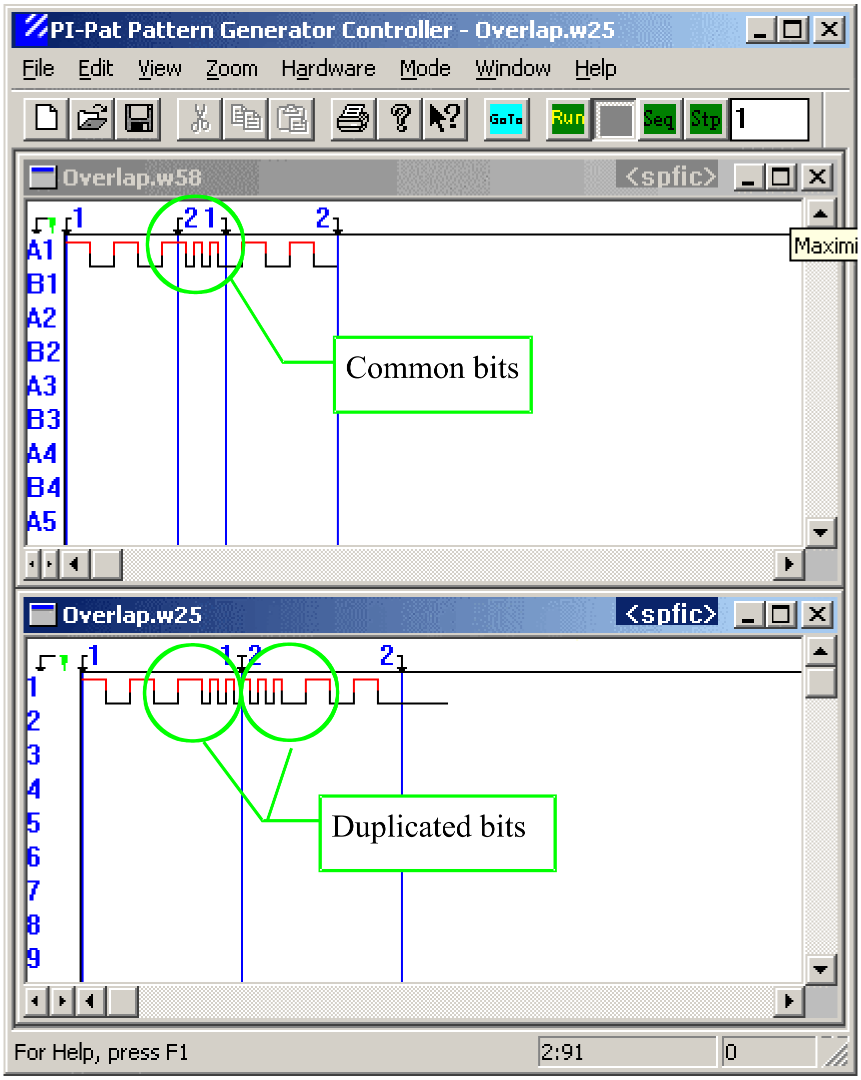


Figure 50: Converted PI-5800A file with overlapping subpatterns duplicated

After conversion, the pattern data for SP 2 has been duplicated and placed at bit positions 20-40, and the definition of SP 2 has been changed appropriately.

When run, this program will produce output identical to the original program.

This conversion can be accomplished several ways, depending on the user's preference:

- Using PI-CONVERT, available from Pulse Instruments, for batch-converting large numbers of PI-PAT or GPIB files
- In PI-PAT, by loading a PI-5800 pattern file and performing a "**Save As**" command in PI-2005 format
- In PI-PAT, by manually editing the bit patterns and subpattern definitions
- Via string-manipulation routines in custom software

6.2.2. Overlapping subpattern editing example

When converting overlapping subpatterns, users should take note that the pattern data that was formerly common to both subpattern definitions is now in two places. Any change to pattern data that is intended for both subpatterns must be changed in both places.

For example, a user wanting to edit bits 15-20 of the original PI-5800 file would only have to edit those bits in one place, and the changed output would be reflected in SP 1 and in SP 2. In the converted file, that change must be made in bits 15-20 and in bits 20-25.

6.3. Maximum programmable delay

The maximum programmable delay on the PI-5800A is 32 ns. The maximum programmable delay supported on the PI-2005 is 6.6 ns.

Because real-world latencies vary among systems, programmable delays should be set by each customer according to the particular requirements of the system being used. If PI-PAT converts a PI-5800 file (.w58) to a PI-2005 file (.w25), it will set all delays to 0.0 ns.

If PI-CONVERT processes a PI-PAT file that contains delays, PI-CONVERT will set all delays to 0.0 ns. If PI-CONVERT processes a GPIB command file containing DELAY commands, PI-CONVERT will comment-out those command lines and make notations in the conversion log.

6.4. GPIB Command Set Summary

The following sections describe differences in the GPIB command sets supported by the PI-5800 and PI-2005. For a complete description of GPIB control, including syntax and examples for each command, please see **Section** 7. PI-2005 Programmer's reference**.**

6.4.1. PI-5800A Commands Supported Without Modification

The PI-2005 implements the following commands with the same syntax and arguments as the PI-5800A (although some commands, such as DUMP INST, also have new, optional arguments).

COMMAND	SHORT DESCRIPTION
:hex data	Loads hexadecimal data.
(binary data) * x	Loads binary/algorithmic data.

COMMAND	SHORT DESCRIPTION
; remarks	Program remarks ignored by Pattern Generator.
BEGIN	Enters Begin instruction line.
CHANNEL	Sets output format of channel specified to INVERT, NONINV, RZ, NRZ, HIZ or NORM.
CLEAR	Clears memory and resets pattern generator. Optional argument clears SP, INST, CHANNEL, FORMAT or all settings.
COMPILE	Selects/compiles program to run.
CYCLE	Runs the currently compiled program one time
DEFINE SP	Defines subpattern.
DELAY	Sets optional programmable delay.
DELETE	Deletes a subpattern definition.
DUMP	Dumps data and settings from Pattern Generator back to controller.
GOTO	Points at channel or bit position to begin loading data.
ID	Returns identification information to controller.
LOAD CHANNEL	Selects channel to be loaded with data.
NUMCHAN	Returns number of channels installed in Pattern Generator.
PERIOD	Sets the internal clock period.
REPEAT	Enters Repeat on instruction line.
RUN	Runs pattern continuously.
SP	Enters Subpattern instruction line.
UPDATE	Sends data into pattern memory.

6.4.2. New PI-2005 GPIB Commands

The following new commands were not available in the PI-5800A.

COMMAND	FUNCTION
CLK	Selects internal clock or external clock with ECL/TTL thresholds.
DUMP CLK	Dumps clock source setting (internal or external)
DUMP DELAY	Dumps delay settings.
DUMP FORMATS	Dumps channel formats
DUMP PATTERN	Dumps pattern data
DUMP STASTP	Dumps External Start/Stop setting
DUMP SPDEF	Dumps subpattern definitions
GETDELAY	Returns presence/absence of optional delay feature.
POWERDOWN	Prepares the PI-2005 for powering down.

COMMAND	FUNCTION
SEQUENCE	Runs through the instruction set once a specified number of times
SETGPIBADDR	Assigns new GPIB address to PI-2005.
STASTP	Enables/Disables Start/Stop from rear panel.

6.4.3. GPIB Command Mismatches for PI-2005

The following GPIB commands are implemented differently in the PI-2005.

COMMAND	DIFFERENCE
RUNMODE	Requests the run mode condition. Responses are: “Stopped”, “Running”, “Paused”, “WaitSeq”, “WaitStep”, “WaitPause”, “WaitStop”, “WaitRun”.
STEP	Steps through the specified number of subpattern loops, whereas the 5800 would step through a specified number of clock periods
STOP	The PI-2005 resets to the first instruction line. This would replace the STOP followed by COMPILE x commands in the PI-5800.

6.4.4. GPIB Command Not Supported in PI-2005

The following GPIB commands are not supported in the PI-2005.

COMMAND	FUNCTION/WORKAROUND
BRANCH	No conditional branching in PI-2005.
C FLAG	No conditional branching in PI-2005.
CASCADE	64 Kb memory depth per channel replaces need for Cascade mode. Cascade mode and CASCADE command not supported.
DEFINE CHANNEL	Tri-State-On-The-Fly replaces mixed-mode in the PI-2005. DEFINE CHANNEL not supported.
DYNAMIC	Dynamic update not supported on the PI-2005.
HALT	Implemented as STASTP.
SRQ ON/OFF	None necessary. Service Request is always on in the PI-2005.

7. PI-2005 PROGRAMMER'S REFERENCE

7.1. Introduction

The previous sections of this PI-2005 Operators Manual are concerned with operation of the PI-2005 via Pulse Instruments PI-PAT software. The PI-2005 also can be controlled by 3rd-party software, running either on the embedded CPU card under Windows 7, or on a remote PC running a variety of operating systems.

In either case, the majority of the ASCII command set is identical, with the only difference being setup of the software interface for passing the commands.

If you have purchased the optional IEEE-488 Card, you can operate the PI-2005 with custom software via commands sent over the IEEE-488 bus from a remote PC. The remote PC can run any operating system supported by GPIB, and it can run any application capable of generating ASCII command strings and outputting them over the GPIB interface.

Custom applications running on the PI-2005's embedded computer must run under Windows 7, and must be capable of utilizing Windows Automation and MFC. Custom control of the PI-2005 on the embedded CPU is accomplished by passing ASCII command strings through an Automation interface to PI-PAT, which must be running at all times, although it may be hidden from view. Your application must contain a class to call this Automation method.

Any action or function that can be performed from PI-PAT can also be performed via the ASCII command set. You should already be familiar with using PI-PAT for any of the functions or features you wish before attempting to program the instrument.

The following sections of the manual describe the interface and commands used to operate the PI-2005 from your custom application.

The PI-2005 GPIB syntax follows the NI-488.2 standard. There are a few exceptions, where the older NI-488 standard is used, in order to ensure backward compatibility with the PI-5800.

We strongly recommend that even programmers familiar with the PI-5800 command set read this section. Although most 5800 commands are implemented identically in the PI-2005, there are some differences in syntax, as well as new commands that should be used for new applications in place of deprecated PI-5800 commands.

We also recommend that users do not use both ASCII commands and PI-PAT's GUI interface to control the instrument simultaneously. Local control may be "locked out" during receipt of commands to prevent interference.

7.2. Local Operation

7.2.1. Configuring the GPIB Interface

Use the **Mode** menu to change the operating mode to **Local Only** (See **Section** 3.4.6. Mode**).**

7.2.2. Automation Setup

To use automation, the user must add at least one class to his program. Code necessary to generate the class is included on the PI-2005 hard drive or on your installation CD in the \PULSE20\pi_pat.tlb type library.

The Automation method is dependent on MFC. There are several classes, one for each type: 5800, 6500, 2000 and 2005; but these are not necessary for using ASCII commands. Assuming m_PI_PatApp is an instance of the class, the user must make the following three calls to talk to PI_Pat:

```
if (m_PI_PatApp.m_lpDispatch == NULL)
  {
   m_PI_PatApp.CreateDispatch(_T("PI_Pat.Application.2"));
   m_PI_PatApp.m_lpDispatch->AddRef();
   m_PI_PatApp.LoadDocument(NULL,4);
  }
```

The NULL parameter passed to LoadDocument can be replaced by a pointer to a string containing the filename of either a valid PI-2005 file (*.w25) or of a text file containing valid ASCII commands. The "4" ensures that PI_Pat will open with a 2005 document. To send the STATUS command, for example, use:

```
CString strCmd(_T("STATUS\r\n"));
m_PI_PatApp.ExecuteCommand((LPCTSTR)strCmd);
```

To receive a response:

```
CString strResponse;
strResponse  = m_PI_PatApp.ReceiveResponse();
```

When finished talking to PI_Pat, the client application should call:

```
m_PI_PatApp.ReleaseDispatch();
```

to release the application.

To call these functions from your application, use the PI-PAT Type Library found in C:\Program Files\Pulse20\PI_Pat.TLB. Please replace “C:” with the appropriate drive letter if PI-PAT has been installed on a different hard drive.

7.3. Remote IEEE-488 Interface Setup

7.3.1. Configuring the GPIB Interface

Use the **Mode** menu to change the operating mode to **Local GPIB** (See **Section** 3.4.6. Mode**).** This allows the instrument to accept and parse incoming GPIB commands. PI-PAT must be running, and in **Local GPIB** mode, to accept incoming commands.

7.3.2. Command Strings and File Exchange

7.3.2.1. Input to the PI-2005/PI-PAT

Commands sent to the PI-2005/PI-PAT must obey the syntax specified in this document. When describing a command and its syntax, this document will use the following conventions:

All commands recognized by the PI-2005 will be printed in all capital letters.

Example: DEFINE SP 1 (1-20)↵

Placeholders for user-supplied values and parameters will be printed in lowercase, boldface, italicized characterized.

Example: PERIOD ***period units***↵

Optional arguments are enclosed in square brackets, e.g. DUMP PATTERN [***chnum***], but the square brackets should **not** be sent as a part of the command string. Strings sent to the PI-2005 need not be in capital letters, but they will be printed here in all capitals for clarity.

All command lines sent to the PI-2005 over GPIB must be terminated with a carriage return and a line feed (CHR$(13)+CHR$(10)). In hexadecimal these strings are "0D" and "0A." In this document, these two characters will be represented together by a "↵" at the end of each command line. Please note that, although it is displayed as a single character in this manual, both a carriage return and a line feed must be sent. The two examples above are correctly terminated.

Commands sent via Automation need not be terminated, as the end of the string delimits the end of the command. The Automation interface will accept the termination characters, though, to permit code reuse in your application.

7.3.2.2. Output From the PI-2005/PI-PAT

All information sent back from the PI-2005 is in the form of strings with the exception of the outputs from the **DUMP** and **STATUS** commands, which are sent as files. Programmers using low-level GPIB routines should use the Receive() function instead of the ibrd() function, especially when reading the output from a DUMP or STATUS command.

Strings are terminated with two characters:

```
0D 0A
```

which are a carriage return and line feed (CHR$(13)+CHR$(10)).

Files are terminated with the three character File Termination Sequence:

```
1A 0D 0A
```

where 1A is a hexadecimal End-of-File character (EOF), 0D is a hexadecimal carriage return and 0A is a hexadecimal line feed.

7.3.2.3. Checking Status of the PI-2005

A significant improvement has been made to the error checking and error handling for the PI-2005. In addition to an enhanced set of error codes, the PI-2005 now supports the use of a GPIB Receive() after every command line is sent.

For commands that return data (such as NUMCHAN and DUMP commands), a Receive() will begin returning the requested data, one line at a time.

For commands that do not return data (such as RUN or DEFINE SP), a Receive() will return the current error code (ERROR-00 if there is no error). This eliminates the need to send the STATUS command after every command has been sent.

Please note that consecutive Receive() commands are permitted with the PI-2005, but consecutive ibrd() commands will result in a time-out.

7.3.3. PI-2005 Message Mode for GPIB Communications

If your application communicates with PI-PAT via Automation, you may skip this section.

Multiple types of Pulse Instruments card can be installed in the same CompactPCI mainframe, and certain commands, such as **STATUS**, are recognized by more than one type of card. When using GPIB communications, there is only one GPIB interface and one GPIB address. Therefore, the PI-2005 uses “message modes” to determine which incoming GPIB commands are intended for which type of card.

A **BIASCLKMSGS** must be sent before a block of commands intended for DC Bias or Clock Driver cards. A **PATMSGS** must be sent before a block of commands intended for Pattern Generator cards. A **DACQMSGS** must be sent before a block of commands intended for Data Acquisition cards (PI-3105).

The **PATMSGS** command must be sent again if any commands have been subsequently sent to a different type of Pulse Instruments CompactPCI cards (such as Pattern Generator cards, for example).

Please see the following command lines for an example of usage:

```
BIASCLKMSGS ; THESE COMMANDS ARE FOR BIAS/CLK CARDS↵
ALL LOLIMB -3.00↵
ALL UPLIMB 5.00↵
MF 0 SLOT 3 CH 2 VOLT —1.80↵
MF 0 SLOT 3 CH 1 VOLT 2.3↵
CONNECT↵
ONLINE↵

PATMSGS ; THESE ARE FOR PATTERN GENERATOR CARDS
COMPILE 1↵
SEQUENCE ↵
SEQUENCE ↵
SEQUENCE ↵
SEQUENCE ↵

BIASCLKMSGS ; THERE ARE FOR BIAS/CLK CARDS↵
OFFLINE↵
```

7.3.4. Channel Numbering Convention

To provide maximum compatibility with programs written for the PI-5800, the PI-2005 accepts commands with A1B1-style channel numbering as well as standard 1-2-3-style numbering. In A1B1-style numbering, Channel 1 is designated A1, channel 2 is designated B1, channel 3 is A2, channel 4 is B2, and so on. The tables below maps PI-5800 channel numbering to PI-2005 channel numbering:

PI-5800	PI-PAT	PI-2005		
Cable	Ch#	Card	J#	Ch#
A1	1	1	1	1
B1	2	1	1	2
A2	3	1	1	3
B2	4	1	1	4
A3	5	1	2	1
B3	6	1	2	2
A4	7	1	2	3
B4	8	1	2	4
A5	9	1	3	1
B5	10	1	3	2
A6	11	1	3	3
B6	12	1	3	4
A7	13	1	4	1
B7	14	1	4	2
A8	15	1	4	3
B8	16	1	4	4

PI-5800	PI-PAT	PI-2005		
Cable	Ch#	Card	J#	Ch#
A9	17	2	1	1
B9	18	2	1	2
A10	19	2	1	3
B10	20	2	1	4
A11	21	2	2	1
B11	22	2	2	2
A12	23	2	2	3
B12	24	2	2	4
A13	25	2	3	1
B13	26	2	3	2
A14	27	2	3	3
B14	28	2	3	4
A15	29	2	4	1
B15	30	2	4	2
A16	31	2	4	3
B16	32	2	4	4

New applications should use 1-2-3 style numbering.

7.4. Programming Methodology

7.4.1. General overview

There are three general types of ASCII commands supported by the PI-2005.

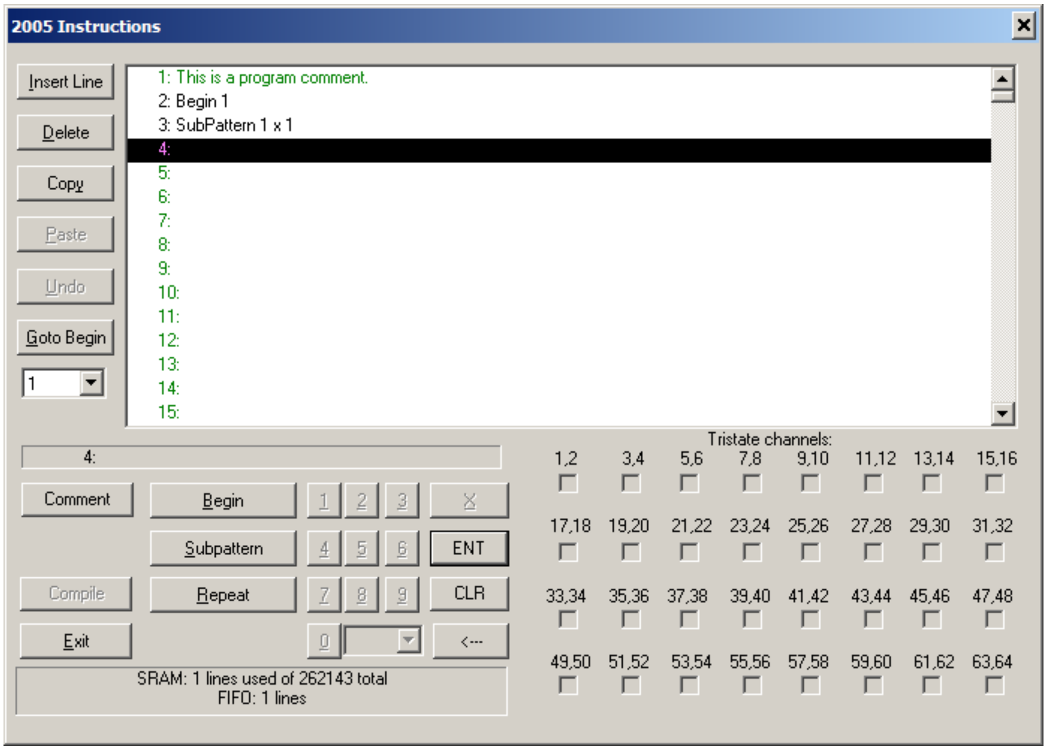
7.4.1.1. Programming Commands

The first general type deals with programming the Pattern Generator with pattern data, subpattern definitions, program instructions and hardware settings. These are commands such as DEFINE SP, BEGIN, and LOAD CHANNEL.

The command syntax for programming commands is the same as the text Import/Export syntax. Therefore, users can use the results of the PI-PAT **File:Export** as a starting point or syntax guide for creating their own GPIB command files.

Users can also test the syntax of their GPIB command files by using **File:Import** in PI-PAT. Note that, although programming commands will be read by the **File:Import** function, Control and Query functions (below) will generate errors when Imported by PI-PAT. These errors do not necessarily indicate any problem with the syntax in the file, but only programming commands have a valid meaning in an Imported file.

Instruction line comments can also be imported and exported from PI-PAT (*.w25) files, although at present time they cannot yet be received over GPIB. Program comments are exported with a bang (!) character. Consider the following program in PI-PAT:



```
1: This is a program comment.
2: Begin 1
3: SubPattern 1 x 1
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
```

4:

SRAM: 1 lines used of 262143 total
FIFO: 1 lines

Figure 51: PI-PAT Program with Comments

This would be exported as follows:

```
1 !This is a program comment.
2 BEGIN 1
3 SP 1 * 1
```

Importing the above text would result in the program shown in **Figure 51: PI-PAT Program with Comments**.

Note that, prior to version 3.465 (Build 355), program comments are ***not*** currently received properly over GPIB, and will generate syntax errors.

7.4.1.2. Pattern Generator Control

The second general command type deals with controlling the Pattern Generator. Commands such as RUN, STOP, SEQUENCE, UPDATE and COMPILE duplicate the functions of the toolbar buttons and dialog box buttons in PI-PAT.

7.4.1.3. Pattern Generator querying

The third general command type returns information about the Pattern Generator across the GPIB. Queries can return information about the configuration of the PI-2005 or return programming data entered by commands of the first type noted above.

7.4.1.3.1. Configuration and Informational queries

Commands such as ID, NUMCHAN and GETDELAY return information about the instrument itself—e.g. the type of instrument and the type of cards installed—to the operator or system controller.

Commands such as RUNMODE and STATUS return the “state” of the instrument, and are useful for interactive control and debugging of an application.

7.4.1.3.2. Instrument settings queries

The third set of query commands returns the programming data that would have been sent with the programming commands described in **Section 7.4.1.1. Programming** With a few exceptions, data is returned from queries in the same format that the complementary command uses to send data. For example, the command for querying the Clock selection:

```
DUMP CLK
```

will return a string that is a valid clock selection command, e.g.:

```
CLKEXTTTL
```

Returned data from DUMP commands can generally be sent back to the PI-2005 as a ASCII command string or else Imported into PI-PAT. GPIB programmers using low-level routines should use the Receive( ) command instead of the ibrd( ) command, especially when reading the output from a DUMP or STATUS command.

With the DUMP commands, the entire state of the instrument can be passed across the to your application, either to be saved and restored to the instrument at a later time, or else to be used to develop of a custom, interactive application for the PI-2005.

Several DUMP commands supported by the PI-5800 are deprecated for the PI-2005. They are supported for backward compatibility, but all new applications should use the newer commands:

PI-5800 Deprecated Command	PI-5800 Sample Returned String	New PI-2005 Command	PI-2005 Sample Returned String
DUMP FORMAT	Block format, e.g. :2000100010C0100	DUMP FORMATS	CHANNEL 1 INVERT CHANNEL 1 HIZ CHANNEL 1 NRZ
DUMP CHANNEL	Block format, e.g. :200020002F02B0	DUMP PATTERN	LOAD CHANNEL A1 :200020002F02B0
DUMP SP	0001 (0001 – 0020)	DUMP SPDEF	DEFINE SP 0001 (0001 – 0020)

The advantage to using these new commands is that the returned strings are in exactly the same format as the corresponding programming commands. If the outputs from the new DUMP commands are saved to a file, the resulting file can be sent back to the instrument, without modification.

There are also several new or improved DUMP commands for returning other pattern generator settings

New PI-2005 Command	PI-2005 Sample Returned String	Feature or Improvement
DUMP CLK	CLK EXTTTL	New for PI-2005
DUMP DELAY	DELAY 1 - 500	New for PI-2005
DUMP INST	5 SP 3 * 1023	Optional line number argument can return a subset of instructions
DUMP PERIOD	PERIOD 10.11	Newly implemented for PI-2005

DUMP STASTP	STASTP 0	New for PI-2005

7.4.2. PI-PAT behavior over GPIB

Users may monitor the GPIB commands sent by PI-PAT in Remote Mode to a connected PI-2005 running PI-PAT in Local Mode using a GPIB monitoring program such as National Instruments' NI-Spy or NI I/O Trace.

PI-PAT may exhibit some non-obvious GPIB behavior due to code re-use among the modules used for programming the Pulse Instruments family of pattern generators. In particular, PI-PAT may send a large number of STATUS commands in between other commands. This behavior is normal, and is intended to clear syntax errors and wait loops in the PI-5800. These STATUS commands are not necessary when sending GPIB commands to the PI-2005, and need not be duplicated in your programs.

7.5. Sample Application/Testing Utility (GPIBCom)

Included with your PI-2005 is an unsupported utility, **GPIBCom.exe** for sending and receiving strings from Pulse Instruments equipment or other GPIB-compatible devices. **GPIBCom** can be run Locally, on the CompactPCI CPU board, or Remotely, on a Windows PC with a National Instruments GPIB interface card. When running Locally, GPIBCom communicates either with Pulse Instruments DLLs (**dacq.dll** and **pipci.dll**) via their exposed functions or with Pulse Instruments applications (**PI-PAT**) via Automation. When running Remotely, **GPIBCom** communicates with any instrument that is capable of communicating over GPIB. In each case, this manual will refer to the instrument, DLL or application as the "target."

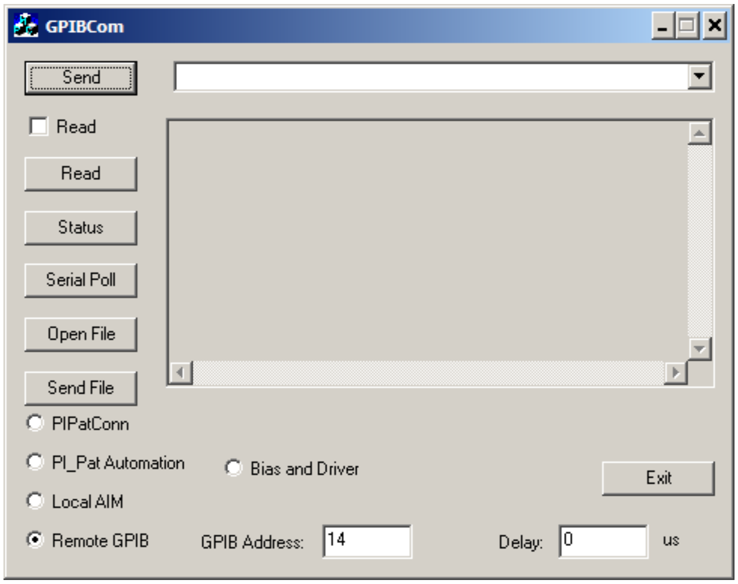
Because Pulse Instruments DLLs and applications use the same command set whether run Locally or Remotely, via the Pulse Instruments GUI or a custom application, **GPIBCom** can be used to learn, test, and troubleshoot the hardware and your custom application.

Complete source code for GPIBCom.exe is provided as a MSVC++ project in the C:\Program Files\Pulse20\CodeSamples directory. Code samples from the project may be copied and used freely in your application.

7.5.1. Local Setup

GPIBCom is located in the C:\Program Files\PULSE20 folder. If you wish to run **GPIBCom** in Local mode, then the executable must remain in this folder, as it links into several Pulse Instruments DLLs that also are in this folder. A shortcut to this executable may be placed anywhere on your hard drive or desktop folder.

Launch **GPIBCom** by double-clicking on its icon or shortcut:



☐ Read

○ PIPatConn
○ PI_Pat Automation
○ Bias and Driver
○ Local AIM
● Remote GPIB

GPIB Address: 14
Delay: 0 us

Figure 52: GPIBCom

By default, **GPIBCom** launches in Remote GPIB mode, and is set up to communicate with a GPIB-compatible device. To communicate with Pulse Instruments software on the CompactPCI CPU, click one of the following radio buttons:

- **PI_Pat Automation**—to communicate with **PI-PAT**, which controls the Pattern Generator hardware. If PI-PAT is not running when the first command is sent, PI-PAT will be launched.
- **Pipci.dll**—to communicate with **pipci.dll**, which controls the DC Bias and Clock Driver cards. If PI-PAT is running on the Local CPU, it **must** be in Local Only mode before **GPIBCom** is launched.
- **Dacq.dll**—to communicate with **dacq.dll**, which controls all data acquisition hardware. If PI-PAT is running on the Local CPU, it **must** be in Local Only mode before **GPIBCom** is launched.

Multiple instances of **GPIBCom** may be launched, in different modes, to communicate with different targets. Note that each DLL or application (in the case of PI-PAT) is quit and re-launched when the radio buttons are clicked in a particular instance of **GPIBCom**. Therefore, if you wish to communicate with **PI-PAT and** with **dacq.dll**, for example, it is highly recommended that you launch two instances of **GPIBCom**, rather than switching one instance of **GPIBCom** back and forth between the two modes.

Note that **GPIBCom** will link to dacq.dll or pipci.dll when used, and therefore will prevent PI-Controller from controlling the relevant hardware. The instance(s) of **GPIBCom** using dacq.dll and pipci.dll should be quit before launching PI-Controller.

7.5.2. Remote Setup

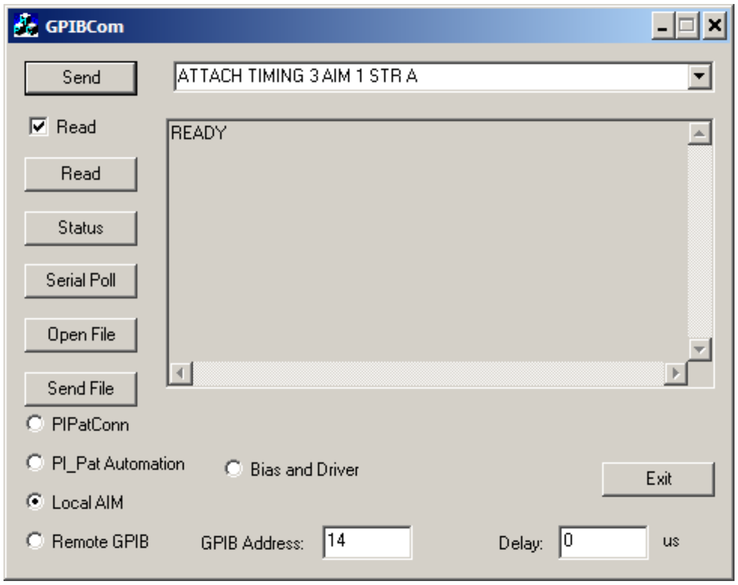
GPIBCom may be copied from your PI-2005 to any Windows PC equipped with a National Instruments GPIB interface card. No other drivers or DLLs are required for remote control. In Remote mode, please ensure that the **Remote GPIB** radio button is selected, and that the **GPIB Address** box corresponds to the GPIB address of the device you are controlling.

Multiple instances of **GPIBCom** may be launched to communicate with multiple instruments at different GPIB addresses. To communicate with multiple instrument types within a PI-11000 Instrument Mainframe, use the GPIB Message Mode commands (**PATMSGS**, **BIASCLKMSGS**, and **DACQMSGS**) to switch among different messaging modes.

When communicating with a PI-11000 Instrument Mainframe or PI-2005 Pattern Generator, ensure that **PI-PAT** is running on the Local CPU in Local GPIB mode.

7.5.3. General Operation

GPIBCom can send either single lines of text or entire text files to a target. To send a single string, set up communications per the instructions above, and then type a command in the text entry box next to the **Send** button:



ATTACH TIMING 3 AIM 1 STR A

☑ Read

```
READY
```

○ PIPatConn
○ PI_Pat Automation
○ Bias and Driver
● Local AIM
○ Remote GPIB

GPIB Address: 14

Delay: 0 us

Figure 53: GPIBCom communicating with dacq.dll

Click the **Send** button to send the command. If the **Read** checkbox is checked, **GPIBCom** will immediately query the target for a response. In the case of Pulse Instruments instruments and DLLs, the response will be either an error message, a no-error message, or the response to the query.

Clicking the **Read** button causes **GPIBCom** to query the target for a response.

Clicking the **Status** button is equivalent to sending the string "**STATUS**", followed by a **Read**.

Clicking **Serial Poll** executes a standard GPIB Serial Poll and displays the result.

GPIBCom can also send the contents of text file, either one line at a time or all at once. To send a file, click the **Open File** button. A standard Windows File Open dialog box will appear. Once you have selected a text file, **GPIBCom** will prompt you for a filename to save query responses to. This must be an existing file, and **GPIBCom** will append strings to the end of it. You can choose not to log strings to a file by pressing **Esc** or clicking **Cancel**.

Once the file has opened, the first line of text from the file will appear in the text entry box. Click **Send** to send a single line of text and read the next line into **GPIBCom**. Click **Send File** to send the entire contents of the file to the target. Please note that, when **Send File** is clicked, the file contents are sent via a series of GPIB Send() commands, not ibwrtfl(). Once a file has been sent via **Send File**, the line counter resets, and clicking Send File will send the file again, from the beginning of the file.

To cancel sending of a file, click the **Close File** button.

If you wish to insert a delay between strings that are sent to the target via **Send File**, type a value in the **Delay** box.

8. PI-2005 COMMAND REFERENCE

:HEX DATA

Syntax:

:*datasize startbyte sync hexdata*↵

datasize is the length of the ***hexdata*** in bytes. ***datasize*** can be 00-20 hex.

startbyte is the byte-position at which to begin loading data. ***startbyte*** can be from 0000-FFFE hex, but ***startbyte*** plus ***datasize*** can not exceed FFFF.

sync must always be the hex value 00.

hexdata is the actual data to load, in hexadecimal.

The colon command ":" begins loading hexadecimal data at the channel selected by the last LOAD CHANNEL or GOTO CHANNEL command.

The spacing in the syntax example is for readability only. There should be no spaces in the actual command string.

Example:

```
LOAD CHANNEL 3↵
:10000100FEDCBA9876543210FEDC
BA9876543210↵
:10001100 0123456789ABCDEF0123
456789ABCDEF↵
```

will load Channel 3 with 32 bytes of data, starting at bit 0008 (hex). Note that this example will load the same data into Channel 3 that the "(binary)" example below loads into Channel 4.

Note that changes to the pattern data will not be reflected in the output until an UPDATE is performed.

The hex load format uses whole bytes only. To load data into bit positions that do not coincide with whole bytes, you can use the binary load format below.

See also: **(binary data)**
CLEAR CHANNEL
LOAD CHANNEL
UPDATE

(binary data)

Syntax:

***(binary) * multiple*↵**

binary is the data to be loaded at the desired bit position. ***binary*** should be a string of "1" and "0" characters up to 255 characters in length.

**** multiple*** number of consecutive times to load the binary data. ***multiple*** can be any value subject to the 64 Kbit limitation of pattern memory per channel.

The parenthesis command "( )" loads binary data at the channel and bit position selected by the last LOAD CHANNEL or GOTO command.

Example:

```
LOAD CHANNEL 4↵
GOTO 9↵
(11111110110111001011101010 01
10000111011001010100001100100
0010000) * 2↵
(00000001001000110100010101 10
01111000100110101011110011011
1101111) * 2↵
```

will load Channel 4 with 8 bytes of data, starting at bit 9 (decimal), with the specified binary data.

Note that this example will load Channel 4 with the same data that above ":hexadecimal" example loads into Channel 3.

Changes to the pattern data will not be reflected in the output until an UPDATE is performed.

See also: **: hex data**
CLEAR CHANNEL

GOTO
LOAD CHANNEL
UPDATE

; (REM)

Syntax:

; ***remarks***↵

remarks is any text.

Allows the user to insert remarks in any command line. Once the semicolon is received, the PI-2005 will ignore anything between it and the next carriage return and line feed.

Example:

```
LOAD CHANNEL 6; Counter↵
```

Remarks are completely ignored by the instrument and are not stored.

See Also: **! (COMMENTS)**

! (COMMENTS)

Syntax:

line **!** ***comments***↵

line is the Instruction Line number at which to place the comment

comments is any text.

Allows the user to insert comments on an instruction line. Instruction comments are preserved within PI-PAT files (.w25), are returned with program instructions via DUMP INST, and are exported with text, but have no effect on program execution. Program instructions do not terminate a program as of PI-DATS version 2.621.

Example:

```
1 BEGIN↵
2 ! Readout for 256x256 mode↵
3 SP 1 * 1↵
```

See Also: **; (REM)**
BEGIN
DUMP INST

BEGIN

Syntax: ***line*** **BEGIN** ***beginnum***↵

line is the Instruction Line number at which to begin the program

beginnum is the BEGIN Number to assign the program.

Enters a BEGIN statement at the specified Instruction Line. Example:

```
1 BEGIN 1↵
```

To delete an instruction line, simply send the line number with no arguments. Example:

```
2↵
```

will delete instruction line 2. To clear all instructions, use the CLEAR INST command.

See Also: **CLEAR INST**
DUMP INST
REPEAT
SP

BIASCLKMSGS

Syntax:

BIASCLKMSGS↵

Multiple types of Pulse Instruments card can be installed in the same CompactPCI mainframe, and certain commands, such as STATUS, are recognized by more than one type of card. Therefore, the PI-11000 uses "message modes" to determine which commands are intended for which type of card.

A BIASCLKMSGS must be sent before the initial block of commands intended for DC Bias or clock driver cards..

The BIASCLKMSGS command must be sent again if any commands have been subsequently sent to a different type of Pulse Instruments CompactPCI cards (such as Pattern Generator cards), using the command.

Example:

```
BIASCLKMSGS ; THESE COMMANDS
ARE FOR BIAS/CLK CARDS↵
ALL LOLIMB -3.00↵
ALL UPLIMB 5.00↵
MF 0 SLOT 3 CH 2 VOLT –1.80↵
MF 0 SLOT 3 CH 1 VOLT 2.3↵
CONNECT↵
ONLINE↵

PATMSGS ; THESE ARE FOR PAT
GEN CARDS
COMPILE 1↵
SEQUENCE ↵

BIASCLKMSGS ; THERE ARE FOR
BIAS/CLK CARDS↵
OFFLINE↵
```

See Also: **PATMSGS**

CLEAR
CLEAR CHANNEL
CLEAR FORMAT
CLEAR INST
CLEAR SP

Syntax: **CLEAR [CHANNEL] [FORMAT] [INST] [SP]↵**

CLEAR issued with no arguments clears all programming data. If the PI-2005 is running, CLEAR will place it in STOP mode.

If the CLEAR command is issued with the optional arguments, it will clear only the specified data or settings, i.e. pattern data (CHANNEL), channel formats (FORMAT), instructions (INST), or subpattern definitions (SP). When used with an optional argument, CLEAR leaves the run-mode unchanged.

Note that

Example:

```
CLEAR FORMAT↵
```

See Also: **DELETE**

CLKEXTECL
CLKEXTTTL
CLKINT

Syntax: **CLKEXTECL↵**
CLKEXTTTL↵
CLKINT↵

Sets the clock source of the PI-2005 to one of the following:

- External clock with ECL thresholds
- External clock with TTL thresholds
- Internal clock as specified by PERIOD command

Example:

```
CLKEXTECL↵
```

The CLK commands do not take effect until a PERIOD command has been sent.

See Also: **DUMP CLK**
PERIOD

COMPILE

Syntax: **COMPILE *beginnum*↵**

beginnum is the BEGIN Number of the program to be compiled. ***beginnum*** can be from 1 to 4,096.

Causes the Pattern Generator to compile the program with the specified BEGIN Number. Equivalent to selecting the **Hardware** menu and then clicking the **Compile** button. When RUN is sent or the Run button is clicked, the PI-2005 will run the program specified by the last COMPILE command.

If COMPILE is sent while the Pattern Generator is in the RUN Mode, the instrument will STOP and COMPILE the indicated program.

Example:

```
COMPILE 2↵
```

See Also: **UPDATE**

CYCLE

Syntax: **CYCLE↵**

Runs through the currently compiled program one time, and then stops. Equivalent to sending SEQUENCE 1, or pressing the **Seq** button with a setting of 1. Example:

```
CYCLE↵
```

See Also: **RUN**
RUNMODE
SEQUENCE
STEP
STOP

DACQMSGS

Syntax:

DACQMSGS↵

The message mode commands (BIASCLKMSGS, DACQMSGS and PATMSGS) are for GPIB communications only. Users linking directly into Pulse Instruments DLLs can ignore these commands.

Multiple types of Pulse Instruments card can be installed in the same CompactPCI mainframe, and certain commands, such as STATUS, are recognized by more than one type of card. Therefore, the PI-11000 uses “message modes” to determine which commands are intended for which type of card.

A DACQMSGS must be sent before the initial block of commands intended for the data acquisition subsystem.

The DACQMSGS command must be sent again if any commands have been subsequently sent to a different type of Pulse Instruments CompactPCI cards (such as Pattern Generator cards), using the PATMSGS command.

Example:

```
PATMSGS ; THESE COMMANDS ARE
FOR PATGEN CARDS↵
UPDATE↵
COMPILE 1↵

DACQMSGS ; THESE ARE FOR DATA
ACQUISITION CARDS
BLOCKING FALSE↵
STARTACQ ↵

PATMSGS↵
RUN↵

DACQMSGS
ACQSTATUS↵
```

If linked from a local application dacq.dll will treat DACQMSGS as a no-op, to allow for code re-use among applications that may run locally or remotely.

See Also: **BIASCLKMSGS**
PATMSGS

DEFINE SP

Syntax: **DEFINE SP *spnum* (*startbit-length*)↵**

spnum is the number of the subpattern to be defined. ***spnum*** can be from 1 to 3,276.

startbit is the position of the subpattern's Start Bit in decimal. **s*tartbit*** can be from 1 to 65,535.

length is the length of the subpattern, in bits. ***length*** must be at least 20. ***length*** + ***startbit*** cannot be greater than 65,536.

Defines a subpattern with the specified parameters. Equivalent to defining a subpattern in the **Edit:Edit Subpatterns** dialog box.

Example:

```
DEFINE SP 4 (21-32)↵
```

To remove a subpattern definition, use the DELETE SP command or else re-define the subpattern with a length of zero.

Example:

```
DEFINE SP 4 (21-0)↵
```

Note that changes to the subpattern definitions will not be reflected in the output until **both** an UPDATE and a COMPILE are performed.

See Also: **BEGIN**
COMPILE
DELETE SP
REPEAT
UPDATE

DELAY

Syntax: **DELAY *chnum* - *delay*↵**

chnum is the channel number for which to set the programmable delay.

delay is the programmable delay, in multiples of 2.5 ps, from 0-2331.

Sets a delay of ***delay*** * 2.5 ps on the specified channel. If the specified channel does not have the optional delay feature, the PI-2005 will return error code 40. Use NUMCHAN and GETDELAY to determine which installed cards have the delay feature.

Channel formatting commands such as DELAY take effect immediately.

Example:

```
DELAY 8 – 1500↵
```

will set the delay to approximately 3.75 ns, relative to other channels on the same card.

See Also: **CLEAR FORMAT**
DUMP DELAY
GETDELAY
HIZ
INVERT
RZ
NONINV
NORM
NRZ

DELETE

Syntax: **DELETE SP*spnum*↵**

spnum is the subpattern definition to delete.

Deletes a subpattern definition. Note that the change will not affect the output of the PI-2005 until both an UPDATE and a COMPILE have been issued.

Example:

```
DELETE SP4↵
```

See Also: **DEFINE SP**

DUMP CHANNEL

Syntax: **DUMP CHANNEL *chnum* ↵**

chnum is the desired channel number

DUMP CHANNEL is deprecated, and provided for backward compatibility purposes only. All new applications should use the new DUMP PATTERN command.

Returns all pattern data from the specified channel in hexadecimal format. Example:

```
DUMP CHANNEL 1↵
```

The output must be read, one line at a time, from the instrument via a series of GPIB read commands. Sample output line:

```
:20000000388ADB65B17000000000
00000000000000000000000000000
000000000000000↵
```

See the **: hex data** command definition above for a description of the data format. Each returned line will contain a 9 character header and 32 pattern bytes. Subsequent reads will return additional lines until the PI-2005 returns the standard File Termination Sequence:

```
1A OD OA
```

See Also: **: HEX DATA**
(BINARY DATA)
CLEAR CHANNEL
DUMP PATTERN

GOTO
LOAD CHANNEL

DUMP CLK

Syntax: **DUMP CLK↵**

Returns the current clock setting (Internal, External TTL, or External ECL). Example:

```
DUMP CLK↵
```

The output must be read from the instrument via a GPIB read command. Sample output line:

```
CLKINT↵
```

The next read will then return the standard File Termination Sequence:

```
1A OD OA
```

See Also: **CLK**
PERIOD

DUMP DELAY

Syntax: **DUMP DELAY↵**

Returns the current delay settings, in 2.5 picosecond multiples, for all channels in the Pattern Generator that have a non-zero delay setting. Example:

```
DUMP DELAY↵
```

The output must be read from the instrument via a series of GPIB read commands. Sample output line:

```
DELAY 1 - 1200↵
```

If channels are installed that do not have the optional delay feature, the returned string for that channel will be a comment line such as:

```
;DELAY 17 – NO DELAY↵
```

Subsequent reads will return delay settings for additional channels until the PI-2005 returns the standard File Termination Sequence:

```
1A OD OA
```

DUMP FORMAT

Syntax: **DUMP FORMAT↵**

DUMP FORMAT is deprecated, and provided for backward compatibility purposes only. All new applications should use the new DUMP FORMATS command.

Returns the current channel format setting, in the PI-5800 block format. Example usage:

```
DUMP FORMAT↵
```

The output must be read from the instrument via a GPIB read command. Sample output line:

```
:20000048484848484848484848↵
```

See Also: **DUMP FORMATS**

DUMP FORMATS

Syntax: **DUMP FORMAT [*chnum*]↵**

chnum optional argument for the desired channel number

Returns the current channel format settings for the specified channel, unless the chnum argument is omitted, in which case it returns the channel format settings for all installed channels. Example:

```
DUMP FORMATS 1↵
```

The output must be read from the instrument via a series of GPIB read commands. Sample output from DUMP FORMATS 1:

```
CHANNEL 1 INVERT↵
CHANNEL 1 NORMAL↵
CHANNEL 1 NRZ↵
```

Sample output from DUMP FORMATS:

```
CHANNEL 1 INVERT↵
CHANNEL 1 HIZ↵
CHANNEL 1 HIZ↵
CHANNEL 2 HIZ↵
CHANNEL 2 HIZ↵
CHANNEL 2 HIZ↵
CHANNEL 3 HIZ↵
CHANNEL 3 HIZ↵
CHANNEL 3 HIZ↵
    .
    .
    .
```

Each queried channel will return three lines, one each for INVERT/NONINV, HIZ/NORMAL, and RZ/NRZ.

Subsequent reads will return format settings for additional channels until the PI-2005 returns the standard File Termination Sequence:

```
1A OD OA
```

See Also: **HIZ**
INV
NONINV
NORMAL
NRZ
RZ

DUMP INST

Syntax: **DUMP INST [*linenum*]↵**

linenum optional argument for the desired line number

Returns instruction lines from the PI-2005. If *linenum* is included, the PI-2005 will return one instruction line. If *linenum* is omitted, the PI-2005 will return all entered instruction lines.

Example:

```
DUMP INST↵
```

The output must be read from the instrument, one line at a time, via a series of GPIB read commands. Sample output lines:

```
1 BEGIN 1↵
2 SP 1*16↵
```

Subsequent reads will return additional instruction lines until there are no more lines to send, at which time the PI-2005 will return the standard GPIB termination string:

```
1A OD OA
```

See Also: **BEGIN**
REPEAT
SP

DUMP NAMES

Syntax: **DUMP NAMES [*chnum*]↵**

chnum optional argument for the desired channel number

Returns user-defined channel names from the PI-2005. If *chnum* is included, the PI-2005 will return one channel name. If *chnum* is omitted, the PI-2005 will return all user-defined channel names.

Example:

```
DUMP NAMES↵
```

The output must be read from the instrument, one line at a time, via a series of GPIB Read commands. Sample output lines:

```
NAME CHANNEL 1 @PHI1↵
NAME CHANNEL 2 @PHI2↵
```

Subsequent reads will return additional channel names until there are no more names to send, at which time the PI-2005 will return the standard GPIB termination string:

```
1A OD OA
```

Channels without user-defined names will not be returned unless explicitly requested with the *chnum* argument.

See Also: **NAME CHANNEL**

DUMP PATTERN

Syntax: **DUMP PATTERN *chnum* *(startbit-numbits)*↵**

chnum desired channel number
startbit first bit number to return
chnum number of bits to return

Returns pattern data from the specified channel in hexadecimal format. Example:

```
    DUMP PATTERN 1 (1-512)↵
```

The output must be read, one line at a time, from the instrument via a series of GPIB read commands. Sample output line:

```
    LOAD CHANNEL 1↵
    :20000000388ADB65B17000000000
    00000000000000000000000000000
    000000000000000↵
```

See the **: hex data** command definition above for a description of the data format. Each returned line will contain a 9 character header and up to 32 data bytes. Subsequent reads will return additional lines of data until the end of desired pattern data, at which time the PI-2005 will return the standard File Termination Sequence:

```
    1A OD OA
```

Note that the returned data is in hex format, and that only whole bytes are returned. Use **DUMP SPDEF** to determine how much pattern data needs to be dumped.

See Also: **: HEX DATA**
(BINARY DATA)
CLEAR CHANNEL
GOTO
LOAD CHANNEL

DUMP PERIOD

Syntax: **DUMP PERIOD↵**

Returns the current Internal Period in nanoseconds or microseconds. Example:

```
    DUMP PERIOD↵
```

The output must be read from the instrument via a GPIB read command. Sample output:

```
    PERIOD 100.0000 US↵
```

The next read will then return the standard File Termination Sequence:

```
    1A OD OA
```

See Also: **CLKINT**
CLKEXTECL
CLKEXTTTL
PERIOD

DUMP SP

Syntax: **DUMP SP↵**

DUMP SP is deprecated, and provided for backward compatibility purposes only. All new applications should use the new DUMP SPDEF command.

Returns all subpattern definitions from the PI-2005. Example:

```
    DUMP SP↵
```

The output must be read from the instrument via a GPIB read command. Sample output line:

```
    0001 (0001-0020)↵
```

Subsequent reads will return additional subpattern definitions until the PI-2005 returns the standard File Termination Sequence:

```
    1A OD OA
```

See Also: **DEFINE SP**
DUMP SPDEF
CLEAR SP

DUMP SPDEF

Syntax: **DUMP SPDEF [*spnum]*↵**

spnum optional argument for the subpattern number

Returns desired subpattern definitions from the PI-2005. Example:

```
    DUMP SPDEF 5↵
```

The output must be read from the instrument via a GPIB read command. Sample output line:

```
    DEFINE SP 0005 (0001-0020)↵
```

If the *spnum* argument is omitted, the PI-2005 will return all the subpattern definitions. Subsequent reads will return additional subpattern definitions until the PI-2005 returns the standard File Termination Sequence:

```
    1A OD OA
```

See Also: **DEFINE SP**
DUMP SP
CLEAR SP

DUMP STASTP

Syntax: **DUMP STASTP↵**

Returns the numeric specification for the current Start/Stop edge combination being used to control the System Clock from the Auxiliary Connector. Example:

```
DUMP STASTP↵
```

The output must be read from the instrument via a GPIB read command. Sample output line:

```
STASTP 6↵
```

The next read will then return the standard File Termination Sequence:

```
1A OD OA
```

The numeric specifications refer to the following edge combinations:

0 Disable
6 Start On + Edge, Stop on + Edge
7. Start/Stop (Gate mode)

. See Also: **PAUSE**
RUN
RUNMODE
SEQUENCE
STASTP
STEP

GETDELAY

Syntax: **GETDELAY↵**

Returns information about which cards are NOT "Delay" cards.

Example:

```
GETDELAY↵
```

The command returns a decimal number where each bit represents a "Non-Delay" flag for each installed pattern card:

Bit 1 (1) - Set if Card 1 is NOT Delay

Bit 2 (2) - Set if Card 2 is NOT Delay

Bit 3 (4) - Set if Card 3 is NOT Delay

Bit 4 (8) - Set if Card 4 is NOT Delay

If the PI-2005 has 4 pattern cards installed (64 channels), and cards 1, 2, and 4 are NOT "Delay" cards, the GETDELAY command would return 11 (1+2+8).

The output must be read from the instrument via a GPIB read command. Sample output line:

```
11↵
```

See Also: **DUMP DELAY**
NUMCHANNELS

GOTO
GOTO CHANNEL
GOTO START/END SP

Syntax: **GOTO *bitnum*↵**
GOTO CHANNEL *chnum*
GOTO START/END
SP*spnum* [± *offset*]↵

bitnum is the bit number of the channel at which to begin loading binary data.

chnum is the channel number at which to begin loading data

spnum is the number of the subpattern definition at which to begin loading binary data.

offset is the desired offset for the bit position (optional)

Sets the bit position at which to load binary data.

If GOTO is followed by CHANNEL, then this command will specify the channel number for loading binary or hexadecimal data: Example:

```
GOTO CHANNEL 8↵
```

Binary or hexadecimal load commands will load data into the channel specified by the last GOTO CHANNEL or LOAD CHANNEL command

In this usage, GOTO CHANNEL is equivalent to LOAD CHANNEL, and is supported for backward compatibility with command files written for the PI-5800.

If GOTO is followed by a number, then the PI-2005 will begin loading binary data at that bit number on the last selected channel. Example:

```
GOTO 21↵
```

If GOTO is followed by START SPx or END SPx, the PI-2005 will begin loading binary data at the specified bit of the specified subpattern, plus any offset. Example:

```
GOTO START SP3↵
```

```
GOTO END SP3-12↵
```

Note that there is no space character between SP and the subpattern number.

See Also: **(binary data)**
: hexdata
LOAD CHANNEL

HIZ

Syntax: **CH *chnum* HIZ↵**

chnum is the number of the desired channel to set to HIZ (Tri-State) format.

Sets the specified channel to HIZ mode. Equivalent to turning on HIZ in the **Edit:Edit Formats** dialog box. Example:

```
CHANNEL 8 HIZ↵
```

Channel formatting commands such as HIZ take effect immediately when received by the PI-2005.

To place a channel in Tri-State mode for the duration of a single instruction line, use the Tri-State-On-The-Fly feature of the **SP** command.

See Also: **CLEAR FORMAT**
DUMP FORMATS
INV
NONINV
NORMAL
NRZ
RZ

I2CMODE

Syntax: **I2CMODE [7|10], [3|5]↵**

Sets the I^{2}C mode to 7- or 10-bit addressing, and sets the I^{2}C signaling voltage to 3.3 V or 5 V. For example the following command:

```
I2CMODE 10, 3↵
```

will set the I^{2}C mode for 10-bits, 3.3 V. The default mode is 7 bits, 3.3 V. All arguments for I2CMODE are decimal.

See Also: **I2CREAD**
I2CREADREG
I2CWRITE
I2CWRITEREG

I2CREAD

Syntax: **I2CREAD *dev_addr, n*↵**

dev_addr the 7 or 10-bit device address, in hex.

n the number of bytes to read, in decimal

Sends an I^{2}C Read command to the specified device address, and reads back the requested number of bytes. For example the following command:

```
I2CREAD 55, 4↵
```

will read 4 bytes from the device at address 0x55. When queried, the PI-2005 will return the bytes read from the device, as a series of comma-delimited values, e.g.:

```
00, 0F, 01, 04↵
```

See Also: **I2CMODE**
I2CREADREG
I2CWRITE
I2CWRITEREG

I2CREADREG

Syntax: **I2CREADREG *dev_addr, reg_addr*↵**

dev_addr the 7- or 10-bit device address, in hex

reg_addr the 8-bit register address to read, in hex

Sends an I^{2}C Read command, with a register address, to the specified device address, and then reads back a single byte. For example the following command:

```
I2CREAD 55, 0F↵
```

will read a single byte from register address 0x0F of the device at address 0x55. When queried, the PI-2005 will return the byte as a string, e.g.:

```
OA↵
```

Note that the I2CREADREG command sends an I^{2}C Start command, the device address and register address, followed by a Restart command. Note that only certain devices (e.g. some Agilent parts) support the “register” I^{2}C commands.

See Also: **I2CMODE**
I2CREAD
I2CWRITE
I2CWRITEREG

I2CWRITE

Syntax: **I2CWRITE *dev_addr, data [, data1][, data2] . . . [, datan]*↵**

dev_addr the 7- or 10-bit device address, in hex

data the byte or bytes to write to the device, in hex

Sends an I^{2}C Write command to the specified device address, followed by one or more data bytes. Example:

```
I2CWRITE 55,01↵
```

I2CWRITE may be used to write several consecutive bytes, e.g.:

```
I2CWRITE 55,01,03,04,1A↵
```

Most I^{2}C devices will automatically increment their internal destination (register) address with each received data byte.

See Also: **I2CMODE**
I2CREAD
I2CREADREG
I2CWRITEREG

I2CWRITEREG

Syntax: **I2CWRITEREG *dev_addr, reg_addr, data [, data1][, data2] . . . [, datan]*↵**

dev_addr the 7- or 10-bit device address, in hex

reg_addr the 8-bit register address, in hex

data the byte or bytes to write to the device, in hex

Sends an I^{2}C Write command to the specified device address and register, followed by one or more data bytes. Example:

```
I2CWRITE 55,0F,01↵
```

I2CWRITE may be used to write several consecutive bytes, e.g.:

```
I2CWRITE 55,0F,01,03,04,1A↵
```

Most I^{2}C devices will automatically increment their internal destination (register) address with each received data byte.

Note that the I2CWRITEREG command sends an I^{2}C Start command, the device address and register address, followed by a Restart command and then the data bytes. Only certain devices (e.g. some Agilent parts) support the “register” I^{2}C commands.

See Also: **I2CMODE**
I2CREAD
I2CREADREG
I2CWRITE

ID

Syntax: **ID↵**

Causes the Pattern Generator to return its model number. Example:

```
ID↵
```

Sample returned string:

```
PI-2005↵
```

See Also: **DUMP** commands
GETDELAY
NUMCHAN
STATUS

INVERT

Syntax: **CHANNEL *chnum* INVERT↵**

chnum is the number of the desired channel to invert

Sets the specified channel to Inverted output. Equivalent to setting an Inverted format from the **Edit:Edit Formats** dialog box. Example:

```
CHANNEL 13 INVERT↵
```

Channel formatting commands such as INVERT take effect immediately when received by the PI-2005.

See Also: **CLEAR FORMAT**
DUMP FORMATS
HIZ
NONINV
NORMAL
RZ
NRZ

LOAD CHANNEL

Syntax: **LOAD CHANNEL *chnum***

chnum is the channel number at which to begin loading data

Sets the channel position at which to load data. Example:

```
LOAD CHANNEL 12↵
```

Binary or hexadecimal load commands will load data into the channel specified by the last LOAD CHANNEL or GOTO CHANNEL command.

See Also: **(binary data)**
: hexdata
DUMP PATTERN
GOTO CHANNEL

NAME CHANNEL

Syntax: **NAME CHANNEL *chnum* *@name***

chnum is the number of the desired channel to name

name is the desired name for the channel

Defines a channel name. Example:

```
NAME CHANNEL 1 @PHI1↵
```

Channel naming applies only to the display in PI-PLOT, and commands for loading or formatting channels still must use channel numbers.

See Also: **DUMP NAMES**

NONINV

Syntax: **CHANNEL *chnum* NONINV ↵**

chnum is the number of the desired channel from which to remove inverted format

Sets the specified channel to Non-Inverted output. Equivalent to selecting Non-Inverted format from the **Edit:Edit Formats** dialog box. Example:

```
CHANNEL 13 NONINV↵
```

Channel formatting commands such as NONINV take effect immediately when received by the PI-2005.

See Also: **CLEAR FORMAT**
DUMP FORMATS
HIZ
INVERT
NORMAL
RZ
NRZ

NORMAL

Syntax: **CHANNEL *chnum* NORMAL↵**

chnum is the number of the desired channel to set to Normal format.

Sets the specified channel to Normal (i.e. not HIZ) data output. Equivalent to turning off HIZ in the **Edit:Edit Formats** dialog box. Example:

```
CHANNEL 13 NORMAL↵
```

Channel formatting commands such as NORM take effect immediately when received by the PI-2005.

See Also: **CLEAR FORMAT**
DUMP FORMATS
HIZ
INV
NONINV
RZ
NRZ

NRZ

Syntax: **CHANNEL *chnum* NRZ↵**

chnum is the number of the desired channel to set to Not-Return-to-Zero format.

Sets the specified channel to NRZ data output. Equivalent to turning off RZ in the **Edit:Edit Formats** dialog box. Example:

```
CHANNEL 13 NRZ↵
```

Channel formatting commands such as NRZ take effect immediately when received by the PI-2005.

See Also: **CLEAR FORMAT**
DUMP FORMATS
HIZ
INV
NONINV
NORMAL
RZ

NUMCHAN

Syntax: **NUMCHAN↵**

Returns the number of channels installed on the PI-2005. Example:

```
NUMCHAN↵
```

The output must be read from the instrument via a GPIB read command. Sample output line:

```
16 CHANNELS↵
```

See Also: **ID**
RUNMODE
STATUS

PATMSGS

Syntax:

PATMSGS↵

Multiple types of Pulse Instruments card can be installed in the same CompactPCI mainframe, and certain commands, such as STATUS, are recognized by more than one type of card. Therefore, the PI-11000 uses “message modes” to determine which commands are intended for which type of card.

A PATMSGS must be sent before the initial block of commands intended for Pattern Generator cards.

The PATMSGS command must be sent again if any commands have been subsequently sent to a different type of Pulse Instruments CompactPCI cards (such as DC Bias or clock driver cards) using the BIASCLKMSGS command.

Example:

```
BIASCLKMSGS ; THESE COMMANDS
ARE FOR BIAS/CLK CARDS↵
ALL LOLIMB -3.00↵
ALL UPLIMB 5.00↵
MF 0 SLOT 3 CH 2 VOLT –1.80↵
MF 0 SLOT 3 CH 1 VOLT 2.3↵
CONNECT↵
ONLINE↵

PATMSGS ; THESE ARE FOR
PATTERN GENERATOR CARDS
COMPILE 1↵
SEQUENCE ↵
SEQUENCE ↵
SEQUENCE ↵

BIASCLKMSGS ; THERE ARE FOR
BIAS/CLK CARDS↵
OFFLINE↵
```

See Also: **BIASCLKMSGS**
DACQMSGS

PERIOD…NS, PERIOD…US, PERIOD…MS

Syntax: **PERIOD *period units*↵**

period is the numeric quantity for the Internal Period.

units is the unit quantity of the Internal Period. ***units*** must be NS, US, or MS.

Sets the Internal Period when the System Clock is Internal. For example,

```
PERIOD 40.022 NS↵
```

will set the Internal Period to 40.022 ns. The desired period must be between 4 ns and 83.31 µs, and must be in increments of 0.0001 ns.

See Also:
CLKINT
CLKEXTTTL
CLKEXTECL
DUMP PERIOD

POWERDOWN

Syntax: **POWERDOWN↵**

Causes the PI-2005 to begin the shut-down procedure. The PI-2005 is safe to turn off 30 seconds after the green LEDs on the Pattern Cards turn off. Example:

```
POWERDOWN↵
```

REPEAT

Syntax: ***line* REPEAT *from* * *loops* ↵**

line is the instruction line number on which to enter the REPEAT statement.

from is the first instruction line of the desired loop. The last line of the desired loop is always ***line*** – 1.

loops is the number of times to repeat the loop.

Enters a REPEAT statement on line ***line***. The loop will begin at ***from*** and repeat ***loops*** times. Example:

```
7 REPEAT 4*16↵
```

To delete an instruction line, simply send the line number with no arguments. Example:

```
2↵
```

will delete instruction line 2. To clear all instructions, use the CLEAR INST command.

See also:
BEGIN
CLEAR INST
SP

RUN

Syntax: **RUN↵**

Places the instrument in the Run Mode. Equivalent to clicking the **Run** button. In the Run Mode the Pattern Generator will continuously repeat the most recently compiled program. Example:

```
RUN↵
```

See Also:
CYCLE
RUNMODE
SEQUENCE
STEP
STOP

RUNMODE

Syntax: **RUNMODE↵**

Requests the Run Mode from the PI-2005, after which the instrument should be read.

Example:

```
RUNMODE↵
```

The output must be read from the instrument via a GPIB read command. Sample returned output:

```
RUNNING↵
```

See Also:
CYCLE
RUN
SEQUENCE
STEP
STOP

RZ

Syntax: **CHANNEL *chnum* RZ↵**

chnum is the number of the desired channel to set to Return-to-Zero format.

Sets the specified channel to RZ data output. Equivalent to turning on RZ in the **Edit:Edit Formats** dialog box. Example:

```
CHANNEL 13 RZ↵
```

Channel formatting commands such as RZ take effect immediately when received by the PI-2005.

See Also: **CLEAR FORMAT**
DUMP FORMATS
HIZ
INV
NONINV
NORMAL
NRZ

SEQUENCE

Syntax: **SEQUENCE [*seqs*]↵**

seqs is the optional number of times to run.

Causes the PI-2005 to run the most recently compiled program ***seqs*** times. If ***seqs*** is omitted, the PI-2005 will run the program one time. Example:

```
SEQUENCE 64↵
```

See also: **CYCLE**
RUN
RUNMODE
STEP
STOP

SETGPIBADDR

Syntax: **SETGPIBADDR *addr*↵**

addr is desired new GPIB address for the PI-2005, in the range from 1-30.

Changes the GPIB address of the PI-2005.

Note that subsequent communications with the PI-2005 will require using the new address. Example:

```
SETGPIBADDR 15↵
```

SP

Syntax: ***linenum* SP *spnum* * *repeat [tristate]*↵**

linenum is the Instruction Line number at which to enter the SUBPATTERN statement.

spnum is the Subpattern Number.

repeat is the number of times to repeat the specified Subpattern.

tristate is the optional argument specifying which channel pairs to Tri-State during the instruction line.

Enters a SUBPATTERN statement at the specified Instruction Line. For example, the command:

```
10 SP 4*32↵
```

will enter a SUBPATTERN 4 x 32 statement at Instruction Line 10. Channel pairs can be put in Tri-State mode “on-the-fly” for the duration of a single instruction line. To place a channel pair in Tri-State mode, place the argument T with the odd channel number following the SP instruction. Example:

```
10 SP 4*32 T1 T5↵
```

will cause channels 1, 2, 5 and 6 to go into Tri-State mode during the execution of program line 10. These channels will return to their previously formatted outputs during the execution of line 11. Please see **Section 5. Tri-State On-The-Fly Notes** for cautionary notes about using this feature.

To place channels in Tri-State mode during the execution of an entire program, use the **HIZ** channel formatting command.

To delete an instruction line, simply send the line number with no arguments. Example:

```
2↵
```

will delete instruction line 2. To clear all instructions, use the CLEAR INST command.

See Also: **BEGIN**
CLEAR INST
HIZ
REPEAT

STASTP

Syntax: **STASTP *edge*↵**

edge is the edge condition to be received from the SMB Connector. ***edge*** must be 0, 1 or 6.

Disables or selects one of 3 possible edge combinations from the Rear Panel SMB Connector to be used to control the System Clock.

0	Disable
6	Start On + Edge, Stop on + Edge
7	Start On/Off

. Example:

```
STASTP 6↵
```

See Also: **DUMP STASTP**

STATUS

Syntax: **STATUS↵**

Causes the Pattern Generator to return a Status File containing information on the status of the System. This command also clears any pending errors and returns their corresponding error codes.

The Status File will contain the following information:

Run/Stop Status
Error Number
1A 0D 0A EOI

The final four bytes are the standard File Termination Sequence.

If there are any errors pending you will not be able to send data or commands to the PI-2005 until they are cleared with the STATUS command.

STEP

Syntax: **STEP *steps*↵**

steps is the number of subpattern repeats to step through. *Steps* must be from 1 to 99,999.

Causes the Pattern Generator to run the most recently compiled program output for a specified number of subpattern loops. . Example:

```
STEP 6↵
```

Equivalent to clicking the **Step** button.

See also: **RUN**
RUNMODE
SEQUENCE
STOP

STOP

Syntax: **STOP↵**

Places the Pattern Generator in the Stop Mode. Equivalent to clicking the **Stp** button. . Example:

```
STOP↵
```

STOP sets the program counter back to the beginning of the program.

See Also: **RUN**
RUNMODE
SEQUENCE

UPDATE

Syntax: **UPDATE↵**

Causes the PI-2005 to write any changes in pattern data to the pattern cards. Changes to pattern data will not be reflected in the output until an UPDATE is performed.

Changes to subpattern definitions will not be reflected in the output until both an UPDATE and a COMPILE are performed. Example:

```
UPDATE↵
```

See Also: **:hex data**
(binary data)
COMPILE
DEFINE SP

9. GPIB ERROR CODES

Code	Error
00	No error. This is the normal response to a Status command.
05	NUMBER TOO LARGE - An argument is outside the valid range for the command.
06	SYNTAX ERROR -INPUT IGNORED - Unrecognized command or bad command syntax was sent.
10	OVERFLOW - A number was entered larger than 65,535, an argument was entered that is greater than the maximum allowed for the command, or an attempt was made to enter pattern data beyond the 64 Kbit boundary.
11	NUMBER TOO SMALL - An attempt was made to enter a number or argument less than 0.
12	SUBPATTERN TOO SHORT - An attempt was made to define a subpattern shorter than 12 or 20 bits. See Section …
14	COMPILE ERROR - An error occurred during program compilation. Either an invalid instruction sequence exists, or referenced subpatterns have not been defined.
18	INVALID BEGIN NUMBER - A COMPILE command was sent with an argument for a BEGIN number that does not exist.
22	INVALID SUBPATTERN NO. - A compile has been attempted of a program that attempts to use a subpattern, which is not yet defined.
24	INVALID PERIOD - A PERIOD command was sent with an argument outside the valid range of 4 NS – 83.31 US (250 MHz – 12 kHz)
34	COMMAND NOT SUPPORTED – A command was sent that was supported by the PI-5800, but not by the PI-2005. See **Section 6.4.3. GPIB Command Mismatches for PI-2005**
35	OVERLAPPING SUBPATTERN DEFINITION – Each bit position may belong to only one subpattern definition. See **Section 6.2. Overlapping subpattern definitions**
36	INVALID START/STOP SETTING – Valid Start/Stop settings for the PI-2005 are 0, 6, and 7. See GPIB command **STASTP**
37	INVALID NUMBER OF STEPS – A number of steps was sent that is outside the valid range of 1 – 99,999
38	INVALID CHANNEL NUMBER – A channel number argument was sent that is either < 1 or larger than the number of installed channels
39	INVALID GPIB ADDRESS – A SETGPIBADDR command was sent with a new address argument outside the valid range of 1 – 30.
40	CHANNEL DOES NOT HAVE DELAY OPTION – A DELAY command was sent for a channel that does not have the optional delay feature
41	INVALID DELAY ARGUMENT – A DELAY command was sent with an argument outside the valid range of 0 – 2,331

Code	Error
42	INVALID TRI-STATE CHANNEL – A subpattern instruction with Tri-State-On-The-Fly parameters was sent incorrectly. TSOTF parameters are specified with odd channels only, but the channels will tri-state in pairs. See **Section 3.4.2.2.5. Tri-State On-The-Fly (TSOTF)**
43	PI-2005 HAS NO ACTIVE WINDOW – Commands were sent to the PI-2005, but PI-PAT (running on the local instrument) has no open document windows. From an attached monitor, keyboard and mouse, choose **File:New** to create a new document.
46	SRAM EXCEEDED – An instruction was entered (or a COMPILE was attempted) that caused an inner loop count greater than 256 K (262,144) loops.
47	FIFO EXCEEDED – An instruction was entered (or a COMPILE was attempted) that caused an outer loop count greater than the FIFO installed on the PI-2005 (either 4 K or 64 K).
48	INVALID BIT NUMBER – A command was sent with a bit position argument outside the valid range 0 – 65,535. The bit position may have been specified explicitly, or with a subpattern offset (e.g. GOTO START SP 1 - 10)
49	NO INSTRUCTIONS – An attempt was made to COMPILE a program that contains no subpattern instructions.
50	REPEAT TOO LARGE – An attempt was made to enter a REPEAT statement with a repeat value greater than 1,023.

10. INDEX

Document change history

03/11/16
Added Enter Pattern and Compile/Unroll features. Updated specifications.

9/22/08
Corrected syntax and example for NAME CHANNEL to include the channel number.

4/13/07
Clarified that DYNAMIC UPDATE is not supported. Added notes about GPIBCom source code.

12/11/06
Reversed 11/8 change to preserve backward compatibility, and note about pointer to filename string

11/8/06
Updated Automation class with new 3rd parameter

08/21/06
Added location of PI-PAT type library

6/6/06
Corrected footers and page numbers

4/24/06
Added note that a comment ends a valid BEGIN block.
Added documentation and warnings for instruction line comments import/export

11/10/05
Added warning about PI-21000 PECL outputs
Added warning about PI-2110x Pattern Card input cable connection before power-up
Added photo of GMS-165 CPU card
Added note about GPIB-PMC. Still need to add photo.
Updated rear-panel photo with new P/S and decals
Add front-panel photo with power switch

10/11/05
Added description of error code 00 to GPIB error table.
Corrected information about Windows 2000 installations
Added description of title bar "synch status" indicator

12/13/04
Corrected labeling of pin numbers on Micro D output from Pattern Card.
Added Windows 2000 as a supported OS.

1/9/02
Corrected STASTP arguments list
Added optional argument for SEQUENCE [steps]
Added CYCLE
Corrected description of DELAY command
Corrected description of ID command
Corrected valid PERIOD arguments
Corrected PERIOD effect on runmode
Corrected syntax for HIZ
Corrected syntax for NORMAL command
Delete erroneous PAUSE command

New features and bugfixes
New DUMP commands

```
        DUMP PATTERN [chnum]
        DUMP SPDEF [spnum]
        DUMP FORMATS [chnum]
        DUMP DELAY
```

New, optional argument for DUMP INST [linenum]
New support for 1-2-3 style numbering
Added error code table
CLEAR command now puts pattern generator in STOP mode