Installation, Verification and Programming Manual

HP 70330A 300MHz Pulse Generator

HP Part No. 70330-90011
Printed in FRG November 1991

Herrenberger Straße 130, 7030 Böblingen, Germany

Installation and Verification of the 300MHz Pulse Generator

The HP 70330A 300MHz Pulse Generator

The HP 70330A 300MHz Pulse Generator module is a 5/8-width module that must be installed into an HP 70001A mainframe.

The Operation Verification Tests

The operation verification tests give you an 80 percent confidence level that the pulse generator is operating correctly and meeting its specifications. You need a computer system and test equipment to perform the operation verification tests on the pulse generator.

In This Book

This book helps you install and verify the operation of the pulse generator. It provides step-by-step instructions of the installation and verification processes.

Chapter 1 describes the HP 70330A 300MHz Pulse Generator, and the steps required to install this product.

Chapter 2 describes the steps involved in verifying the operation of the module.

Chapter 3, 4, 5, and 6 contain information about programming the pulse generator.

Appendix A lists the specifications for the module.

Appendix B lists the error messages returned by the module.

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

This Hewlett-Packard instrument product is warranted against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard. Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error-free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HEWLETT-PACKARD SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

Safety Symbols

The following safety symbols are used throughout this manual. Familiarize yourself with each symbol and its meaning before operating this instrument.

Warning

The *warning* sign denotes a hazard. It calls attention to a procedure that, if not correctly performed or adhered to, could result in injury or loss of life. Do not proceed beyond a *warning* sign until you have fully understood and met the indicated conditions.

Caution

The *caution* sign denotes a hazard. It calls attention to a procedure that, if not correctly performed or adhered to, could result in damage to or destruction of the instrument. Do not proceed beyond a *caution* sign until you have fully understood and met the indicated conditions.

General Safety Considerations

Warning

***Before switching this instrument on*, make sure it has been properly grounded through the protective conductor of the ac power cable to a socket outlet provided with protective earth contact.**

Any interruption of the protective (grounding) conductor, inside or outside the instrument, or disconnection of the protective earth terminal can result in personal injury.

Warning

There are many points in the instrument that can, if contacted, cause personal injury. Be extremely careful.

Any adjustments or service procedures that require operation of the instrument with protective covers removed should be performed only by trained service personnel.

Caution

Before switching this instrument on, make sure its primary power circuitry has been adapted to the voltage of the ac power source.

Failure to set the ac power input to the correct voltage could cause damage to the instrument when the ac power cable is plugged in.

Contents

1. Installation

Tools needed 1-1
Installation overview 1-1
Installing the HP 70330A 300MHz Pulse Generator 1-2
Optional: Changing the HP-IB/HP-MSIB Address of the HP 70330A Pulse Generator Module 1-4
Possible Problems after Switching On 1-5

2. Operation Verification Testing

Introduction 2-1
Recommended Test Equipment 2-1
Step 1. Check the HP-MSIB/HP-IB 2-2
Step 2 Check the Pulse Generation Circuitry 2-2
The Operation Verification Program Listing 2-3

3. Programming the HP 70330A

Introduction 3-1
A Brief Overview of the Instrument 3-1
Interface Types 3-1
The HP 70330A HP-IB Capabilities 3-2
Remote Display Indicators 3-2
Command Messages 3-3
Message Exchange 3-3
The Input Queue 3-3
Clearing the Input Queue 3-3
The Output Queue 3-4
The Error Queue 3-4
Syntax Diagram Conventions 3-5
Short Form and Long Form 3-5
Command and Query Syntax 3-5

4. Remote Commands, Part I: Common Commands

*CLS 4-1
*ESE 4-2
*ESE? 4-3
*ESR? 4-3
*IDN? 4-4
*LRN? 4-4
*OPC 4-5
*OPC? 4-5
*RCL 4-5
*RST 4-6
*SAV 4-7
*SRE 4-8
*SRE? 4-8
*STB? 4-9

*TRG 4-9
*TST? 4-10
*WAI 4-10

5. Remote Commands, Part II: Device Commands
:DIAGnostic 5-3
:DIAGnostic:BOOT:SOURce 5-3
:DIAGnostic:BOOT:SOURce? 5-3
:INPut 5-4
:INPut:TRIGger:MODE 5-4
:INPut:TRIGger:MODE? 5-5
:INPut:TRIGger:SLOPe 5-5
:INPut:TRIGger:SLOPe? 5-5
:INPut:TRIGger:STATe 5-6
:INPut:TRIGger:STATe? 5-6
:INPut:TRIGger:THReshold 5-6
:INPut:TRIGger:THReshold? 5-7
:OUTPut 5-8
:OUTPut:PULSe:CSTate 5-8
:OUTPut:PULSe:CSTate? 5-8
:OUTPut:PULSe:POLarity 5-9
:OUTPut:PULSe:POLarity? 5-9
:OUTPut:PULSe:STATe 5-9
:OUTPut:PULSe:STATe? 5-10
:PULSe 5-11
:PULSe:COUNt 5-11
:PULSe:COUNt? 5-11
:PULSe:EDGE:LEADing 5-12
:PULSe:EDGE:LEADing? 5-13
:PULSe:EDGE:TRAiLing 5-13
:PULSe:EDGE:TRAiLing? 5-14
:PULSe:LEVel:AMPLitude 5-15
:PULSe:LEVel:AMPLitude? 5-16
:PULSe:LEVel:HIGH 5-16
:PULSe:LEVel:HIGH? 5-16
:PULSe:LEVel:LIMit 5-17
:PULSe:LEVel:LIMit? 5-17
:PULSe:LEVel:LIMit:AMPLitude? 5-17
:PULSe:LEVel:LIMit:HIGH? 5-18
:PULSe:LEVel:LIMit:LOW? 5-18
:PULSe:LEVel:LIMit:OFFSet? 5-18
:PULSe:LEVel:LOW 5-18
:PULSe:LEVel:LOW? 5-19
:PULSe:LEVel:OFFSet 5-19
:PULSe:LEVel:OFFSet? 5-20
:PULSe:TIMing:DELay 5-20
:PULSe:TIMing:DELay? 5-21
:PULSe:TIMing:DOUBle 5-21
:PULSe:TIMing:DOUBle? 5-23
:PULSe:TIMing:DOUBle:MODE 5-23
:PULSe:TIMing:DOUBle:MODE? 5-23
:PULSe:TIMing:DutyCYCle 5-24
:PULSe:TIMing:DutyCYcle? 5-24
:PULSe:TIMing:DutyCYCle:MODE 5-24
:PULSe:TIMing:DutyCYCle:MODE? 5-25

:PULSe:TIMing:PERiod 5-25
:PULSe:TIMing:PERiod? 5-26
:PULSe:TIMing:WIDTh 5-26
:PULSe:TIMing:WIDTh? 5-27
:SYSTem 5-28
:SYSTem:DERRor? 5-28
:SYSTem:ERRor? 5-28
:SYSTem:SET 5-29
:SYSTem:SET? 5-29

6. Remote Programming Examples
Example 1 - Checking Communication 6-2
Example 2 - Status Registers and Queues 6-3
Example 3 - A Sample Session 6-6

A. Specifications
Specifications A-1
Timing Parameters A-1
Common Specifications A-1
Measurement conditions A-1
Period A-1
Width A-1
Delay A-1
Double Pulse A-2
Transition Times A-2
Output Parameters A-3
Pulse Performance A-3
Operating Characteristics A-3
Timing Parameters A-3
Duty Cycle A-3
Inputs and Outputs A-4
Main Outputs A-4
External Input A-4
Trigger Output A-4
Trigger Modes A-5
Additional Features A-5
Remote Control A-5
HP-IB Capabilities A-6
HP-IB Interface Function Code: A-6
HP-MSIB Capabilities A-6
General A-6
Environmental A-6
Ordering Information A-6

B. Errors

Index

Figures

5-1. Leading Edge/Trailing Edge Transition Time Relationships 5-12
5-2. Output Voltage Window 5-15

Tables

2-1. Recommended Test Equipment 2-1
3-1. HP-IB Capabilities 3-2
4-1. Common Command Summary 4-1
4-2. Reset State (Standard Setting) 4-7
5-1. :DIAGnostic Command List 5-1
5-2. :INPut Command List 5-1
5-3. :OUTPut Command List 5-1
5-4. :PULSe Command List 5-2
5-5. :SYSTem Command List 5-2
5-6. Maximum Values of WIDTH, DCYC and DELAY versus PERIOD 5-21
5-7. Maximum Values of DOUB versus PERIOD 5-22
A-1. Maximum Width and Delay Values versus Pulse Period A-1
A-2. Maximum Double Pulse Values versus Pulse Period A-2

1

Installation

The HP 70330A 300MHz Pulse Generator is a 5/8-width module that must be installed into an HP 70001A mainframe.

This chapter describes installing the module into the mainframe, how to change the HP-IB/HP-MSIB address, and possible causes of failure.

Tools needed

The only tool required for the basic installation of the HP 70330A is the 8mm hex-ball driver (HP P/N 8710-1651) that was supplied with your mainframe.

Installation overview

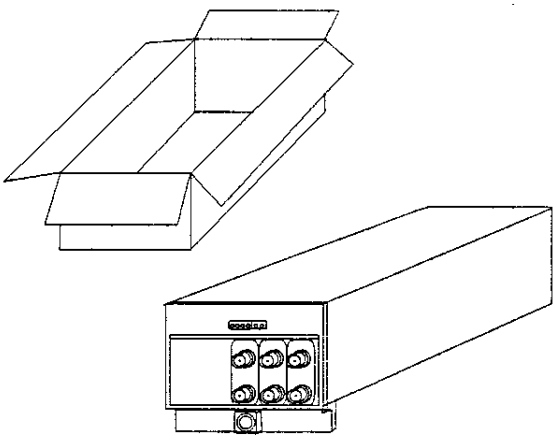
The HP 70330A 300MHz Pulse Generator installation is straightforward and requires minimal time to complete.

Note

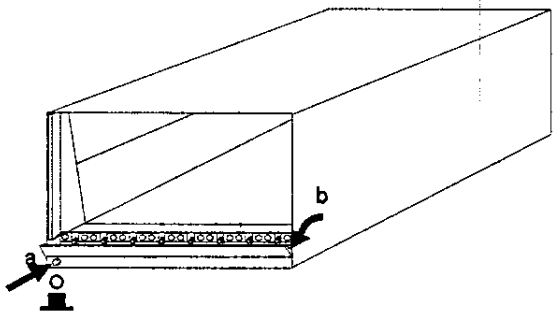
If you want to change the HP-IB/HP-MSIB address of the module, you should do this before installing it (see "Optional: Changing the HP-IB/HP-MSIB Address of the HP 70330A Pulse Generator Module"). It is possible to change the address after installation through the rear of the mainframe, if this is accessible.

Installing the HP 70330A 300MHz Pulse Generator

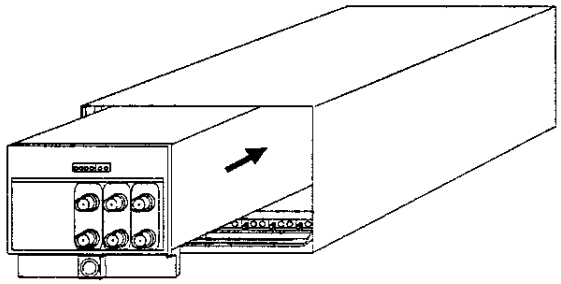
1 Unpack the HP 70330A 300MHz Pulse Generator from its shipping containers. Inspect the HP 70330A thoroughly to ensure that it was not damaged during shipment.



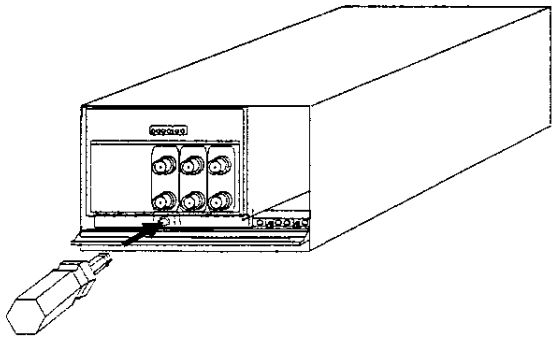
2 Make sure the mainframe is off (powered down) and that the front panel door is open.



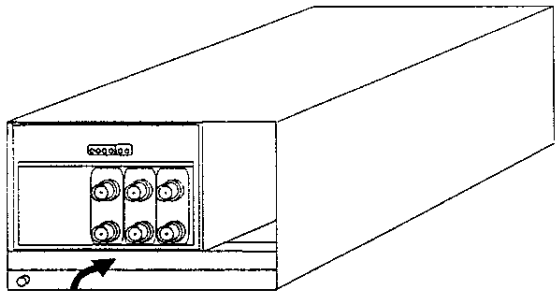
3 Slide the pulse generator into the mainframe.



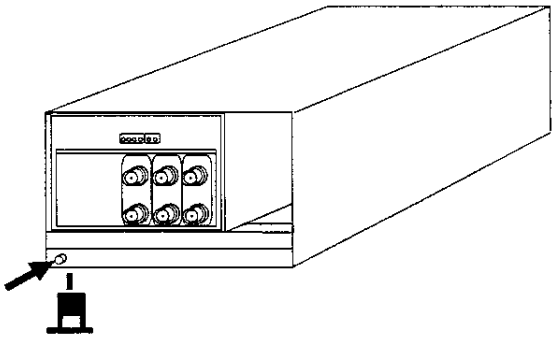
4 Secure the pulse generator by pressing against its front panel while tightening the hex nut latch with the hex-ball driver.



5 Close the front panel door.



6 Press the mainframe's front-panel **LINE** buton to switch on the system.

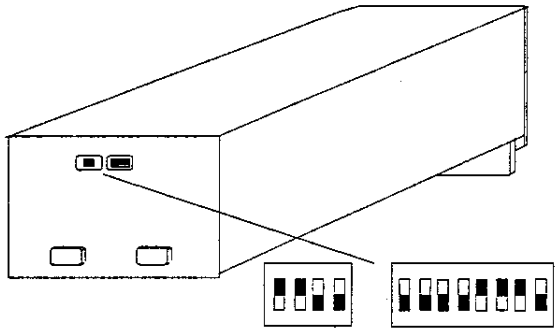


Optional: Changing the HP-IB/HP-MSIB Address of the HP 70330A Pulse Generator Module

1 Locate the eight HP-

MSIB address switches at the rear of the module. These switches are factory preset to row address 0 and column address 14.

The four switches to the left of the address switches set the operation of the instrument. Do not change this switch setting.



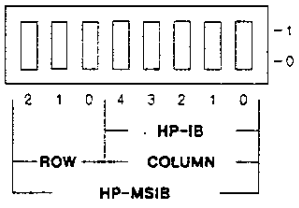
Factory Preset
(Black = Selected)

Each element must have a unique 8-bit binary HP-MSIB address. You can set this address using the 8 address switches. The three most significant bits (MSB) decide the row address. Only the modules and the stand-alone instruments that have addresses in row 0 can communicate over the HP-IB bus. The five least significant bits (LSB) decide the column address. This manual refers to the decimal equivalent of the binary address.

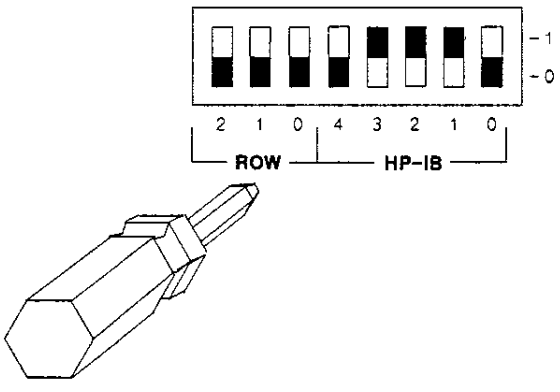
Address 0, 31 is an illegal address. No elements may be set to this address.

You should only set this module to addresses in Row 0 (because only devices in Row 0 can be addressed over the HP-IB).

A thin, nonconductive stylus, such as a toothpick or similar object, is useful for setting the address switches.

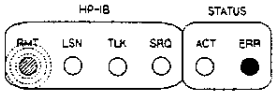


2 Set the address switches of the HP 70330A module to an HP-MSIB address in row 0 that does not have a module assigned to it.

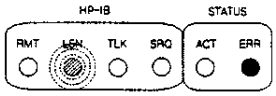


Possible Problems after Switching On

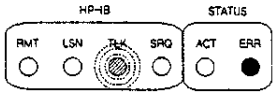
If any of the following errors occur, contact your Hewlett-Packard representative, or the nearest Hewlett-Packard Sales and Support Office.



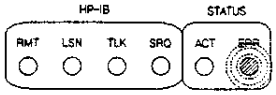
ERR LED on
RMT LED blinking — ROM error



ERR LED on
LSN LED blinking — RAM error



ERR LED on
TLK LED blinking — Internal hardware configuration error



ERR LED blinking — MSIB error

2

Operation Verification Testing

Introduction

This chapter describes the pulse generator operation verification tests. To verify the performance of the pulse generator, use the performance tests in the HP 70330A Service Manual (HP P/N 70330-90001).

Recommended Test Equipment

The external test equipment and accessories required to perform the operation verification tests are listed in Table 2-1. If the recommended test equipment is not available, substitutions may be used if they meet the critical specifications listed in the table.

A controller is necessary to perform the operation verification tests but is not listed in the table.

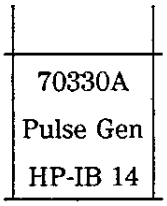
Table 2-1. Recommended Test Equipment

Equipment Name	Critical Specifications	Recommended Equipment
Mainframe with five free slots	Modular Measurement System	HP 70001A
Color Display\Mainframe	Modular Measurement System	HP 70004A
Cables	HP-MSIB (2 needed)	HP 70800A/B/C/D/E
Oscilloscope	Bandwidth ≥500 MHz	HP 54503A Oscilloscope
Cable	50Ω coax BNC(m) connectors	HP 10502A/HP 10503A

Step 1. Check the HP-MSIB/HP-IB

1. Connect the HP 70001A Mainframe containing the module to the HP 70004A Color Display/Mainframe using the two HP-MSIB Cables.
2. Switch on both mainframes.
3. On the display, press the (Display) key under the screen, and then the `Address Map` softkey to the left of the screen.
4. Rotate the knob under the screen to find the entry for the module.

   For example, if you use the factory preset address, the entry is in Row 0, Column 14, and looks like this:



5. When you have selected this box (the frame of a box is highlighted when it is selected), the ACT LED on the front panel of the module should light.

Step 2 Check the Pulse Generation Circuitry

Note A short program for this test is listed after the test description.

Alternatively, use the program given as programming example 2.

1. Connect the HP 70001A Mainframe containing the pulse generator to the controller.
2. Connect the Channel 1 OUTPUT to the oscilloscope.
3. Switch on the pulse generator, the controller and the oscilloscope.
4. Set the oscilloscope timebase to 500μs/DIV, and the ΔV to 0.2V/DIV. Set the trigger to Automatic.
5. Make sure the status bytes are clear by sending the command `*CLS`.
6. Recall the standard setting by sending the `*RST` command.
7. Enable the output by sending the `:OUTP:PULS:STAT ON`
8. Check that the oscilloscope displays a square wave with period 1.00ms, an amplitude of 1V and an offset of 0V.
9. Connect the Channel 1 $\overline{\text{OUTPUT}}$ to the oscilloscope.
10. Enable the output by sending the `:OUTP:PULS:CST ON`
11. Check that the oscilloscope displays a square wave with period 1.00ms, an amplitude of 1V and an offset of 0V.

12. Connect the Channel 2 OUTPUT to the oscilloscope.
13. Enable the output by sending the `:OUTP2:PULS:STAT ON`
14. Check that the oscilloscope displays a square wave with period 1.00ms, an amplitude of 1V and an offset of 0V.
15. Connect the Channel 2 $\overline{\text{OUTPUT}}$ to the oscilloscope.
16. Enable the output by sending the `:OUTP2:PULS:CST ON`
17. Check that the oscilloscope displays a square wave with period 1.00ms, an amplitude of 1V and an offset of 0V.

The Operation Verification Program Listing

```
10   !---------------------------------------
20   !
30   ! HP 70330A Operation Verification Program
40   !
50   !---------------------------------
60   !
70   ! Declarations and Initializations
80   !
90   Pg=714
100  !
110  ! Step 1
120  !
130  CLEAR SCREEN
140  PRINT TABXY(1,1);"Check The HP-MSIB/HP-IB (press ENTER to continue)"
150  INPUT A$
160  !
170  ! Clear the status bytes and restore the standard setting
180  !
190  OUTPUT Pg;"*cls;*rst"
200  !
210  PRINT TABXY(1,3);"Set the 'scope to 500us/DIV and 0.2V/DIV, automatic trigger"
220  !
230  ! Check the channel 1 output
240  !
250  PRINT TABXY(1,5);"Connect the 'scope to the channel 1 output and then press ENTER"
260  INPUT A$
270  OUTPUT Pg;":outp:puls:stat on"
280  PRINT TABXY(3,6);"Check that you have a square wave with period 1.00ms, amp. 1V and offset 0V"
290  !
300  ! Check the channel 1 inverted output
310  !
320  PRINT TABXY(1,8);"Connect the 'scope to the channel 1 inverted output and then press ENTER"
330  INPUT A$
340  OUTPUT Pg;":outp:puls:stat off;cst on"
350  PRINT TABXY(3,9);"Check that you have a square wave with period 1.00ms, amp. 1V and offset 0V"
360  !
370  ! Check the channel 2 output
380  !
390  PRINT TABXY(1,11);"Connect the 'scope to the channel 2 output and then press ENTER"
400  INPUT A$
410  OUTPUT Pg;":outp:puls:cst off"
420  OUTPUT Pg;":outp2:puls:stat on"
430  PRINT TABXY(3,12);"Check that you have a square wave with period 1.00ms, amp. 1V and offset 0V"
440  !
450  ! Check the channel 2 inverted output
460  !
470  PRINT TABXY(1,14);"Connect the 'scope to the channel 2 inverted output and then press ENTER"
480  INPUT A$
490  OUTPUT Pg;":outp2:puls:stat off;cst on"
```

```
500   PRINT TABXY(3,15);"Check that you have a square wave with period 1.00ms, amp. 1V and offset 0V"
510   !
520   !And Finish
530   !
540   PRINT TABXY(4,17);"Press ENTER to continue"
550   INPUT A$
560   OUTPUT Pg;":outp2:puls:cst off"
570   END
```

3

Programming the HP 70330A

Introduction

This chapter gives general information on how to control the HP 70330A remotely. Descriptions for the actual commands for the HP 70330A are in the following chapters. The information in these chapters is specific to the HP 70330A, and assumes that you are already familiar with using the HP-IB or the HP-MSIB.

A Brief Overview of the Instrument

The HP 70330A is a dual channel pulse generator offering 300MHz timing with variable transition times and variable widths and delays. Into 50Ω, the output amplitude can be varied up to 5V, within a window of ±5V. An external trigger input enables the generation of counted or gated burst, dc, single or double pulses. The external trigger input can also be used to recover distorted pulses.

Further details can be found in Appendix A.

Interface Types

There are two communications interfaces used in the HP 70330A Series. The HP-MSIB (Measurement System Interface Bus) and the HP-IB (Hewlett-Packard Interface Bus).

HP-MSIB The Measurement System Interface Bus is the interface used for internal communication between system modules on the Modular Measurement System (MMS).

Operational details of the HP-MSIB interface can be ignored when you are controlling the HP 70330A remotely.

HP-IB The Hewlett-Packard Interface Bus is the interface used for communication between a controller and an external device such as the HP 70330A. The HP-IB conforms to IEEE standard 488-1978, ANSII standard MC 1.1 and IEC recommendation 625-1.

The HP 70330A HP-IB Capabilities

If you are not familiar with the HP-IB, then refer to the following books:

- HP publication 5952-0156, *Tutorial Description of HP-IB.*
- ANSI/IEEE-488.1-1978, *IEEE Standard Digital Interface for Programmable Instrumentation,* and ANSI/IEEE-488.2-1987, *IEEE Standard Codes, Formats, and Common Commands,* published by the Institute of Electrical and Electronic Engineers.

The HP 70330A interfaces to the HP-IB as defined by the IEEE Standards 488.1 and 488.2. The table shows the interface functional subset that the HP 70330A implements.

Table 3-1. HP-IB Capabilities

Mnemonic	Function
SH1	Complete source handshake capability
AH1	Complete acceptor handshake capability
T6	Basic talker; serial poll; unaddressed to talk if addressed to listen
TE0	Basic extended talker; serial poll; unaddressed to talk if addressed as secondary in primary listener state
L4	Basic listener; unaddressed to listen if addressed to talk; no listen only
SR1	Complete service request capability
RL1	Complete remote/local capability
PP0	No parallel poll capability
DC1	Device clear capability
DT1	Device trigger capability (accepted but ignored)
C0	No controller capability

Remote Display Indicators

There are four indicators that you may see when you are controlling the HP 70330A remotely.

RMT lights when the HP 70330A is operating over the HP-IB.

LSN lights when the HP 70330A is addressed as a listener.

TLK lights when the HP 70330A is addressed as a talker.

SRQ (service request) lights when the HP 70330A has requested service.

Command Messages

A program message is a message from the controller to the HP 70330A. The following are a few points about program messages:

- Either upper-case or lower-case characters can be used.
- Common commands are defined by the IEEE 488.2 standard and begin with an asterisk (*).
- Other commands are defined by the SCPI standard. These commands are built of one or more command nodes, separated by colons (:). These commands start with a colon.
- Several commands can be sent in a single message. Each command must be separated from the next one by a semicolon, (;).
- A program message is ended by a line feed (LF) character, or any character sent with End-Or-Identify (EOI).
- Any valid number/unit combination can be used.

  Example 10km, 10000m and 1.0e4m are all equivalent.

  If you do not specify a unit, then the default unit is assumed. The default unit for a command is with the reference to the command.

Message Exchange

The HP 70330A exchanges messages using an input and an output queue. Error messages are kept in a separate error queue.

The Input Queue

The input queue is a FIFO queue (first-in first-out). Incoming bytes are stored in the input queue as follows:

- Receiving a byte:
  - Clears the output queue.
  - Clears Bit 7 (MSB).
- No modification is made inside strings or binary blocks.

  Outside strings and binary blocks, the following modifications are made:
  - Lower-case characters are converted to upper-case.
  - The characters 00_{16} to 09_{16} and $0B_{16}$ to $1F_{16}$ are converted to spaces (20_{16}).
  - Two or more blanks are truncated to one.
- An EOI (End Or Identify) sent with any character is put into the input queue as the character followed by a line feed (LF, $0A_{16}$). If EOI is sent with a LF, only one LF is put into the input queue.
- The parser starts if the LF character is received or if the input queue is full.

Clearing the Input Queue

Switching the power off causes commands that are in the input queue, but have not been executed to be lost.

The Output Queue

The output queue contains responses to query messages. The HP 70330A transmits any data from the output queue when a controller addresses the instrument as a talker.

Each response message ends with a carriage return (CR, $0D_{16}$) and a LF ($0A_{16}$), with EOI = TRUE. If no query is received, or if the query has an error, the output queue remains empty.

The message available, Message Available (MAV), bit is set in bit four of the Status Byte register whenever there is data in the output queue.

The Error Queue

The error queue is 10 errors long. It is a FIFO queue (first-in first-out). That is, the first error read is the oldest error to have occurred. A new error is only put into the queue if it is not already in it.

If more than 16 errors are put into the queue, the message '777 <Queue Overflow>' is placed as the last message in the queue.

Syntax Diagram Conventions

Short Form and Long Form

The instrument accepts messages in short or long forms. For example, the message `:INPUT:TRIGGER:STATE ON` is in long form, the short form of this message is `:INP:TRIG:STAT ON`.

In this manual the messages are written in a combination of upper and lower case. The upper case characters are used for the short form parts of the message. For example, the above command would be written `:INPut|TRIGger:STATe ON`.

Command and Query Syntax

command| All characters not between angled brackets must be sent exactly as shown.

The characters between angled brackets (< ... >) show the kind of data that you require, or that you get in a response. You do not type the angled brackets in the actual message. Descriptions of these items follow the syntax description. The more common of these are listed below.

string is ascii data. A string is contained between a " at the start and the end, or a ' at the start and the end.

value is numeric data in integer (12), decimal (85.5) or exponential format (99.9E-9).

wsp is a white space.

Other kinds of data are described as required.

The characters between square brackets ([...]) show optional information that you can include with the message.

The bar (|) shows an either-or choice of data, for example, *a*|*b* means either *a* or *b*, but not both simultaneously.

Extra spaces are ignored; they can be inserted to improve readability.

4

Remote Commands, Part I: Common Commands

Table 4-1. Common Command Summary

Command	Function
*CLS	Clear Status
*ESE	Standard Event Status Enable
*ESE?	Standard Event Status Enable Query
*ESR?	Standard Event Status Register Query
*IDN?	Identification Query
*LRN?	Learn Device Setup Query
*OPC	Operation Complete
*OPC?	Operation Complete Query
*RCL	Recall
*RST	Reset
*SAV	Save
*SRE	Service Request Enable
*SRE?	Service Request Enable Query
*STB?	Read Status Byte Query
*TRG	Trigger
*TST?	Self Test Query
*WAI	Wait to Continue

*CLS

Clear status command.

Syntax *CLS

Definition The *CLS command clears the following:

- Error queue
- Standard event status register (ESR)
- Status byte register (STB)

After the *CLS command the instrument is left in the idle state. The instrument setting is unaltered by the command, though *OPC/*OPC? actions are canceled.

If the *CLS command occurs directly after a program message terminator, the output queue and MAV, bit 4, in the status byte register are cleared, and if condition bits 2-0 of the status byte register are zero, MSS, bit 6 of the status byte register is also zero.

Example

```
OUTPUT 714;"*CLS"
```

---

*ESE

Standard event status enable command.

Syntax

*ESE <wsp><value>
0 ≤ value ≤ 255

Definition

The *ESE command sets bits in the standard event status enable register (ESE) that enable the corresponding bits in the standard event status register (ESR).

The register is cleared:

- At power-on
- By sending a value of zero

The register is not changed by the *RST and *CLS commands.

BIT	MNEMONIC	BIT VALUE
7	PON	128
6	Not used	0
5	CME	32
4	EXE	16
3	DDE	8
2	QYE	4
1	Not used	0
0	OPC	1

The Event Status Enable Register

Example

```
OUTPUT 714;"*ESE 21"
```

*ESE?

Standard event status enable query.

Syntax *ESE?

Definition The standard event status enable query returns the contents of the standard event status enable register.

$0 \leq$ contents ≤ 255

Example
```
OUTPUT 714;"*ESE?"
ENTER 714; A$
```

*ESR?

Standard event status register query.

Syntax *ESR?

Definition The standard event status register query returns the contents of the standard event status register. The register is cleared after being read.

$0 \leq$ contents ≤ 255

BITS	MNEMONICS	BIT VALUE
7	PON	128
6	Not used	0
5	CME	32
4	EXE	16
3	DDE	8
2	QYE	4
1	Not used	0
0	OPC	1

The Standard Event Status Register

Example
```
OUTPUT 714;"*ESR?"
ENTER 714; A$
```

*IDN?

Identification query.

Syntax `*IDN?`

Definition The identification query commands the instrument to identify itself over the interface.

Response: `HEWLETT-PACKARD, 70330A, 0, n.n`

`HEWLETT-PACKARD`: manufacturer

`70330A`: instrument model number

`0`: indicates serial numbers are not provided.

`n.n`: firmware revision level

Example

```
DIM A$ [100]
OUTPUT 714;"*IDN?"
ENTER 714; A$
```

*LRN?

Learn device setting query.

Syntax `*LRN?`

Definition The learn query returns the whole instrument setup and stores as a binary block of data.

The response message can be transmitted again as a program message without requiring any alterations.

The returned setup has the same form as described for `:SYST:SET?` query.

Related Command `:SYSTem:SET`

Example

```
DIM A$ [1000]
OUTPUT 714;"*LRN?"
ENTER 714; A$
```

*OPC

Operation complete command.

Syntax *OPC

Definition The instrument parses all program message units in the input queue and sets the operation complete bit in the standard event status register (ESR). There is a delay of approximately 0.1 seconds between interpreting the command and setting the bit.

Example
```
OUTPUT 714;"*CLS;*ESE 1;*SRE 32"
OUTPUT 714;"*OPC"
```

*OPC?

Operation complete query.

Syntax *OPC?

Definition The instrument parses all program message units in the input queue and places an ASCII '1' in the output queue. There is a delay of approximately 0.1 seconds between interpreting the command and puting the '1' in the queue.

Example
```
OUTPUT 714;"*OPC?"
ENTER 714;A$
```

*RCL

Recall command

Syntax *RCL <wsp> <location>

$1 \leq$ location ≤ 19

Definition An instrument setting from the internal RAM is made the current instrument setting.

You can recall user settings from locations 1-19. See "*SAV".

Example
```
OUTPUT 714;"*RCL 3"
```

*RST

Reset command.

Syntax *RST

Definition The reset setting (standard setting) stored in ROM is made the instrument setting.

Instrument state: the instrument is placed in the idle state awaiting a command.

The following are not changed:

- HP-IB (interface) state
- Instrument interface address
- Output queue
- Service request enable register (SRE)
- Standard event status enable register (ESE)

The commands and parameters of the reset state are listed in the following table.

Table 4-2. Reset State (Standard Setting)

Commands			Parameters (Default)
:INPut			
	:TRIGger		
		:STATe	OFF
		:MODE	AUTO
		:SLOPe	POS
		:THREshold	0.0V
:PULSe			
	:COUNt		1
	:EDGE		
		:LEADing	1.0E-6
		:TRAILing	1.0E-6
	:TIMing		
		:PERiod	1.00ms
		:WIDTh	100μs
		:DELay	0.00ps
		:DOUBle	200μs
		:MODE	OFF
		:DutyCYCle	50 %
		:MODE	OFF
	:LEVel		
		:HIGH	+0.50V
		:LOW	-0.50V
		:AMPLitude	1.00V
		:OFFSet	0.00V
		:LIMit	OFF
:OUTPut			
	:PULSe		
		:POLarity	NORM
		:STATe	OFF
		:CSTate	OFF

Example `OUTPUT 714;"*RST"`

*SAV

Save command.

Syntax *SAV <wsp> <location>

1 ≤ location ≤ 19

Definition The instrument setting is stored in RAM. You can store settings in locations 1-19. The scope of the saved setting is identical with the scope of the standard setting described in "*RST".

Example `OUTPUT 714;"*SAV 3"`

*SRE

Service request enable register.

Syntax *SRE <wsp> <value>

0 ≤ value ≤ 255

Definition The service request enable command sets bits in the service request enable register that enable the corresponding status byte register bits.

The register is cleared:

- At power-on
- By sending a value of zero.

The register is not changed by the *RST and *CLS commands.

BITS	MNEMONICS	BIT VALUE
7	Not used	0
6	RQS/MSS	64
5	ESB	32
4	MAV	16
3	Not used	0
2	Not used	0
1	Not used	0
0	Not used	0

The Service Request Enable Register

Example
```
OUTPUT 714;"*SRE 48"
```

*SRE?

Service request enable query.

Syntax *SRE?

Definition The service request enable query returns the contents of the service request enable register.

0 ≤ contents ≤ 255

Example
```
OUTPUT 714;"*SRE?"
ENTER 714; A$
```

*STB?

Read status byte query.

Syntax *STB?

Definition The read status byte query returns the contents of the status byte register.

$0 \leq$ contents ≤ 255

The MSS message is reported in bit six of the status byte register.

BITS	MNEMONICS	BIT VALUE
7	Not used	0
6	MSS	64
5	ESB	32
4	MAV	16
3	Not used	0
2	Not used	0
1	Not used	0
0	Not used	0

The Status Byte Register

Example
```
OUTPUT 714;"*STB?"
ENTER 714; A$
```

*TRG

Trigger command.

Syntax *TRG

Definition The trigger command has the same effect as a Group Execute Trigger (GET). If you have selected trigger or burst operating mode, a trigger event occurs, and one pulse, double-pulse, or burst is generated.

:INPut:TRIGger:STATe ON changes to :STATE OFF when *TRG processed.

Example
```
OUTPUT 714;"*TRG"
```

*TST?

Self-test query.

Syntax `*TST?`

Definition The self-test query commands the instrument to perform a self-test and place the results of the test in the output queue.

Returned value: $0 \leq$ value ≤ 657.

A value of zero indicates no errors.

When the self-test fails, the results are also put in the error queue. It is recommended that you read self-test results from the error queue. Explanations for the non-zero results of the self-test are given in Appendix B.

No further commands are allowed while the test is running.

The instrument is returned to the setting that was active at the time the self-test query was processed.

The self-test does not require operator interaction beyond sending the *TST? query.

Example

```
OUTPUT 714;"*TST?"
ENTER 714; A$
```

*WAI

Wait to continue command.

Syntax `*WAI`

Definition The wait-to-continue command prevents the instrument from executing any further commands, all pending operations are completed. There is a delay of approximately 0.1 seconds between the completion of the final operation and the parsing of the next command.

Example

```
OUTPUT 714;"*WAI"
```

5

Remote Commands, Part II: Device Commands

Table 5-1. :DIAGnostic Command List

Command	Parameter	Min	Max	Default
:DIAGnostic				
:BOOT				
[:SOURce]	LOADer|INSTrument			INST
[:SOURce]?				

Table 5-2. :INPut Command List

Command	Parameter	Min	Max	Default
:INPut				
:TRIGger				
:MODE	AUTO|TRIGger|GATE| BURSt|EWIDth			AUTO
:MODE?				
:SLOPe	POSitive|NEGative			POS
:SLOPe?				
:STATe	OFF|ON|0|1			OFF
:STATe?				
:THReshold	<value>|MIN|MAX	-5.0V	5.0V	0.0V
:THReshold?				

Table 5-3. :OUTPut Command List

Command	Parameter	Min	Max	Default
:OUTPut[1|2]				
:PULSe				
:CSTate	OFF|ON|0|1			OFF
:CSTate?				
:POLarity	NORMal|COMPlement			NORM
:POLarity?				
:STATe	OFF|ON|0|1			OFF
:STATe?				

Table 5-4. :PULSe Command List

Command	Parameter	Min	Max	Default
:PULSe				
:COUNt	<value>|MIN|MAX	1	9999	1
:COUNt?				
:EDGE				
:LEADing	<value>|MIN|MAX	1ns	100μs	1.00μs
:LEADing?				
:TRAiLing	<value>|MIN|MAX	1ns	100μs	1.00μs
:TRAiLing?				
:LEVel				
:AMPLitude	<value>|MIN|MAX	100mV_{p-p}	5.00V_{p-p}	1.00V_{p-p}
:AMPLitude?				
:HIGH	<value>|MIN|MAX	-4.90V	5.00V	0.50V
:HIGH?				
:LIMit	OFF|ON|0|1			OFF
:AMPLitude?				
:HIGH?				
:LOW?				
:OFFSet?				
:LIMit?				
:LOW	<value>|MIN|MAX	-5.00V	4.90V	-0.50V
:LOW?				
:OFFSet	<value>|MIN|MAX	-4.95V	4.95V	0.00V
:OFFSet?				
:TIMing				
:DELay	<value>|MIN|MAX	0.00ns	90.0ms	0.00ps
:DELay?				
:DOUBle	<value>|MIN|MAX	3.33ns	90.0ms	200μs
:MODE	OFF|ON|0|1			OFF
:MODE?				
:DOUBle?				
:DutyCYCle	<value>|MIN|MAX	1%	90%	50%
:MODE	OFF|ON|0|1			OFF
:MODE?				
:DutyCYCle?				
:PERiod	<value>|MIN|MAX	3.33ns	99.9ms	1.00ms
:PERiod?				
:WIDTh	<value>|MIN|MAX	1.50ns	90.0ms	1.00μs
:WIDTh?				

Table 5-5. :SYSTem Command List

Command	Parameter	Min	Max	Default
:SYSTem				
:DERRor?	[NUMeric|STRing|SCPI]			SCPI
:ERRor?	[NUMeric|STRing|SCPI]			SCPI
:SET	<data>			
:SET?				

:DIAGnostic

:DIAGnostic:BOOT:SOURce

Syntax `:DIAGnostic:BOOT[:SOURce]<wsp>LOADer|INSTrument`

Description This command selects the next switch-on boot mode of the instrument.

`:DIAGnostic:BOOT:SOURce INSTrument` (Default)
This command selects the instrument switch-on boot mode.

`:DIAGnostic:BOOT:SOURCE LOADer`
This command selects the loader switch-on boot mode. The instrument will start up the firmware load application after the next power cycle.

The firmware loader is a special piece of software that loads new instrument firmware revisions into the instrument via the HP-IB or HP-MSIB. For more details about the firmware loader see the HP 70330A Service Manual (HP P/N 70330-90001).

Example
```
OUTPUT 714;":DIAG:BOOT:SOUR INST"
```

:DIAGnostic:BOOT:SOURce?

Syntax `:DIAGnostic:BOOT[:SOURce]?`

Description This command returns the next switch-on boot mode of the instrument. The response is one of `INSTRUMENT`, if normal instrument operation will be started after the next power cycle, or `LOADER`, if the firmware loader will be started after the next power cycle.

Example
```
OUTPUT 714;":DIAG:BOOT:SOUR?"
ENTER 714;A$
```

:INPut:TRIGger:MODE

Syntax :INPut:TRIGger:MODE<wsp>AUTO|TRIGger|GATE|BURSt|EWIDth

Description

:INPut:TRIGger:MODE AUTO (Default)
This command places the instrument into automatic operating mode. A continuous waveform (free run mode) is generated. The external input is disabled.

:INPut:TRIGger:MODE TRIGger
This command places the instrument into trigger operating mode. One pulse or double pulse signal is generated per trigger signal. A trigger signal comes from the external input or from the *TRG command.

Note :INPut:TRIGger:MODE TRIGger and :PULse:TIMing:DutyCYCle:MODE ON are incompatible.

:INPut:TRIGger:MODE GATE
This command places the instrument into gate operating mode. Pulses or double pulses are generated for the duration of the gate.

:INPut:TRIGger:MODE BURSt
This command places the instrument into burst operating mode. A specified number of pulses or double pulses are generated for each trigger signal. A trigger signal comes from the external input or from the *TRG command.

Note If :PULSe:TIMing:PERiod < 5.00ns then :INPut:TRIGger:MODE BURSt is not allowed.

Related command: :PULSe:COUNt <value>|MIN|MAX

:INPut:TRIGger:MODE EWIDth
This command places the instrument into external width operating mode. Pulse width and period are controlled by a signal applied at the EXT INPUT.

All modes are common to channels 1 and 2 and cannot be set individually for each channel.

Example

```
OUTPUT 714;":INP:TRIG:MODE AUTO"
```

:INPut:TRIGger:MODE?

Syntax `:INPut:TRIGger:MODE?`

Description This command returns the current operating mode. The response is one of `AUTO`, `TRIGGER`, `GATE`, `BURST`, or `EWIDTH`

Example
```
OUTPUT 714;":INP:TRIG:MODE?"
ENTER 714;A$
```

:INPut:TRIGger:SLOPe

Syntax `:INPut:TRIGger:SLOPe<wsp>POSitive|NEGative`

Description Use this command to set whether the instrument is triggered on the positive or negative slope of the input trigger signal.

`:INPut:TRIGger:SLOPe POSitive` (Default)
This command sets the instrument so that it is triggered on the positive edge of a pulse at the external input.

`:INPut:TRIGger:SLOPe NEGative`
This command sets the instrument so that it is triggered on the negative edge of a pulse at the external input.

Example
```
OUTPUT 714;":INP:TRIG:SLOP POS"
```

:INPut:TRIGger:SLOPe?

Syntax `:INPut:TRIGger:SLOPe?`

Description This command returns the edge of the pulse at the external input that is used to trigger the instrument. The response is either `POSITIVE` or `NEGATIVE`.

Example
```
OUTPUT 714;":INP:TRIG:SLOP?"
ENTER 714; A$
```

:INPut:TRIGger:STATe

Syntax :INPut:TRIGger:STATe<wsp>ON|OFF|1|0

Description This command enables or disables triggering from the EXT INPUT connector.

:INPut:TRIGger:STATe OFF or :INPut:TRIGger:STATe 0 (Default)
This command disables the external input. When this input is disabled trigger signals received by the instrument have no effect.

:INPut:TRIGger:STATe ON or :INPut:TRIGger:STATe 1
This command enables the external input. When this input is enabled trigger signals received by the instrument are effective.

The state is common to channels 1 and 2, it cannot be set individually for each channel.

If *TRG is processed, :STATE ON changes to :STATE OFF.

Example

```
OUTPUT 714;":INP:TRIG:STAT ON"
```

:INPut:TRIGger:STATe?

Syntax :INPut:TRIGger:STATe

Description This command returns the current state of the input trigger. The response is either OFF or ON. OFF indicates that the input trigger is disabled, ON indicates that the input trigger is enabled.

Example

```
OUTPUT 714;":INP:TRIG:STAT?"
ENTER 714;A$
```

:INPut:TRIGger:THReshold

Syntax :INPut:TRIGger:THReshold<wsp><value>|MIN|MAX

-5.0 ≤ <value> ≤ 5.0 or MIN or MAX
Specified Range: -5V to 5V
Resolution: 0.1V
Default: 0.0V

Description This command sets the threshold level of the input trigger signal at which the trigger is generated. The value can be set between -5.0 and +5.0 volts. Specify the value in volts in an integer or decimal format, with or without the units (V). The default units (that is, the units used when you do not enter units) are volts. Selecting MIN automatically sets the threshold to -5.0V, selecting MAX automatically sets the threshold to 5.0V.

Example

```
OUTPUT 714;":INP:TRIG:THR 3.5V"
```

:INPut:TRIGger:THReshold?

Syntax `:INPut:TRIGger:THReshold`

Description This command returns the current setting of the threshold value. The response is in decimal form, for example: `3.5`. The units of the returned value are volts.

It is also possible to find the minimum or maximum possible threshold value of the instrument by the commands `THR MIN` and `THR MAX` respectively. These return the values `-5.0` for the minimum and `5.0` for the maximum.

Example

```
OUTPUT 714;":INP:TRIG:THR?"
ENTER 714;A$
```

:OUTPut

The OUTPut commands are channel specific and can be applied to either channel. You specify the channel to which you direct the command by placing the channel number immediately after the OUTPut. This can be done in three ways.

OUTPut	(Default) When you do not specify channel the command is directed to CHANNEL 1.
OUTPut1	The command is directed to CHANNEL 1.
OUTPut2	The command is directed to CHANNEL 2.

:OUTPut:PULSe:CSTate

Syntax :OUTPut[1|2]:PULSe:CSTate<wsp>OFF|ON|0|1

Description This command enables or disables the $\overline{\text{OUTPUT}}$.

:OUTPut:PULSe:CSTate OFF or :OUTPut:PULSe:CSTate 0 (Default)
The $\overline{\text{OUTPUT}}$ is disabled.

:OUTPut:PULSe:CSTate ON or :OUTPut:PULSe:CSTate 1
The specified channel's $\overline{\text{OUTPUT}}$ is enabled.

$\overline{\text{OUTPUT}}$ is disabled at switch-on and after a reset.

Example

```
OUTPUT 714;":OUTP1:PULS:CSTate ON"
```

:OUTPut:PULSe:CSTate?

Syntax :OUTPut[1|2]:PULSe:CSTate?

Description This command asks about the current state of the $\overline{\text{OUTPUT}}$. The response is either OFF, indicating that the $\overline{\text{OUTPUT}}$ is disabled, or ON, indicating that the $\overline{\text{OUTPUT}}$ is enabled.

Example

```
OUTPUT 714;":OUTP1:PULS:CSTate?"
ENTER 714;A$
```

:OUTPut:PULSe:POLarity

Syntax `:OUTPut[1|2]:PULSe:POLarity<wsp>NORMal|COMPlement`

Description This command sets whether the output from the channel is inverted or not. This command applies to both the OUTPUT and $\overline{\text{OUTPUT}}$ signals simultaneously. In both cases (NORM or COMP) $\overline{\text{OUTPUT}}$ is the inverted form of OUTPUT.

`:OUTPut:PULSe:POLarity NORMal` (Default)
The output signals are output as specified by the other setting parameters.

`:OUTPut:PULSe:POLarity COMPlement`
The output signals are inverted with respect to the other setting parameters.

Example
```
OUTPUT 714;":OUTP1:PULS:POL COMP"
```

:OUTPut:PULSe:POLarity?

Syntax `:OUTPut[1|2]:PULSe:POLarity?`

Description This command returns whether the output signals are as specified by the other setting parameters, or whether the signals are inverted. The response is either `NORM` or `COMP`. When `COMP` is returned it indicates that the outputs are in their inverted forms.

Example
```
OUTPUT 714;":OUTP1:PULS:POL?"
ENTER 714;A$
```

:OUTPut:PULSe:STATe

Syntax `:OUTPut[1|2]:PULSe:STATe<wsp>OFF|ON|0|1`

Description This command enables or disables the OUTPUT.

`:OUTPut:PULSe:STATe OFF` or `:OUTPut:PULSe:STATe 0` (Default)
The OUTPUT is disabled.

`:OUTPut:PULSe:STATe ON` or `:OUTPut:PULSe:STATe 1`
The specified channel's OUTPUT is enabled.

OUTPUT is disabled at switch-on and after a reset.

Example
```
OUTPUT 714;":OUTP1:PULS:STATe ON"
```

:OUTPut:PULSe:STATe?

Syntax `:OUTPut[1|2]:PULSe:STATe?`

Description This command asks about the current state of the OUTPUT. The response is either `OFF`, indicating that the OUTPUT is disabled, or `ON`, indicating that the OUTPUT is enabled.

Example

```
OUTPUT 714;":OUTP1:PULS:STATe?"
ENTER 714;A$
```

:PULSe

The PULSe commands are channel specific and can be applied to either channel. You specify the channel to which you direct the command by placing the channel number immediately after the PULSe. This can be done in three ways.

PULSe	(Default) When you do not specify a channel the command is directed to CHANNEL 1.
PULSe1	The command is directed to CHANNEL 1.
PULSe2	The command is directed to CHANNEL 2.

:PULSe:COUNt

Syntax :PULSe[1|2]:COUNt<wsp><value>|MIN|MAX

1 ≤ <value> ≤ 9999 or MIN or MAX
Specified Range: 1 to 9999
Resolution: 1
Default: 1

Description This command sets the number of pulses emitted for each trigger input in burst operating mode. The number of pulses can be from 1 through 9999. Specify the count as an integer. You can also specify MIN, which sets the pulse count to 1, or MAX, which sets the pulse count to 9999.

The command :INPut:TRIG:MODE BURSt is closely related to this command.

Example

```
OUTPUT 714;":PULS:COUN 1000"
```

:PULSe:COUNt?

Syntax :PULSe[1|2]:COUNt?

Description This command returns the number of pulses contained in a burst. The response is in integer form, for example: 1000.

Example

```
OUTPUT 714;":PULS:COUN?"
ENTER 714;A$
```

:PULSe:EDGE:LEADing

Syntax `:PULSe[1|2]:EDGE:LEADing<wsp><value>|MIN|MAX`

`670.0ps` ≤ <value> ≤ `100.0`μs or `MIN` or `MAX`
Specified Range: 1ns to 0.1ms
Resolution:

Range	Resolution
1.00ns - 9.99ns	0.01ns
10.0ns - 99.9ns	0.1ns
100ns - 999ns	1ns
1.00μs - 9.99μs	0.01μs
10.0μs - 100μs	0.1μs

Default: 1.00μs

Description This command sets the transition time for the leading edge of the pulse to rise from 10% to 90% of its amplitude. The leading edge of the pulse can be programmed from 670ps through 100μs. Enter the value for the transition time in seconds in integer, decimal or exponential form, with or without units. The default units (that is, the units used when you do not enter units) are seconds. You can also specify `MIN`, which sets the transition time to the minimum value of the range in which the other edge is set, or `MAX`, which sets the transition time to the maximum value of the range in which the other edge is set.

The leading and trailing edge transition times are independently programmable, but they must remain within the following, overlapping ranges:

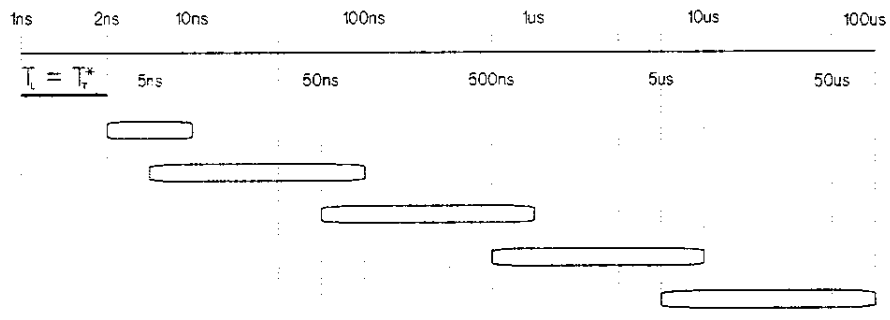


Figure 5-1. Leading Edge/Trailing Edge Transition Time Relationships

* Leading and trailing edges *must* be assigned the same value.

Note

If you are changing from one of these ranges to another, you must change both the leading and trailing edges in the same command. If you do not do this an error occurs and neither edge is changed.

It is possible to set leading and trailing edges, with other timing parameters, so that the full amplitude of the output signal cannot be reached

Example

```
OUTPUT 714;":PULS:EDGE:LEAD 1.23US" or
OUTPUT 714;":PULS:EDGE:LEAD 1.23E-6"
```

```
OUTPUT 714;":PULS:EDGE:LEAD 1.23US;TRAIL 5US"
```

:PULSe:EDGE:LEADing?

Syntax `:PULSe[1|2]:EDGE:LEADing?`

Description This command returns the leading edge transition time of the pulse. The response is the transition time in exponential format, for example `1.23E-3`. The units of the returned value are seconds.

Example
```
OUTPUT 714;"PULS:EDGE:LEAD?"
ENTER 714;A$
```

:PULSe:EDGE:TRAiLing

Syntax `:PULSe[1|2]:EDGE:TRAiLing<wsp><value> |MIN|MAX`

670.0ps ≤ <value> ≤ 100μs or `MIN` or `MAX`
Specified Range: 1ns to 0.1ms
Resolution:

Range	Resolution
1.00ns - 9.99ns	0.01ns
10.0ns - 99.9ns	0.1ns
100ns - 999ns	1ns
1.00μs - 9.99μs	0.01μs
10.0μs - 100μs	0.1μs

Default: 1.00μs

Description This command sets the transition time for the trailing edge of the pulse to fall from 90% to 10% of its amplitude. The trailing edge of the pulse can be programmed from 670ps through 100μs. Enter the value for the transition time in seconds in integer, decimal or exponential format, with or without units. The default units (that is, the units used when you do not enter units) are seconds. You can also specify `MIN`, which sets the transition time to the minimum value of the range in which the other edge is set, or `MAX`, which sets the transition time to the maximum value of the range in which the other edge is set.

Note

The leading and trailing edge transition times are related. See ":PULSe:EDGE:LEADing" for the relationship.

Example
```
OUTPUT 714;":PULS:EDGE:TRA 1.23MS" or
OUTPUT 714;":PULS:EDGE:TRA 1.23E-3"
```

:PULSe:EDGE:TRAiLing?

Syntax `:PULSe[1|2]:EDGE:TRAiLing?`

Description This command returns the trailing edge transition time of the pulse. The response is the transition time in exponential format, for example `1.23E-3`. The units of the returned value are seconds.

Example

```
OUTPUT 714;"PULS:EDGE:TRA?"
ENTER 714;A$
```

:PULSe:LEVel:AMPLitude

Syntax :PULSe"[1|2]:LEVel:AMPLitude<wsp><value>|MIN|MAX

0.1V ≤ <value> ≤ 5.2V or MIN or MAX
Specified Range: $100mV_{p-p}$ to $5.00V_{p-p}$
Resolution: 0.01V
Default: 1.00V

Description This command sets the amplitude of the output pulse. The amplitude can be set from 0.10V through 5.0V. The value can be specified, in volts, in an integer or decimal format, with or without the units. The default units (that is, the units used when you do not enter units) are volts. MIN or MAX can be specified.

The possible value for the amplitude of the output pulse is dependent on the level set by :PULSe:LEVel:OFFSet.

The amplitude is directly coupled to the values set by :PULSe:LEVel:HIGH and :PULSe:LEVel:LOW.

Amplitude ≡ *High Level – Low Level*

Overvoltage Disabling:
The window in the following diagram defines the output voltage conditions under which an enabled output remains enabled.

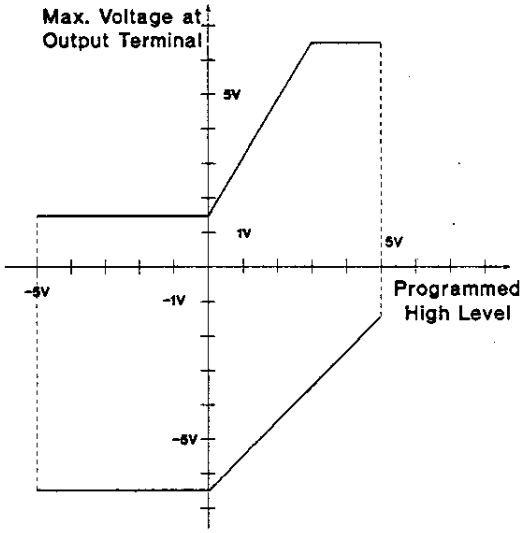


Figure 5-2. Output Voltage Window

A voltage that exceeds a window limit causes an output to be automatically disabled.

Note If an output drives into an open circuit, the output voltage is doubled. The instrument disables the outputs if the amplitude exceeds 6.5V or if the level goes above 7.0V or below -7.0V

Example

```
OUTPUT 714;":PULS1:LEV:AMPL 2.10V"
```

:PULSe:LEVel:AMPLitude?

Syntax :PULSe"[1|2]:LEVel:AMPLitude?

Description This command returns the amplitude setting of the output signal. The response is in decimal form, for example: 2.10. The units of the returned value are volts.

Example
```
OUTPUT 714;":PULS1:LEV:AMPL?"
OUTPUT 714;A$
```

:PULSe:LEVel:HIGH

Syntax :PULSe"[1|2]:LEVel:HIGH<wsp><value>|MIN|MAX

-5.1V ≤ <value> ≤5.2V or MIN or MAX
Specified Range: -4.90V to 5.00V
Resolution: 0.01V
Default: 0.50V

Description This command sets the high level of the output pulse. The high level can be set from -4.90V through 5.0V. The value can be specified, in volts, in an integer or decimal format, with or without the units. The default units (that is, the units used when you do not enter units) are volts. MIN or MAX can be specified.

The possible value for the high level of the output pulse is dependent on the level set by :PULSe:LEVel:LOW.

The high level, is directly coupled to the values set by :PULSe:LEVel:AMPLitude and :PULSe:LEVel:OFFSet.

$$High\ Level \equiv Offset + (\frac{Amplitude}{2})$$

Overvoltage Disabling: See ":PULSe:LEVel:AMPLitude"

Example
```
OUTPUT 714;":PULS1:LEV:HIGH 1.55V"
```

:PULSe:LEVel:HIGH?

Syntax :PULSe"[1|2]:LEVel:HIGH?

Description This command returns the value of the high level of the output signal. The response is in decimal form, for example: 1.55. The units of the returned value are volts.

Example
```
OUTPUT 714;":PULS1:LEV:HIGH?"
ENTER 714;A$
```

:PULSe:LEVel:LIMit

Syntax :PULSe''[1|2]:LEVel:LIMit<wsp>OFF|ON|0|1

Description This command enables or disables the output signal limit. Enabling the limit function makes the current high and low levels the limit levels. The levels of the output signal will not exceed these limits.

:PULSe:LEVel:LIMit OFF or :PULSe:LEVel:LIMit 0 (Default)
The limit function is disabled.

:PULSe:LEVel:LIMit ON or :PULSe:LEVel:LIMit 1
The limit function is enabled.

Example
```
OUTPUT 714;":PULS:LEV:LIM ON"
```

:PULSe:LEVel:LIMit?

Syntax :PULSe''[1|2]:LEVel:LIMit?

Description This command returns the status of the limit function. The response is either OFF, if the limit function is disabled, or ON if the limit function is enabled.

Example
```
OUTPUT 714;":PULS:LEV:LIM?"
ENTER 714;A$
```

:PULSe:LEVel:LIMit:AMPLitude?

Syntax :PULSe''[1|2]:LEVel:LIMit:AMPLitude?

Description This command returns the value of the output signal amplitude limit. The response is in decimal form, for example: 1.55. The units of the returned value are volts.

Example
```
OUTPUT 714;":PULS1:LEV:LIM:AMPL?"
ENTER 714;A$
```

:PULSe:LEVel:LIMit:HIGH?

Syntax :PULSe''[1|2]:LEVel:LIMit:HIGH?

Description This command returns the value of the output signal high level limit. The response is in decimal form, for example: 1.55. The units of the returned value are volts.

Example
```
OUTPUT 714;":PULS1:LEV:LIM:HIGH?"
ENTER 714;A$
```

:PULSe:LEVel:LIMit:LOW?

Syntax :PULSe''[1|2]:LEVel:LIMit:LOW?

Description This command returns the value of the output signal low level limit. The response is in decimal form, for example: 1.55. The units of the returned value are volts.

Example
```
OUTPUT 714;":PULS1:LEV:LIM:LOW?"
ENTER 714;A$
```

:PULSe:LEVel:LIMit:OFFSet?

Syntax :PULSe''[1|2]:LEVel:LIMit:OFFSet?

Description This command returns the value of the output signal offset limit. The response is in decimal form, for example: 1.55. The units of the returned value are volts.

Example
```
OUTPUT 714;":PULS1:LEV:LIM:OFFS?"
ENTER 714;A$
```

:PULSe:LEVel:LOW

Syntax :PULSe''[1|2]:LEVel:LOW<wsp><value>|MIN|MAX

-5.2V ≤ <value> ≤5.1V or MIN or MAX
Specified Range: -5.00V to 4.90V
Resolution: 0.01V
Default: -0.50V

Description This command sets the low level of the output pulse. The low level can be set from -5.00V through 4.90V. The value can be specified, in volts, in an integer or decimal format, with or without the units. The default units (that is, the units used when you do not enter units) are volts. MIN or MAX can be specified.

The possible value for the low level of the output pulse is dependent on the level set by :PULSe:LEVel:HIGH.

The low level, is directly coupled to the values set by :PULSe:LEVel:AMPLitude and :PULSe:LEVel:OFFSet.

$$Low\ Level \equiv Offset - (\frac{Amplitude}{2})$$

Overvoltage Disabling: See ":PULSe:LEVel:AMPLitude"

Example

```
OUTPUT 714;":PULS1:LEV:LOW 1.45V"
```

:PULSe:LEVel:LOW?

Syntax :PULSe''[1|2]:LEVel:LOW?

Description This command returns the value of the output signal low level. The response is in decimal form, for example: 1.45. The units of the returned value are volts.

Example

```
OUTPUT 714;":PULS1:LEV:LOW?"
ENTER 714;A$
```

:PULSe:LEVel:OFFSet

Syntax :PULSe''[1|2]:LEVel:OFFSet<wsp><value>|MIN|MAX

-5.15V ≤ <value> ≤ 5.15V or MIN or MAX
Specified Range: -4.95V to 4.95V
Resolution: 0.01V
Default: 0.00V

Description This command sets the offset of the output pulse. The offset can be set from -4.95V through 4.95V. The value can be specified, in volts, in an integer or decimal format, with or without the units. The default units (that is, the units used when you do not enter units) are volts. MIN or MAX can be specified.

The possible value for the offset of the output pulse is dependent on the level set by :PULSe:LEVel:AMPLitude.

The offset is directly coupled to the values set by :PULSe:LEVel:HIGH and :PULSe:LEVel:LOW.

$$Offset \equiv \frac{(High\ Level + Low\ Level)}{2}$$

Overvoltage Disabling: See ":PULSe:LEVel:AMPLitude".

Example

```
OUTPUT 714;":PULS1:LEV:OFFS 2.10V"
```

:PULSe:LEVel:OFFSet?

Syntax :PULSe''[1|2]:LEVel:OFFSet?

Description This command returns the offset setting of the output signal. The response is in decimal form, for example: 2.10. The units of the returned value are volts.

Example
```
OUTPUT 714;":PULS1:LEV:OFFS?"
OUTPUT 714;A$
```

:PULSe:TIMing:DELay

Syntax :PULSe[1|2]:TIMing:DELay<wsp><value>|MIN|MAX

0.00 ≤ <value> ≤ 99.9ms or MIN or MAX
Specified Range: 0.00ns to 90.0ms
Resolution:

Range	Resolution
0.00ns - 9.99ns	0.01ns
10.0ns - 99.9ns	0.1ns
100ns - 999ns	1ns
1.00μs - 9.99μs	0.01μs
10.0μs - 99.9μs	0.1μs
100μs - 999μs	1μs
1.00ms - 9.99ms	0.01ms
10.0ms - 99.9ms	0.1ms

Default: 0.00ps
Fixed delay (from the trigger output to the differential outputs): 18ns

Description This command sets the delay time between the trigger (as output at the TRIG OUTPUT) and the start of the pulse. There is always a fixed delay between the trigger and the start of the pulse. This is 14ns in trigger and external width mode, and 16ns in burst and gate modes. The value set here is added to this fixed delay. The delay can be programmed from 0.00ps to 99.9ms. Enter the value for the delay in seconds in integer, decimal or exponential form, with or without units. The default units (that is, the units used when you do not enter units) are seconds. You can also specify MIN or MAX.

The possible values for the delay are dependent on the values set using :PULSe[1|2]:TIMing:DOUBle, :PULSe[1|2]:TIMing:DOUBle:MODE, :PULSe[1|2]:TIMing:DutyCYCle, :PULSe[1|2]:TIMing:DutyCYCle:MODE, :PULSe[1|2]:TIMing:PERiod, and :PULSe[1|2]:TIMing:WIDTh.

See Table 5-6 for a description of the timing relationships.

Table 5-6.
Maximum Values of WIDTH, DCYC and DELAY versus PERIOD

PERIOD(ns)	On-Time	DELAY
3.33 ... 4.99	0.5×PERIOD	0.5×PERIOD-1ns
5.00 ... 19.9	0.7×PERIOD-1ns	0.7×PERIOD-2ns
≥ 20.0	0.9×PERIOD-5ns	0.9×PERIOD-6ns

Note On-Time in this table refers to the value set by DCYC (duty cycle) as well as by WIDTH.

Note The value for the delay is effective only when `:PULS:TIM:DOUB:MODE` is `OFF`.

Example
```
OUTPUT 714;":PULS1:TIM:DEL 11.1E-9" or
OUTPUT 714;":PULS1:TIM:DEL 11.1ns"
```

:PULSe:TIMing:DELay?

Syntax `:PULSe[1|2]:TIMing:DELay?`

Description This command returns the value currently set for the delay parameter. The response is in exponential form, for example: 11.1E-9. The units of the returned value are seconds.

Example
```
OUTPUT 714;":PULS:TIM:DEL?"
ENTER 714;A$
```

:PULSe:TIMing:DOUBle

Syntax `:PULSe[1|2]:TIMing:DOUBle<wsp><value>|MIN|MAX`

`2.50ns ≤ <value> ≤ 99.9ms` or `MIN` or 'MAX''
Specified Range: 3.33ns to 90.0ms
Resolution:

Range	Resolution
3.33ns - 9.99ns	0.01ns
10.0ns - 99.9ns	0.1ns
100ns - 999ns	1ns
1.00μs - 9.99μs	0.01μs
10.0μs - 99.9μs	0.1μs
100μs - 999μs	1μs
1.00ms - 9.99ms	0.01ms
10.0ms - 99.9ms	0.1ms

Default: 200μs

Description This command sets the delay time between the first and second pulse in double pulse mode. The delay can be from 3.00ns to 99.9ms. Enter the value for the delay in seconds in integer, decimal or exponential form, with or without units. The default units (that is, the units used when you do not specify units) are seconds. You can also specify `MIN` or `MAX`.

The possible values for the delay between pulses are dependent on the values set using `:PULSe[1|2]:TIMing:DELay`, `:PULSe[1|2]:TIMing:DOUBle:MODE`, `:PULSe[1|2]:TIMing:DutyCYCle`, `:PULSe[1|2]:TIMing:DutyCYCle:MODE`, `:PULSe[1|2]:TIMing:PERiod`, and `:PULSe[1|2]:TIMing:WIDTh`.

See Table 5-7 for a description of the timing relationships.

Table 5-7. Maximum Values of DOUB versus PERIOD

PERIOD(ns)	DOUB
3.33 ... 7.19	n/a
7.20 ... 9.99	0.5×PERIOD
≥ 10.0	0.9×PERIOD-4ns

Note

The value for the delay between pulses is effective only when `:PULS:TIM:DOUB:MODE` is `ON`.

Example

```
OUTPUT 714;":PULS1:TIM:DOUB 11.1E-9" or
OUTPUT 714;":PULS1:TIM:DOUB 11.1ns"
```

:PULSe:TIMing:DOUBle?

Syntax `:PULSe[1|2]:TIMing:DOUBle?`

Description This command returns the value currently set for the double pulse delay parameter. The response is in exponential form, for example: 11.1E-9. The units of the returned value are seconds.

Example
```
OUTPUT 714;":PULS:TIM:DOUB?"
ENTER 714;A$
```

:PULSe:TIMing:DOUBle:MODE

idx|Pulse:double|

Syntax `:PULSe[1|2]:TIMing:DOUBle:MODE<wsp>OFF|ON|0|1`

Description This command selects whether the channel is in double pulse mode or pulse delay mode. The two modes are mutually exclusive, pulse delay (delay of the first pulse) with respect to trigger is not available in the double pulse mode.

`:PULSe:TIMing:DOUBle:MODE OFF` or `:PULSe:TIMing:DOUBle:MODE 0` (Default)
Pulse delay is selected.

`:PULSe:TIMing:DOUBle:MODE ON` or `:PULSe:TIMing:DOUBle:MODE`
Double pulse is selected

Example
```
OUTPUT 714;":PULS1:TIM:DOUB:MODE ON"
```

:PULSe:TIMing:DOUBle:MODE?

idx|Pulse:double|

Syntax `:PULSe[1|2]:TIMing:DOUBle:MODE?`

Description This command indicates whether double pulse or pulse delay operation is currently enabled. The response is either `OFF`, if pulse delay operation is enabled, or `ON`, if double pulse operation is enabled.

Example
```
OUTPUT 714;":PULS1:TIM:DOUB:MODE?"
ENTER 714;A$
```

:PULSe:TIMing:DutyCYCle

Syntax :PULSe[1|2]:TIMing:DutyCYCle<wsp><value>|MIN|MAX

1 ≤ <value> ≤ 99 or MIN or MAX
Specified Range: 1% to 90%
Resolution: 1%
Default: 50%

Description This command sets the duty cycle of the pulse. This is a percentage value input in integer form, with or without units. Alternatively MIN or MAX can be input. The units for percentage are expressed by PCT.

The possible values for the delay between pulses are dependent on the values set using :PULSe[1|2]:TIMing:DELay, :PULSe[1|2]:TIMing:DOUBle, :PULSe[1|2]:TIMing:DOUBle:MODE, :PULSe[1|2]:TIMing:DutyCYCle:MODE, :PULSe[1|2]:TIMing:PERiod, and :PULSe[1|2]:TIMing:WIDTh.

See Table 5-6 for a description of the timing relationships.

Note The value for the duty cycle is effective only when :PULS:TIM:DCYC:MODE is ON.

Example
```
OUTPUT 714;":PULS1:TIM:DCYC 11PCT"
```

:PULSe:TIMing:DutyCYcle?

Syntax :PULSe[1|2]:TIMing:DutyCYCle?

Description This command returns the duty cycle. It is returned in integer format, for example: 11. The returned value is a percentage.

Example
```
OUTPUT 714;":PULS1:TIM:DCYC?"
ENTER 714;A$
```

:PULSe:TIMing:DutyCYCle:MODE

Syntax :PULSe[1|2]:TIMing :DutyCYCle:MODE<wsp>OFF|ON|0|1

Description This command selects between the duty cycle or width parameters to decide the on-time of the pulse.

If Double pulse is inactive then Pulse width = PERIOD*DCYC/100
If Double pulse is inactive then Double pulse width = PERIOD*DCYC/200

:PULSe:TIMing:DutyCYCle:MODE OFF or :PULSe:TIMing:DutyCYCle:MODE 0 (Default)
Duty cycle is disabled. The pulse width or double pulse width is specified by the WIDTH parameter.

:PULSe:TIMing:DutyCYCle:MODE ON or :PULSe:TIMing:DutyCYCle:MODE 1
Duty cycle is enabled.

:INP:TRIG:MODE TRIG and :PULS:TIM:DCYC:MODE ON are incompatible.

Example

```
OUTPUT 714;":PULS1:TIM:DCYC:MODE ON"
```

:PULSe:TIMing:DutyCYCle:MODE?

Syntax :PULSe[1|2]:TIMing:DutyCYCle:MODE?

Description This command returns whether the pulse on-time is decided by the duty cycle or the width parameter. The response is one of OFF or ON. OFF indicates that the on-time is determined by the width, ON indicates that the on-time is calculated from the duty cycle and the period.

Example

```
OUTPUT 714:":PULS1:TIM:DCYC:MODE?"
ENTER 714;A$
```

:PULSe:TIMing:PERiod

Syntax :PULSe[1|2]:TIMing:PERiod<wsp><value>|MIN|MAX

3.00ns ≤ <value> ≤ 99.9ms or MIN or MAX
Specified Range: 3.33ns to 99.9ms
Resolution:

Period, Delay, Double, Width and Transition Time Ranges

Range	Resolution
3.33ns - 9.99ns	0.01ns
10.0ns - 99.9ns	0.1ns
100ns - 999ns	1ns
1.00μs - 9.99μs	0.01μs
10.0μs - 99.9μs	0.1μs
100μs - 999μs	1μs
1.00ms - 9.99ms	0.01ms
10.0ms - 99.9ms	0.1ms

Default: 1.00ms

Description This command sets the period of the output signal during in the automatic, burst and gate operating modes. In the trigger and external width operating modes the period is controlled by the EXT INPUT signal. The period can be programmed between 3.00ns and 99.9ms. The value is input as in integer, decimal or exponential format, with or without units. The default units (that is, those units used when you do not input units) are seconds. Alternatively you can input MIN or MAX.

The period parameter is common to Channels 1 and 2. It cannot be set individually for each channel.

The possible values for the delay between pulses are dependent on the values set using :PULSe[1|2]:TIMing:DELay, :PULSe[1|2]:TIMing:DOUBle, :PULSe[1|2]:TIMing:DOUBle:MODE, :PULSe[1|2]:TIMing:DutyCYCle, :PULSe[1|2]:TIMing:DutyCYCle:MODE, and :PULSe[1|2]:TIMing:WIDTh.

See Table 5-6 for a description of the timing relationships.

Example

```
OUTPUT 714;":PULS2:TIM:PER 1.11E-3" or
OUTPUT 714;":PULS2:TIM:PER 1.11ms"
```

:PULSe:TIMing:PERiod?

Syntax `:PULSe[1|2]:TIMing:PERiod?`

Description This command returns the value set for the period of the output signal. The response is in exponential form, for example 1.11E-3. The units of the returned value are seconds.

Example

```
OUTPUT 714;":PULS2:TIM:PER?"
ENTER 714;A$
```

:PULSe:TIMing:WIDTh

Syntax `:PULSe[1|2]:TIMing:WIDTh<wsp><value>|MIN|MAX`

`1.00ns` ≤ <value> ≤ `99.9ms` or `MIN` or `MAX`
Specified Range: 1.5ns to 90.0ms
Resolution:

Range	Resolution
1.50ns - 9.99ns	0.01ns
10.0ns - 99.9ns	0.1ns
100ns - 999ns	1ns
1.00μs - 9.99μs	0.01μs
10.0μs - 99.9μs	0.1μs
100μs - 999μs	1μs
1.00ms - 9.99ms	0.01ms
10.0ms - 99.9ms	0.1ms

Default: 100μs

Description This command sets the on-time of the pulse. The value can be programmed from 1.00ns to 99.9ms. The value is input in seconds in integer, decimal or exponential form, with or without units. The default units (that is, the units used when you do not input units) are seconds.

The possible values for the delay between pulses are dependent on the values set using `:PULSe[1|2]:TIMing:DELay`, `:PULSe[1|2]:TIMing:DOUBle`, `:PULSe[1|2]:TIMing:DOUBle:MODE`, `:PULSe[1|2]:TIMing:DutyCYCle`, `:PULSe[1|2]:TIMing:DutyCYCle:MODE`, and `:PULSe[1|2]:TIMing:PERiod`.

See Table 5-6 for a description of the timing relationships.

Note The value for the width is effective only when `:PULS:TIM:DCYC:MODE` is `OFF`.

Example `OUTPUT 714;":PULS1:TIM:WIDT 111E-6"` or
`OUTPUT 714;":PULS1:TIM:WIDT 111μs"`

:PULSe:TIMing:WIDTh?

Syntax `:PULSe[1|2]:TIMing :WIDTh?`

Description This command returns the setting for the width of the pulse. The response is in exponential form, for example: 111E-6. The units of the response are seconds.

Example
```
OUTPUT 714;":PULS1:TIM:WIDT?"
ENTER 714;A$
```

:SYSTem

:SYSTem:DERRor?

Syntax :SYSTem:DERRor?<wsp>[NUMeric|STRing|SCPI]

Description This command is an alias for :SYSTem:ERRor?. It is implemented for compatibility reasons.

:SYSTem:ERRor?

Syntax :SYSTem:ERRor?<wsp>[NUMeric|STRing|SCPI]

Description This command returns an error code from the error queue, the error queue is organized as a First-In, First-Out (FIFO) queue, this means that it is always the oldest error in the queue that is returned. Only one error is returned per query.

Appendix B contains a list of errors reported by :SYST:ERR?.

Error code 0 = no errors

:SYSTem:ERRor? SCPI (Default)
The error code is returned in SCPI format, for example, -222, "Data out of range; Amplitude value too low".

:SYSTem:ERRor? NUMeric
The error code is returned, for example, -222.

:SYSTem:ERRor? STRing
The oldest error code plus a brief description of the error is returned, for example, -222,<Data out of range>,<Amplitude value too low>.

Example

```
DIM A$ [1000]
OUTPUT 714;":SYST:ERR?" or
OUTPUT 714;":SYST:ERR? SCPI" or
OUTPUT 714;":SYST:ERR? NUM" or
OUTPUT 714;":SYST:ERR? STR"
ENTER 714;A$
```

:SYSTem:SET

Syntax :SYSTem:SET<wsp><data>

Description This command transfers the setting information for the instrument as a string of binary data.

Example See the example for the application of :SYST:SET?.

:SYSTem:SET?

Syntax :SYSTem:SET?

Description This command returns the current setting of the instrument. The response is as binary data.

The block of data is identical with the block of data saved and recalled by the *SAV and *RCL commands.

Example

```
10  DIM Query$[100],Setting$[200]BUFFER
20  !
30  ASSIGN @Hpib_device TO 714
40  ASSIGN @Path TO BUFFER Setting$
50  !
60  !Read current setting from the HP 70330A
70  Query$=":SYST:SET?"
80  GOSUB Fetch
90  !
100 !Write stored setting to the HP 70330A
110 Count_out=146
120 GOSUB Write
130 !
140 STOP
150 !
160 Fetch: !
170 Output @Hpib_device;Query$
180 TRANSFER @Hpib-device TO @Path;END,WAIT
190 RETURN
200 !
210 Write: !
220 TRANSFER @Path TO @Hpib_device;
    COUNT count_out, WAIT
230 RETURN
240 !
250 END
```

6

Remote Programming Examples

This chapter gives some programming examples. The language used for the programming is BASIC 5.1 Language System used on HP 9000 Series 200/300 computers.

These programming examples do not cover the full command set for the instrument. They are intended only as an introduction to the method of programming the instrument. The programming examples use the HP-IB.

Example 1 - Checking Communication

Function

This program sends a query, and displays the reply.

Listing

```
10    !--------------------------------
20    !
30    ! HP 70330A Programming Example 1
40    !
50    ! A simple communications check
60    !
70    !--------------------------------
80    !
90    !  Definitions and initialisation
100   !

110   Pg=714
```

This statement sets the address of the pulse generator. The first 7 is to access the HP-IB card in the controller, the 14 is the HP-IB address of the pulse generator.

```
120   DIM String$[50]
130   !
140   CLEAR SCREEN
150   PRINT TABXY(5,10);"Programming Example 1, Simple communications"
160   !
170   ! Send an IDN query to the pulse generator to get the identification
180   !
190   OUTPUT Pg;"*idn?"
200   ENTER Pg;String$
210   PRINT TABXY(10,12);"Identification : ";String$
220   !
230   END
```

Example 2 - Status Registers and Queues

Function

This program sends commands and queries typed in by the user. The contents of the status byte and the standard event status register are displayed. These registers are updated each time a Service ReQuest (SRQ) occurs. The number of the most recent error, and the most recent contents of the output queue are also displayed.

Note

Do not use the program given here with the :SYSTEM:SET? query. This command returns binary data. An example for the :SYSTEM:SET? query is given in chapter 5.

Listing

```
10    !-----------------------------------------------
20    !
30    ! HP 70330A Programming Example 2
40    !
50    ! Status Structure and a useful self learning tool
60    !
70    !-----------------------------------------------
80    !
90    !  Definitions and initialisation
100   !
110   INTEGER Value,B,Quot,Xpos,Ypos
120   DIM Inp$[100]
130   DIM A$[300]
140   Pg=714
150   ON INTR 7 GOSUB Pmm_srq
160   !
170   ! Mask the registers
180   !

190   OUTPUT Pg;"*sre 48;*ese 189"
```

*The *SRE 48 command enables bits 5 (ESB) and 4 (MAV) in the status byte (bit 6 (SRQ) cannot be disabled in this register). The *ESE 189 command enables all of the used bits in the Event Status Register.*

```
200   !
210   ! Set up the screen
220   !
230   CLEAR SCREEN
240   PRINT TABXY(40,3);"Status Byte"
250   PRINT TABXY(4,1);"     SRQ ESB MAV"
260   PRINT TABXY(4,2);"+---+---+---+---+---+---+---+---+"
270   PRINT TABXY(4,3);":   :   :   :   :   :   :   :   :"
280   PRINT TABXY(4,4);"+---+---+---+---+---+---+---+---+"
290   PRINT TABXY(4,5);"           ^"
300   PRINT TABXY(4,6);"           :"
310   PRINT TABXY(4,7);"+------------------------------+"
320   PRINT TABXY(4,8);":          OR                  :"
330   PRINT TABXY(4,9);"+------------------------------+"
340   PRINT TABXY(4,10);"  ^       ^   ^   ^   ^       ^"
350   PRINT TABXY(4,11);"+---+---+---+---+---+---+---+---+"
360   PRINT TABXY(4,12);":   :   :   :   :   :   :   :   :"
370   PRINT TABXY(4,13);"+---+---+---+---+---+---+---+---+"
380   PRINT TABXY(4,14);" PON     CME EXE DDE QYE     OPC"
390   PRINT TABXY(40,12);"Standard Event Status Register"
400   PRINT TABXY(4,17);"Last Error   :"
```

Example 2 - Status Registers and Queues

```
410   PRINT TABXY(4,18);"Output Queue :"
420   !
430   ! Start the program loop and enable the interrupt for the errors
440   !
450   Ende=0
460   ENABLE INTR 7;2
470   !
480   ! The Central Loop
490   !
500   REPEAT
510     INPUT "Command ? ",Inp$
520     OUTPUT Pg;Inp$
530     WAIT 1.0
540   UNTIL Ende=1
550   GOTO 1370
560   !
570   !-------------------------------------------------------------------------
580 Pmm_srq:   ! Interrupt handling subroutine to display the status and the
590   ! error and output queues
600   !-------------------------------------------------------------------------
610   !
620   ! Get the value for the status byte
630   !
640   Value=SPOLL(Pg)
650   !
660   ! Initialize and start the display of the registers
670   !
680   Ypos=3
690   FOR Z=0 TO 1
700     B=128
710     Xpos=6
720     !
730     ! Do it for each bit
740     !
750     REPEAT
760       Quot=Value DIV B
770       !
780       ! If the bit is set then display 1
790       !
800       IF Quot>0 THEN
810         PRINT TABXY(Xpos,Ypos);"1"
820         Value=Value-B
830         !
840         ! If MAV is set, then get and display the output queue contents
850         !
860         IF Z=0 THEN
870           IF B=16 THEN
880             ENTER Pg;A$
890             PRINT TABXY(21,18);"                                          "
900             PRINT TABXY(21,18);A$
910           ELSE
920             PRINT TABXY(21,18);"                                          "
930           END IF
940         END IF
950         !
960         ! If the bit is not set then display 0
970         !
980       ELSE
990         PRINT TABXY(Xpos,Ypos);"0"
1000      END IF
1010      !
```

```
1020      ! Set up for the next iteration
1030      !
1040      B=B DIV 2
1050      Xpos=Xpos+4
1060    UNTIL B=0
1070    !
1080    ! Now that the status byte is displayed, get the Standard Events
1090    ! Status register
1100    !
1110    OUTPUT Pg;"*esr?"
1120    ENTER Pg;Value
1130    !
1140    ! Set up to display the ESR
1150    !
1160    Ypos=12
1170  NEXT Z
1180  !
1190  ! Read and display any messages in the error queue
1200  !
1210  PRINT TABXY(21,17);"                                                    "
1220  REPEAT

1230    OUTPUT Pg;":syst:err? str"
1240    ENTER Pg;A$
```

The :SYST:ERR? query gets the last error in the error queue.

```
1250    IF A$<>"0,<No error>" THEN
1260      PRINT TABXY(21,17);"                                                  "
1270      PRINT TABXY(21,17);A$
1280    END IF
1290  UNTIL A$="0,<No error>"
1300  !
1310  ! Clear the status structure and reenable the interrupt before returning
1320  !
1330  OUTPUT Pg;"*cls"
1340  ENABLE INTR 7
1350  RETURN
1360  !
1370  END
```

Example 3 - A Sample Session

Function

This short program sets up the instrument to output a 1MHz pulse.

Listing

```
10    !--------------------------------
20    !
30    ! HP 70330A Programming Example 3
40    !
50    ! Setting up the instrument
60    !
70    !--------------------------------
80    !
90    ! Declarations and Initializations
100   !
110   Pg=714
120   OUTPUT Pg;"*cls;*rst"
130   !
140   ! Set up the instrument
150   !
160   ! Set the trigger mode to automatic
170   OUTPUT Pg;":INP:TRIG:MODE AUTO"
180   ! Set the period to 1us
190   OUTPUT Pg;":PULS:TIM:PER 1E-6"
200   ! Set the channel 1 pulse width to 1ns, and the delay to 0s
210   OUTPUT Pg;":PULS1:TIM:WIDT 1.5E-9;DEL 0"
220   ! Set the leading and trailing edge times for channel 1
230   OUTPUT Pg;":PULS1:EDGE:LEAD 1E-9;TRA 1E-9"
240   ! Set the channel 1 pulse to vary between 0 and 1V
250   OUTPUT Pg;":PULS1:LEV:HIGH 1;LOW 0"
260   !
270   ! Enable the output
280   !
290   OUTPUT Pg;":OUTP1:PULS:STAT ON"
300   !
310   ! Wait for 5 seconds, then end.
320   !
330   WAIT 5
340   OUTPUT Pg;":OUTP1:PULS:STAT OFF"
350   END
```

A

Specifications

Specifications

Specifications describe the instrument's warranted performance. Non-warranted values are described as "typical". All specifications apply after a 30 minute warm-up phase with 50Ω load resistance at all outputs, and are valid at 0°C to 55°C ambient temperature.

Timing Parameters

Common Specifications

Measurement conditions. Auto mode, measured at 50% of amplitude and fastest transitions.

Resolution: 3 digits, best case: 10ps

Repeatability: factor 4 better than accuracy

RMS-jitter: 0.025% of programmed value + 15ps (0.05% of programmed value +15ps for range 10ns to 100ns)

Period

Range: 3.33ns to 99.9ms

Accuracy: ±5% of programmed value ±100ps

Width

Range: 1.5ns to 90.0ms (see Table A-1 for maximum values)

Accuracy: ±5% of programmed value ±250ps

Delay

(measured between trigger output and main output)

Fixed delay (trigger out to main out): 18ns

Variable range: from 0ns to 90.0ms (see Table A-1 for maximum value)

Accuracy: ±5% of programmed value ±1.5ns

Table A-1. Maximum Width and Delay Values versus Pulse Period

Period (ns)	Width	Delay
3.33 to 4.99	50% of Period	(50% of Period) - 1ns
5.00 to 19.9	(70% of Period) - 1ns	(70% of Period) - 2ns
≥20.0	(90% of Period) - 5ns	(90% of Period) - 6ns

Double Pulse

(period >7.20ns)
(double pulse and delay are mutually exclusive)

Range: 3.33ns to 90.0ms (see Table A-2 for maximum value)

Accuracy: ±5% of programmed value ±250ps

Table A-2. Maximum Double Pulse Values versus Pulse Period

Period (ns)	Double Pulse
3.33 to 7.19	not available
7.20 to 9.99	50% of Period
≥10.0	(90% of Period) - 4ns
	at minimum width

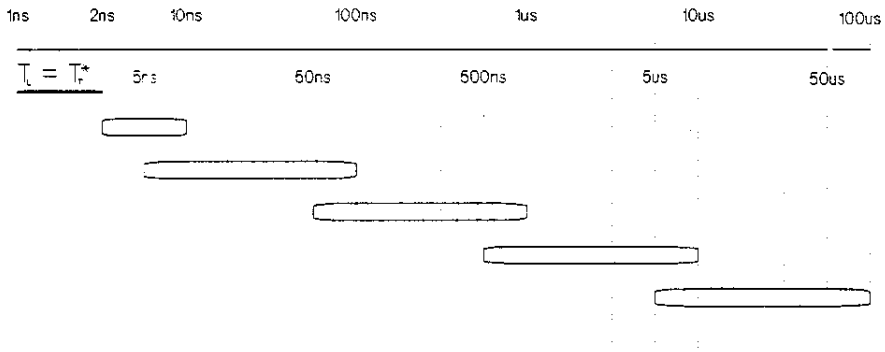
Transition Times

(measured between 10% and 90% of amplitude)

Range: 1ns* to 0.1ms
* typical at 20% to 80% of amplitude: 600ps

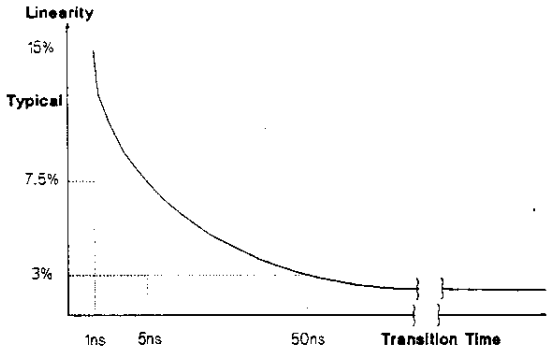
Accuracy: ±10% of programmed value ±300ps

Leading and trailing edges are independently programmable within the following ranges:



* Leading and trailing edges are assigned the same value.

Linearity:



Output Parameters

Output levels double when driving into open circuits. Instrument disables outputs if levels exceed ±7.0V or amplitude exceeds 7.0Vp-p.

High level:	-4.90V to + 5.00V
Low Level:	-5.00V to +4.90V
Resolution:	3 digits, best case: 10mV
Level accuracy:	±1% of programmed value ±50mV
Repeatability:	factor 4 better than accuracy
Settling time:	typical 15ns (at fastest transition time)

Pulse Performance

Overshoot:	≤10% of amplitude ±20mV
Ringing:	≤10% of amplitude ±20mV
Preshoot:	≤5% of amplitude ±20mV

Operating Characteristics

Operating characteristics describe the instruments typical, non- warranted performance.

Timing Parameters

Duty Cycle

(width and duty cycle are mutually exclusive)

Range:	1% to 90%
Resolution:	1%

Subject to width and period specifications

Inputs and Outputs

All inputs and outputs have BNC connectors on the front panel.

Main Outputs

(differential outputs)

Amplitude:	100mVp-p to 5Vp-p into 50Ω for pulse widths less than 15ns, the amplitude may decrease by 10% of programmed value.
Offset:	-4.95V to +4.95V into 50Ω
Source impedance:	50Ω±1Ω
Maximum external voltage:	±5V
Short circuit current	(I_{sc}): -200mA ≤ I_{sc} ≤ +200mA
Skew	Between differential outputs of same channel: < 100ps

External Input

Trigger, Gate, Burst and External width mode: Trigger slopes can be selected positive or negative.

Input impedance:	50Ω±2.5Ω
Threshold:	-5V to +5V
Resolution:	100mV
Maximum input voltage:	±10V
Input transition:	<50ns
Input frequency:	dc to 300MHz
Minimum pulse width:	1.5ns
Input sensitivity:	≥300mVp-p

Trigger Output

Levels: high at 2.4V, low at 0.3V into 50Ω

Trigger pulse width:

PERiod	Pulse width
3.33ns to 99.9ns	50% of PER
100ns to 999ns	95% of PER
1.00μs to 9.99μs	99.5% of PER
10.0μs to 99.9ms	99.95% of PER

Transition times:	< 1ns
Source impedance:	50Ω±2.5Ω
Delay	from external input to trigger output: in Trigger and External Width mode: 14ns In Gate and Burst mode: 16ns
Max/Min external voltage:	+7/-2V

Trigger Modes

Auto:	continuous pulse stream.
Trigger:	each active input transition generates a single output pulse or double pulse.
Gate:	external signal enables period generator. First output pulse synchronous with active edge. Last pulse always completed. Period of first pulse may be 25%±500ps longer than subsequent pulses.
External Burst:	each active input transition generates a preprogrammed number of pulses (1 to 9999), minimum burst period is 5ns. Period of first pulse may be 25%±500ps longer than subsequent pulses.
External width:	pulse recovery (external edges toggle output). Output levels and transition times are selectable.

Additional Features

Complement:	normal/complement selectable
Disable:	relays connect/disconnect outputs.
Non-volatile memory:	Current settings are saved on power-down.
Save/Recall:	19 complete settings can be stored in the instrument.
Reset:	instrument is set to default setting
Selftest:	executed at pwer-up and on command

Remote Control

The instrument operates by remote control, there is no local user interface.

ASCII Command	**Execution Time**
One Parameter	<15ms
Change instrument mode	<45ms
Recall complete setting	<75ms†
Save complete setting	<5ms†

† Add transfer time for sending learnstrings.

HP-IB Capabilities

operates according to IEEE standard 488.2, 1987.

HP-IB Interface Function Code:. SH1, AH1, T6, TE0, L4, LE0, SR1, RL0, PP0, DC1, DT1, C0

HP-MSIB Capabilities

operates according to HP-MSIB specifications.

General

Environmental

Operating temperature:	0°C to +55°C
Storage temperature:	-40°C to +70°C
Humidity	(0°C to 40°C): 95%
EMC:	conducted and radiated interference is in compliance with MIL-STD 461B, RE02/part7 +10dB +1dB/100MHz @ f>100MHz
Power Requirements:	see requirements for HP 70001A. All power requirements are supplied by the mainframe HP 70001A.
Power Consumption:	150W max
Dimensions:	5/8-rack-width-module
Weight:	8kg (17.6lb)
Recalibration Period:	1 year recommended
Warranty:	1 year

Ordering Information

HP 70330A 300MHz Pulse Generator with Installation, Verification and Programming Manual included

Option 910 Additional Set of all manuals

Option 915 Service Manual with Component Level Information included

Option W30 3 Year Customer Return Repair Coverage

Option W32 3 Year Customer Return Calibration Coverage

Option H10 No documentation shipped with Instrument

Technical Information in this document is subject to change without notice.

B

Errors

This Appendix lists all non-trivial error messages. The list is ordered by the absolute value of the error number (that is, the negative signs are ignored)

The non-trivial error messages listed here are error messages which are not generated by plain command syntax analysis. Such errors are fully SCPI compatible and are documented in the SCPI manual.

Note 

This appendix describes the set of error messages implemented in revision 01.10 of the HP70330A/HP70332A firmware.

100 **Static RAM (number 2) test failed**
The CPU board selftest detected an error during write/read on static RAM number 2. At least one memory cell did not keep its contents.

101 **Static RAM (number 1) test failed**
The CPU board selftest detected an error during write/read on static RAM number 1. At least one memory cell did not keep its contents.

102 **FLASH EPROM signature test failed**
The CPU board selftest builds a signature from the FLASH EPROM contents. It then compares this with the signature stored in the EPROMs. This test failed. The contents of the FLASH EPROMs are corrupted.

104 **Timer cycle test failed**
The microprocessor did not receive interrupts with the expected timing or did not receive interrupts at all. This indicates an error in the timing generation circuitry of the CPU board.

110 **MSIB test failed**
The MSIB test failed.

-131 **Invalid suffix; Channel suffix must be 1 or 2**
One of the commands that accept a numeric channel suffix was given with a suffix outside the range 1 to 2.

-141 **Invalid character data; Ambiguous MIN/MAX usage for levels**
A program message containing one or more of the level commands (HIGH, LOW, AMPLitude, OFFSet) was sent to the instrument with a mixture of MIN/MAX values that could not be resolved unambiguously. The request was therefore rejected.

Some examples of such ambiguous program messages are:

`PULS:LEV:LOW MAX;AMPL MAX`
The maximum value for the low level depends on the current amplitude and vice versa.
`PULS:LEV:LOW MIN ; HIGH MAX` The minimum value for the low level depends on the current high level and the maximum value for the high level depends on the current low level. This happens to be so because the

distance between the highest allowable high level and the lowest allowable low level is greater than the greatest allowable amplitude.

-141 **Invalid character data; Ambiguous MIN/MAX usage for slopes**
A program message programming both slopes (edges) with a mixture of the symbolic values MIN and MAX (e.g. PULS:EDGE:LEAD MIN; TRA MAX) was sent to the instrument.

Since slopes are managed in terms of slope ranges such a request can not be satisfied unambiguously. The request is therefore rejected.

200 **Adjust values on timing board EEPROM destroyed**
The checksum of the EEPROM on the timing board does not match with the checksum calculated from the contents of the EEPROM. One or more of the correction values are corrupted, correct timing is not possible.

201 **Period test on timing board failed**
The signal from the timing board does not have the expected period or does not toggle at all.

Possible reasons are the failure of the period generation circuitry on the timing board or a bad connection from the timing board to the output board(s).
If error 200 is also generated the reason is possibly the corruption of the adjustment values because, in this case, the period cannot be programmed correctly.

211 **Width test on timing board failed (channel 1)**
The signal from the timing board for channel 1 does not have the expected width.

Possible reasons are the failure of the width generation circuitry on the timing board or a bad connection from the timing board to the output board(s).
If error 200 is also generated, the reason is possibly the corruption of the adjustment values because, in this case, the width cannot be programmed correctly manner.

212 **Delay test on timing board failed (channel 1)**
The signal from the timing board for channel 1 does not have the expected delay.

Possible reasons are the failure of the delay generation circuitry on the timing board or a bad connection from the timing board to the output board(s).
If error 200 is also generated, the reason is possibly the corruption of the adjustment values because, in this case, the delay cannot be programmed correctly.

213 **Timing signal low level failed (channel 1)**
The static low level signal from the timing board (measured on the output board of channel 1) does not have the expected level.

Possible reasons are a failure on the timing board or a bad connection from the timing board to the ouput board.

214 **Timing signal high level failed (channel 1)**
The static high level signal from the timing board (measured on the output board of channel 1) does not have the expected level.

Possible reasons are a failure on the timing board or a bad connection from the timing board to the ouput board.

-221 **Settings conflict; Burst mode not allowed for period < 5 ns**
The trigger mode INPut:TRIGger:MODE BURSt is not allowed for periods less than 5 ns.

-221 **Settings conflict; Double mode not allowed for period < 7.2 ns**
The period value is too low for double mode.

-221 **Settings conflict; High level is above variable limit**
The high level value (given explicitly or computed during a MIN/MAX evaluation) is higher than the currently effective high level limit that was enabled (and set) by the `PULS:LEV:LIM ON` command.

-221 **Settings conflict; Low level is below variable limit**
The low level value (given explicitly or computed during a MIN/MAX evaluation) is lower than the currently effective low level limit that was enabled (and set) by the `PULS:LEV:LIM ON` command.

-221 **Settings conflict; Slopes must be equal for fast rise/fall times**
At least one of the slopes falls into the lowest slope range, in this case both slopes must be equal.

-221 **Settings conflict; Slopes too different**
The difference between the leading edge and the trailing edge is too big to find a slope range that accomodates both.

-221 **Settings conflict; TRIG mode incompatible with DCYC mode**
The trigger mode INPut:TRIGger:MODE TRIGger is incompatible with the mode PULSe:TIMing:DCYCle ON.

221 **Width test on timing board failed (channel 2)**
The signal from the timing board for channel 2 does not have the expected width.

Possible reasons are the failure of the width generation circuitry on the timing board or a bad connection from the timing board to the output board(s).
If error 200 is also generated, the reason is possibly the destruction of the adjustment values because, in this case, the width can not be programmed correctly.

-222 **Data out of range; Amplitude value too high**
Amplitude value is greater than its high limit.

After the final outcome of a program message containing commands out of the AMPL, OFFS, HIGH, LOW group is computed, the following checks are applied in order:

The amplitude is checked against its high limit.
The amplitude is checked against its low limit.
The high level is checked against its high limit.
The high level is checked against its low limit.
The low level is checked against its high limit.
The low level is checked against its low limit.

This error is generated when the first check that fails is the amplitude high limit check.

-222 **Data out of range; Amplitude value too low**
Amplitude value is lower than its low limit.

After the final outcome of a program message containing commands out of the AMPL, OFFS, HIGH, LOW group is computed, the following checks are applied in order:

The amplitude is checked against its high limit.
The amplitude is checked against its low limit.
The high level is checked against its high limit.
The high level is checked against its low limit.
The low level is checked against its high limit.
The low level is checked against its low limit.

This error is generated when the first check that fails is the amplitude low limit check.

-222 **Data out of range; Delay too high**
The delay value is above its absolute limit.

-222 **Data out of range; Delay too low**
The delay value is below its absolute limit.

-222 **Data out of range; Double too high**
The double value is above its absolute limit.

-222 **Data out of range; Double too low**
The double value is below its absolute limit.

-222 **Data out of range; Duty cycle too high**
The duty cycle value is above its absolute limit.

-222 **Data out of range; Duty cycle too low**
The duty cycle value is below its absolute limit.

-222 **Data out of range; High Level value too high**
High level value is greater than its high limit.

After the final outcome of a program message containing commands out of the AMPL, OFFS, HIGH, LOW group is computed, the following checks are applied in order:

The amplitude is checked against its high limit.
The amplitude is checked against its low limit.
The high level is checked against its high limit.
The high level is checked against its low limit.
The low level is checked against its high limit.
The low level is checked against its low limit.

This error is generated when the first check that fails is the high level high limit check.

-222 **Data out of range; High Level value too low**
High level value is lower than its low limit.

After the final outcome of a program message containing commands out of the AMPL, OFFS, HIGH, LOW group is computed, the following checks are applied in order:

The amplitude is checked against its high limit.
The amplitude is checked against its low limit.
The high level is checked against its high limit.
The high level is checked against its low limit.
The low level is checked against its high limit.
The low level is checked against its low limit.

This error is generated when the first check that fails is the high level low limit check.

-222 **Data out of range; Leading edge transition time**
The leading edge transition time is outside absolute limits.

-222 **Data out of range; Low Level value too high**
Low level value is greater than its high limit.

After the final outcome of a program message containing commands out of the AMPL, OFFS, HIGH, LOW group is computed, the following checks are applied in order:

The amplitude is checked against its high limit.
The amplitude is checked against its low limit.
The high level is checked against its high limit.
The high level is checked against its low limit.
The low level is checked against its high limit.
The low level is checked against its low limit.

This error is generated when the first check that fails is the low level high limit check.

-222 **Data out of range; Low Level value too low**
Low level value is lower than its low limit.

After the final outcome of a program message containing commands out of the AMPL, OFFS, HIGH, LOW group is computed, the following checks are applied in order:

The amplitude is checked against its high limit.
The amplitude is checked against its low limit.
The high level is checked against its high limit.
The high level is checked against its low limit.
The low level is checked against its high limit.
The low level is checked against its low limit.

This error is generated when the first check that fails is the low level low limit check.

-222 **Data out of range; Period too high**
The period value is above its absolute limit.

-222 **Data out of range; Period too low**
The period value is below its absolute limit.

-222 **Data out of range; SAV/RCL index out of range**
The *SAV / *RCL command was sent to the instrument with an argument outside the allowable range 0 to 19 for *RCL or the range 1 to $19 for *SAV.

-222 **Data out of range; Trailing edge transition time**
The trailing edge transition time is outside absolute limits.

-222 **Data out of range; Trigger level too high**
The requested trigger level is higher than the absolute upper limit (+5.0 V).

-222 **Data out of range; Trigger level too low**
The requested trigger level is lower than the absolute lower limit (-5.0 V).

-222 **Data out of range; Width too high**
The width value is above its absolute limit.

-222 **Data out of range; Width too low**
The width value is below its absolute limit.

222 **Delay test on timing board failed (channel 2)**
The signal from the timing board for channel 2 does not have the expected delay. Possible reasons are a failure of the delay generation circuitry on the timing board or a bad connection from the timing board to the output board(s). If error 200 is also generated the reason is possibly the destruction of the adjustment values because, in this case, the delay can not be programmed correctly.

223 **Timing signal low level failed (channel 2)**
The static low level signal from the timing board measured on the output board of channel 2 does not have the expected level.

Possible reasons are a failure on the the timing board or a bad connection from the timing board to the ouput board.

224 **Timing signal high level failed (channel 2)**
The static high level signal from the timing board measured on the output board of channel 2 does not have the expected level.

Possible reasons are a failure on the timing board or a bad connection from the timing board to ouput board.

-230 **Data corrupt or stale; Setup data block has wrong ID code**
The binary data block sent to the instrument as an argument of the SYST:SET command has the correct size but the load format identification code stored in the data block is not correct.

Possible causes for this error are:

The data block was not obtained from this instrument via the SYST:SET? command, specifically the block obtained from a 70330 may not be loaded into a 70332 and vice versa.
The data block was obtained from a different software revision in this instrument that is not compatible with the currently running version.
The data block has been corrupted somehow.

-230 **Data corrupt or stale; Setup data block has wrong signature**
The binary data block sent to the instrument as an argument of the SYST:SET command has the correct size and the correct load format identification code but the signature value generated when the SYST:SET command was executed does not match the one stored in the data block.

The possible cause for this error is that the data block has been corrupted somehow.

-230 **Data corrupt or stale; Setup data block has wrong size**
The binary data block sent to the instrument as an argument of the SYST:SET command does not have the correct size.

Possible causes for this error are:

The data block was not obtained from this instrument via the SYST:SET? command.
The data block was obtained from a different software revision in this instrument that is not compatible with the currently running version.
The data block has been corrupted somehow.

-241 **Hardware missing; Unexpected board id**
The firmware allows operation even if configuration errors have been detected. In that case the user could possibly try to execute a command which does not

make sense for that board (e.g. programming output levels on an ECL output board). This error message indicates such an event.

-300 **Device-specific error; Some Functionality is not yet implemented**
Preliminary error message signalling that a section of code is not yet implemented.

300 **Normal output high level of hybrid board failed (fast slope)**
The high level of the normal output for fast slopes is not as expected (channel 1). This indicates an error on the hybrid board.

301 **Inverted output low level of hybrid board failed (fast slope)**
The low level of the inverted output for fast slopes is not as expected (channel 1). This indicates an error on the hybrid board.

302 **Inverted output high level of hybrid board failed (fast slope)**
The high level of the inverted output for fast slopes is not as expected (channel 1). This indicates an error on the hybrid board.

303 **Normal output low level of hybrid board failed (fast slope)**
The low level of the normal output for fast slopes is not as expected (channel 1). This indicates an error on the hybrid board.

304 **Normal output of SMD board failed (fast slope)**
The level of the normal output from the SMD board, for fast slopes, is not as expected (channel 1). This indicates an error on the SMD board.

305 **Inverted output of SMD board failed (fast slope)**
The level of the inverted output from the SMD board, for fast slopes, is not as expected (channel 1). This indicates an error on the SMD board.

308 **Slow slope output low level of SMD board failed**
The low level of the slow slope output from the SMD board is not as expected (channel 1). This indicates an error on the SMD board.

309 **Slow slope output high level of SMD board failed**
The high level of the slow slope output from the SMD board is not as expected (channel 1). This indicates an error on the SMD board.

-310 **System error; Driver error after reload from cstate**
This error only shows up when the instrument hardware is defective or incorrectly configured.

The error is generated when a hardware driver routine returns an error even when the command is known to be OK. This can only happen when the piece of hardware that involved is defective or missing.

-310 **System error; Instrument parameter RAM corrupted**
The parameter consistency check performed at power-on was not able to detect anything more precise than corrupted memory. This error implies that the primary instrument state and all *SAV/*RCL registers are lost.

-310 **System error; Instrument state lost, set to *RST**
The parameter consistency check performed at power-on detected that the primary instrument setting is—or might be—corrupted.

All the primary instrument state variables are set to their *RST values.

-310 **System error; Too many commands already pending**
This error signals a software configuration problem. It occurs when many commands are sent in one program message.

310 **Normal output high level of hybrid board failed (slow slope)**
The high level of the normal output for slow slopes is not as expected (channel 1). This indicates an error on the hybrid board.

311 **Inverted output low level of hybrid board failed (slow slope)**
The low level of the inverted output for slow slopes is not as expected (channel 1). This indicates an error on the hybrid board.

312 **Inverted output high level of hybrid board failed (slow slope)**
The high level of the inverted output for slow slopes is not as expected (channel 1). This indicates an error on the hybrid board.

313 **Normal output low level of hybrid board failed (slow slope)**
The low level of the normal output for slow slopes is not as expected (channel 1). This indicates an error on the hybrid board.

-314 **Save/recall memory lost; One or more SAV/RCL register lost, set to *RST**
The parameter consistency check performed at power-on detected that one or more of the *SAV/*RCL registers might be corrupted.

The suspect recall-registers are set to the *RST setting.

314 **Switch output to complement failed**
After switching to complement mode the levels of the normal and/or complement output are not as expected. This indicates an error on the relays used to switch to complement mode.

320 **One or more ECL normal output(s) high level failed**
The high level of one or more of the ECL normal outputs did not pass the selftest (channel 1).

321 **One or more ECL normal output(s) low level failed**
The low level of one or more of the ECL normal outputs did not pass the selftest (channel 1).

322 **One or more ECL inverted output(s) low level failed**
The low level of one or more of the ECL inverted outputs did not pass the selftest (channel 1).

323 **One or more ECL inverted output(s) high level failed**
The high level of one or more of the ECL inverted outputs did not pass the selftest (channel 1).

324 **Switch to complement failed**
The instrument could not switch the outputs to complement mode (channel 1).

350 **Adjust values on output board EEPROM destroyed**
The checksum of the EEPROM on the output board does not match with the checksum built from the contents of the EEPROM (channel 1). One or more correction values are corrupted, correct slopes and levels are not guaranteed.

400 **Normal output high level of hybrid board failed (fast slope)**
The high level of the normal output for fast slopes is not as expected (channel 2). This indicates an error on the hybrid board.

401 **Inverted output low level of hybrid board failed (fast slope)**
The low level of the inverted output for fast slopes is not as expected (channel 2). This indicates an error on the hybrid board.

402 **Inverted output high level of hybrid board failed (fast slope)**
The high level of the inverted output for fast slopes is not as expected (channel 2). This indicates an error on the hybrid board.

403 **Normal output low level of hybrid board failed (fast slope)**
The low level of the normal output for fast slopes is not as expected (channel 2). This indicates an error on the hybrid board.

404 **Normal output of SMD board failed (fast slope)**
The level of the normal SMD board output for fast slopes is not as expected (channel 2). This indicates an error on the SMD board.

405 **Inverted output of SMD board failed (fast slope)**
The level of the inverted SMD board output for fast slopes is not as expected (channel 2). This indicates an error on the SMD board.

408 **Slow slope output low level of SMD board failed**
The low level of the slow slope output from the SMD board is not as expected (channel 2). This indicates an error on the SMD board.

409 **Slow slope output high level of SMD board failed**
The high level of the slow slope output from the SMD board is not as expected (channel 2). This indicates an error on the SMD board.

410 **Normal output high level of hybrid board failed (slow slope)**
The high level of the normal output for slow slopes is not as expected (channel 2). This indicates an error on the hybrid board.

411 **Inverted output low level of hybrid board failed (slow slope)**
The low level of the inverted output for slow slopes is not as expected (channel 2). This indicates an error on the hybrid board.

412 **Inverted output high level of hybrid board failed (slow slope)**
The high level of the inverted output for slow slopes is not as expected (channel 2). This indicates an error on the hybrid board.

413 **Normal output low level of hybrid board failed (slow slope)**
The low level of the normal output for slow slopes is not as expected (channel 2). This indicates an error on the hybrid board.

414 **Switch output to complement failed**
After switching to complement mode the levels of the normal and/or complement output are not as expected. This indicates an error in the relays used to switch to complement mode.

420 **One or more ECL normal output(s) high level failed**
The high level of one or more of the ECL normal outputs did not pass the selftest (channel 2).

421 **One or more ECL normal output(s) low level failed**
The low level of one or more of the ECL normal outputs did not pass the selftest (channel 2).

422 **One or more ECL inverted output(s) low level failed**
The low level of one or more of the ECL inverted outputs did not pass the selftest (channel 2).

423 **One or more ECL inverted output(s) high level failed**
The high level of one or more of the ECL inverted outputs did not pass the selftest (channel 2).

424 **Switch to complement failed**
The instrument could not switch the outputs to complement mode (channel 2).

450 **Adjust values on output board EEPROM destroyed**
The checksum of the EEPROM on the output board does not match with the

checksum calculated from the contents of the EEPROM (channel 2). One or more correction values are corrupted, correct slopes and levels are not guaranteed.

501 **No timing board installed**
During initialization of the instrument no timing board could be found.

Possible reasons are that there is no timing board installed or the board identification of the timing board is incorrect.

502 **No two channel timing board installed**
During initialization of the instrument a timing board for one channel only was detected along with two output boards.

Possible reasons are that there is only one channel timing board installed or that the board identification of the timing board is incorrect.

503 **More than one timing board installed**
During initialization of the instrument more than one timing board was detected.

Possible reasons are that there are two timing boards installed or that the board identification of at least one board is incorrect.

504 **70330A output board installed in 70332A instrument**
The instrument detected a 70332A processor board but at least one output board for the 70330A.

Possible reasons are the wrong output board is installed in the instrument or the board identification of the processor and/or output board is incorrect.

505 **Only one 70332A output board installed**
Only one output board for the 70332A could be detected.

Possible reasons are that the second output board is not installed or that the board identification is incorrect.

506 **No 70332A output board installed**
No output board for the 70332A could be detected.

Possible reasons are that there is no output board installed or the board identification is incorrect.

507 **70332A output board installed in 70330A instrument**
The instrument detected a 70330A processor board but at least one output board for the 70332A.

Possible reasons are the wrong output board is installed in the instrument or the board identification of the processor and/or output board is incorrect.

508 **Only one 70330A output board installed**
Only one output board for the 70330A could be detected.

Possible reasons are that the second output board is not installed or that the board identification is incorrect.

509 **No 70330A output board installed**
No output board for the 70330A could be detected.

Possible reasons are that there is no output board installed or that the board identification is incorrect.

510 **No output board with trigger amplifier installed**
There are two kinds of output boards. One contains a trigger amplifier, the

other does not. During the initialization of the instrument no output board with a trigger amplifier could be detected.

Possible reasonsare that there is no output board with trigger amplifier installed or the board identification is incorrect.

511 **More than one output board with trigger amplifier installed**
At least two output boards with a trigger amplifier have been detected during power up but only one is necessary.

Possible reasons are that there are two output boards with trigger amplifiers installed or that at least one of the board identifications is incorrect.

555 **Internal configuration error**
The system detected a configuration which could not be resolved.

Possible reasons are that an illegal combination of processor-, timing- and output-boards have been installed in the instrument or that the board identification of at least one board is incorrect.