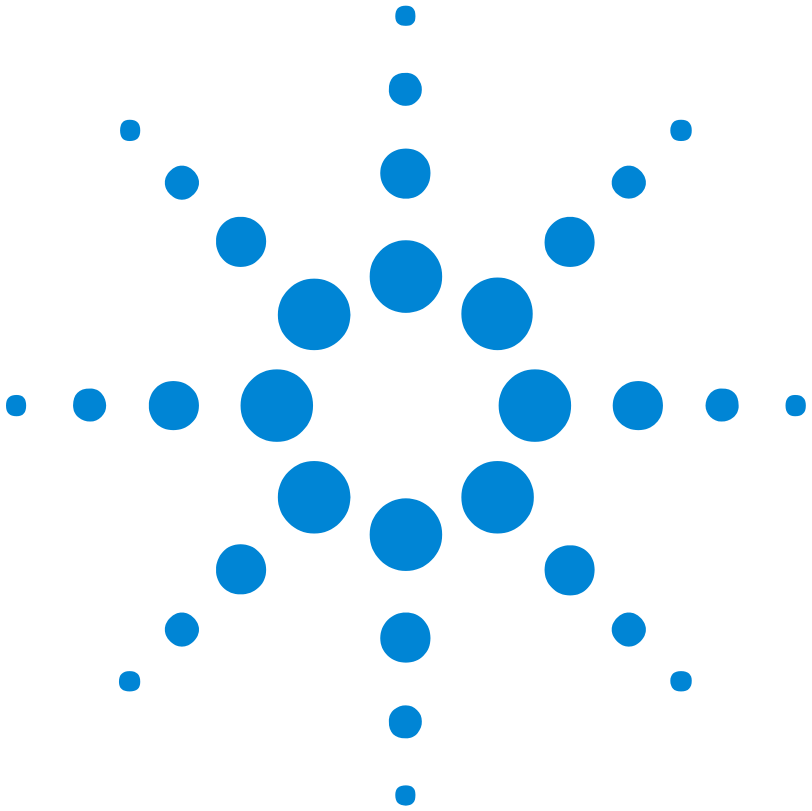


Agilent 81133A/81134A Pulse Generator

Programming Guide

Agilent Technologies

Important Notice

Manual Part Number

5988-7402EN

Revision

March 2007

Printed in Germany

Agilent Technologies
Herrenberger Straße 130
D-71034 Böblingen
Germany

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Trademarks

Windows NT ® and MS Windows ® are U.S. registered trademarks of Microsoft Corporation.

About This Programming Guide

This guide provides information about programming the Agilent 81133A/81134A Pulse/Pattern Generator through the available remote interfaces.

- *"Introduction" on page 9* provides information about the different remote programming interfaces.
- *"Connecting to the Pulse/Pattern Generator for Remote Programming" on page 11* provides information about how to connect to the instrument and gives examples.
- *"SCPI Commands Reference" on page 17* provides detailed information about the available SCPI commands.
- *"Troubleshooting" on page 97* lists the error messages and shows how to solve the errors.
- *"Differences between the 8133A and the 81133A/81134A" on page 101* provides information on how to adapt a program written for the 8133A to the 81133A/81134A instrument.

For examples for setting up generic and advanced signals, please refer to the User Guide.

Contents

Introduction

For controlling the Agilent 81133A/81134A remotely, the instrument provides three different interfaces:

- GPIB

  Using the GPIB connector, the instrument can be controlled from a PC or a UNIX Workstation.

- LAN

  Using the LAN connector, the instrument can be connected to a local area network and can be programmed from a PC.

- USB

  USB is the replacement for GPIB when used on the bench. The language is the same as with GPIB.

NOTE Your instrument's firmware might not be set up for USB. USB functionality is not available with the first release but will be included in a later release of the firmware. Check the Agilent Web page for update information.

Firmware Server and SCPI Commands

All interfaces use the same SCPI-like language to communicate with the instrument's firmware server. The firmware server implements a client server architecture, allowing to connect multiple clients simultaneously.

The GUI also uses this language to communicate with the firmware server. Therefore, everything that can be done via the user interface can also be done via the programming interfaces.

Connecting to the Pulse/Pattern Generator for Remote Programming

The following sections show how to establish the connection between your control PC and the instrument through the available remote interfaces.

Connecting to the Instrument via GPIB

You can use GPIB connections only for controlling the instrument by means of SCPI commands.

To connect to the instrument via GPIB you have to:

- Use GPIB cables to connect the instrument to the test environment.
- Specify the instrument's GPIB address.

  The address is displayed on the user interface. The default address is 13. It can be changed on the user interface in the *Config* Page or with the command *":SYSTem:COMMunicate:GPIB[:SELF]:ADDR" on page 85*.

Example for Connecting via GPIB

The following code example shows how to use the VISA library to connect to the instrument via GPIB.

This example queries a GPIB device for an identification string and prints the results.

Implementation

```
#include <visa.h>
#include <stdio.h>

void main () {

   ViSession defaultRM, vi;
   char buf [256] = {0};


   /* Open session to GPIB device at address 22 */

   viOpenDefaultRM (&defaultRM);
   viOpen (defaultRM, "GPIB0::22::INSTR", VI_NULL,VI_NULL, &vi);


   /* Initialize device */

   viPrintf (vi, "*RST\n");


   /* Send an *IDN? string to the device */

   viPrintf (vi, "*IDN?\n");


   /* Read results */

   viScanf (vi, "%t", &buf);


   /* Print results */

   printf ("Instrument identification string: %s\n", buf);


   /* Close session */

   viClose (vi);
   viClose (defaultRM);

}
```

Connecting to the Instrument via LAN

For connecting over the LAN, you would do have the following:

- Connect the instrument to the LAN physically.
- Configure the Agilent IO Libraries on the remote machine.
- On the user interface, either specify the LAN address or—if a DHCP server is available—enable the DHCP. The DHCP will automatically set up the LAN connection.
- After the connection has been established, the following commands can be used to modify the settings:
  - Enable/disable DHCP with
    `:SYST:COMMunicate:LAN[:SELF]:DHCP`
  - Set the instrument's LAN name with
    `:SYST:COMMunicate:LAN[:SELF]:NAME`
  - Set the instrument's IP address with
    `:SYST:COMMunicate:LAN[:SELF]:ADDRess`
  - Set the instrument's subnet mask with
    `:SYST:COMMunicate:LAN[:SELF]:SMASk`
  - Set the instrument's gateway with
    `:SYST:COMMunicate:LAN[:SELF]:DGATeway`

Configuring the Agilent IO Libraries

Suite 14 of the Agilent IO Libraries does not directly support interfaces with a SICL name of "lan0". When you add a LAN interface, the default SICL name is "inst0". To ensure compatability with current code, it is recommended that you change the SICL name to "lan0".

To set up and configure the interface:

1. Run Agilent Connection Expert and configure your TCPIP instrument according to the instructions provided with the Agilent Connection Expert.
2. Close Agilent Connection Expert.

3 Run IO Config.

The IO Config utility (iocfg32.exe) can be found at (default location):
C:\Program Files\Agilent\IO Libraries Suite\bin

Note that you can also open the IO Config from the Agilent IO Libraries Control icon in the task bar.

4 Edit “inst0” to “lan0”.

Select “TCPIP Lan”, then click *Edit*. This will bring up the list of TCPIP devices already configured. Select the device you need to change, then click *Edit Device*. You can now change the device name to “lan0”.

NOTE You will see a red circle with “!” in the Agilent Connection Expert.

Example for Connecting via LAN

The following code snippet shows how to use the VISA library to connect to the instrument via LAN.

This example queries a device for an identification string and prints the results.

Implementation

```
#include <visa.h>
#include <stdio.h>

void main () {
   ViSession defaultRM, vi;
   char buf [256] = {0};

   /* Open session to the device */
   viOpenDefaultRM (&defaultRM);
   viOpen (defaultRM,
   "TCPIP0::123.123.123.123::lan0::INSTR"
   VI_NULL,VI_NULL, &vi);

   /* Initialize device */
   viPrintf (vi, "*RST\n");

   /* Send an *IDN? string to the device */
   viPrintf (vi, "*IDN?\n");

   /* Read results */
   viScanf (vi, "%t", &buf);

   /* Print results */
   printf ("Instrument identification string: %s\n", buf);

   /* Close session */
   viClose (vi);
```

```
   viClose (defaultRM);
}
```

Connecting to the Instrument via USB

NOTE The control PC must have USB capability for USB connections (Windows NT is not supported).

For connecting over the USB, please refer to the Help delivered with the USB driver.

SCPI Commands Reference

The following sections describe the SCPI Commands available to program the 81133A/81134A remotely. The commands are divided into the following functional blocks:

Command Structure

Each command description has at least some of the following items:

- Full command syntax
- Form
  - Set

    The command can be used to program the instrument.
  - Query

    The command can be used to interrogate the instrument. A question mark (?) is added to the command, the parameters may also change.
- Brief description
- Parameters
- Parameter Suffix

  The suffixes that may follow the parameter.
- Functional Coupling

  Any other commands that are implicitly executed by the command.
- Value Coupling

  Any other parameter that is also changed by the command.
- Range Coupling

  Any other parameter whose valid ranges may be changed by the command.
- *RST value

  The value/state following a *RST command
- Specified Limits
- Short example

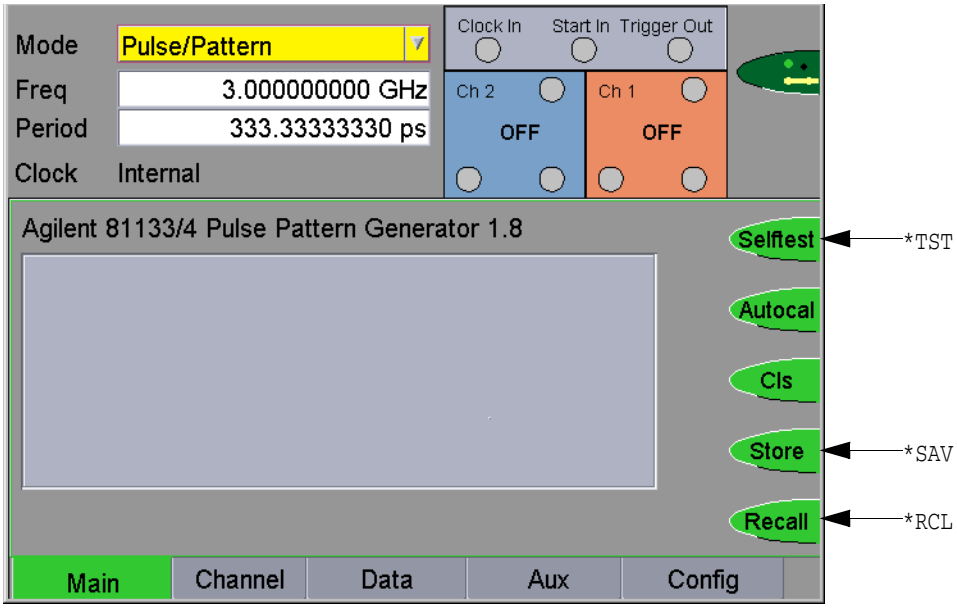
Common Commands

The following table shows the IEEE 488.2 Common Commands available with the Agilent 81133A/81134A Pulse Generator.

Command	Parameter	Description
*CLS	-	Clears the status register
*ESE	<0–255>	Sets the event status register mask
*ESR?	-	Reads the event status register
*IDN?	-	Reads the instrument's identification string
*LRN?	-	Reads a complete instrument setting
*OPC	-	Sets the operation complete bit when all pending actions are complete
*OPT?	-	Reads the installed options
*RCL	<1–9>	Reads a complete instrument setting from memory
*RCL	<0>	Reads the standard settings from the memory. For a list of standard settings, see *"Standard Settings" on page 21*.
*RST	-	Resets the instrument to standard settings. For a list of standard settings, see *"Standard Settings" on page 21*.
*SAV	<1–9>	Saves the complete instrument setting to the memory
*SRE	<0–255>	Sets the service request enable mask
*STB?	-	Reads the status byte
*TST?	-	Executes the instrument's self-test
*WAI	-	Waits until all pending actions are complete

Commands in the User Interface

The following figure shows how the IEEE 488.2 Common commands are implemented in the 81133A/81134A user interface.



Standard Settings

The following table shows the standard settings (Memory 0).

Parameter	SCPI Command	Reset Value	Range
Outputs On/Off	:OUTP[0|1|2] <value>	0 (OFF)	0 | 1| ON | OFF
Output Normal	:OUTP[1|2]:POS <value>	0 (OFF)	0 | 1 | ON | OFF
Output Complement	:OUTP[1|2]:NEG	0 (OFF)	0 | 1 | ON | OFF
Instrument Mode	:FUNC <value>	PATT	PATTern|BURSt, <number>|RBURSt, <number1>, <number2>
Burst	:FUNC BURSt, <value>	1	1 ... 16384
Repeated Burst	:FUNC RBURS	4, 4	For both, 4 ... 16384 in increments of 4
Frequency	:FREQ <value>	15 MHz	15 MHz ... 3.35 GHz
Period	:PER <value>	66.666667ns	0.298507 ps ... 66.666667 ns
Clock Mode	:TRIG:SOUR	Internal	IMMediate, EXTernal, REFerence, IDIRect, EDIRect
Channel Mode	:FUNC:MOD[1|2] <value>	PULSe	PULSe |SQUAre |DATa |PRBS, <numeric>
PRBS Number	:FUNC:MOD[1|2] PRBS, <value>	23 (2^{23} - 1)	5|6|7|8|9|10|11|12|13|14|15|23|31
Freq. Divider	:OUTP[0|1|2]:DIV <value>	1	1, 2, 4, ... 128
Data Signal Mode	:DIG[1|2]:SIGN:FORM <value>	NRZ	R1, RZ, NRZ
Var. Crossover	:DIG[1|2]:SIGN:CROS <value>	50 %	30 ... 70 %
Var. Crossover mode	:DIG[1|2]:SIGN:CROS:STAT <value>	0 (disabled)	0| 1 | OFF | ON
Data Polarity	:DIG[1|2]:SIGN:POL <value>	NORMal	NORMal, INVerted
Pulse Perf.	DIAG:CHAN[1|2]:PPER <value>	NORMal	NORMal|FAST|SMOoth
Delay Control Input	:PM[1|2] <value>	OFF	OFF, ON
Delay Control Input Sensitivity	:PM[1|2]:SENS <value>	25 ps	25 ps | 250 ps
Delay	:DEL[1|2] <value>	0 ns	-5 ns ... +230 ns
Phase	:PHAS[1|2] <value>	0	see Delay
Pulse Width	:WIDT[1|2] <value>	33.333333 ns	100 ps ... (period -100 ps)
Duty Cycle	:DCYC[1|2] <value>	50 %	See Pulse Width
Deskew	:DESK[1|2] <value>	0ps	-10 ns ... +10 ns
Polarity	:POL[1|2] <value>	NORMal	NORMal|COMPlement

Parameter	SCPI Command	Reset Value	Range
Low Level	:VOLT[0|1|2]:LOW <value>	-50 mV	-2.00 V ... +2.95 V
High	:VOLT[0|1|2]:HIGH <value>	50 mV	-1.95 V ... +3.00 V
Offset	:VOLT[0|1|2]:OFFS <value>	0 mV	-1.975 V ... +2.975 V
Amplitude	:VOLT[0|1|2] <value>	100 mV	50 mV ... 2.00 V
Term. Voltage	:VOLT[0|1|2]:TERM <value>	0 mV	-2.00 V ... +3.00 V
Limit to current levels	:VOLT[1|2]:LIM <value>	OFF	OFF, ON
Data Length	:DIG:PATT:LENG <value>	32	32 ... 8192 (in increments of 32)
Clock Input Termination	:TRIG:TERM:STATE <value>	OFF	ON, OFF
Clock Input Term. Voltage	:TRIG:TERM <value>	0 mV	-2.00 ... +3.00 V
Trigger Output Mode	:OUTP0:SOUR <value>	PERiodic	PERiodic, BITStream
Trigger Output Divider	:OUTP0:DIV <value>	1	1, 2, 3, ... (2^{31} - 1)
Trigger Output High	see High Level	50 mV	-1.95 V ... +3.00 V
Trigger Output Low	see Low Level	-50 mV	-2.00 ... +2.95 V
Trigger Output Term. Voltage	see Term. Voltage	0 mV	-2.00 ... +3.00 V
Start Input Start Mode	:ARM:SOUR <value>	IMMediate	IMMediate|MANual|EXTernal
Start Input Term. Voltage	:ARM:TERM <value>	0 mV	-2.00 ... +3.00 V
Start Input Threshold	:ARM:LEV <value>	100 mV	-2.00 ... +3.00 V
Start Input Start On	:ARM:SLOP <value>	POS (rising)	POS, NEG (rising/falling)

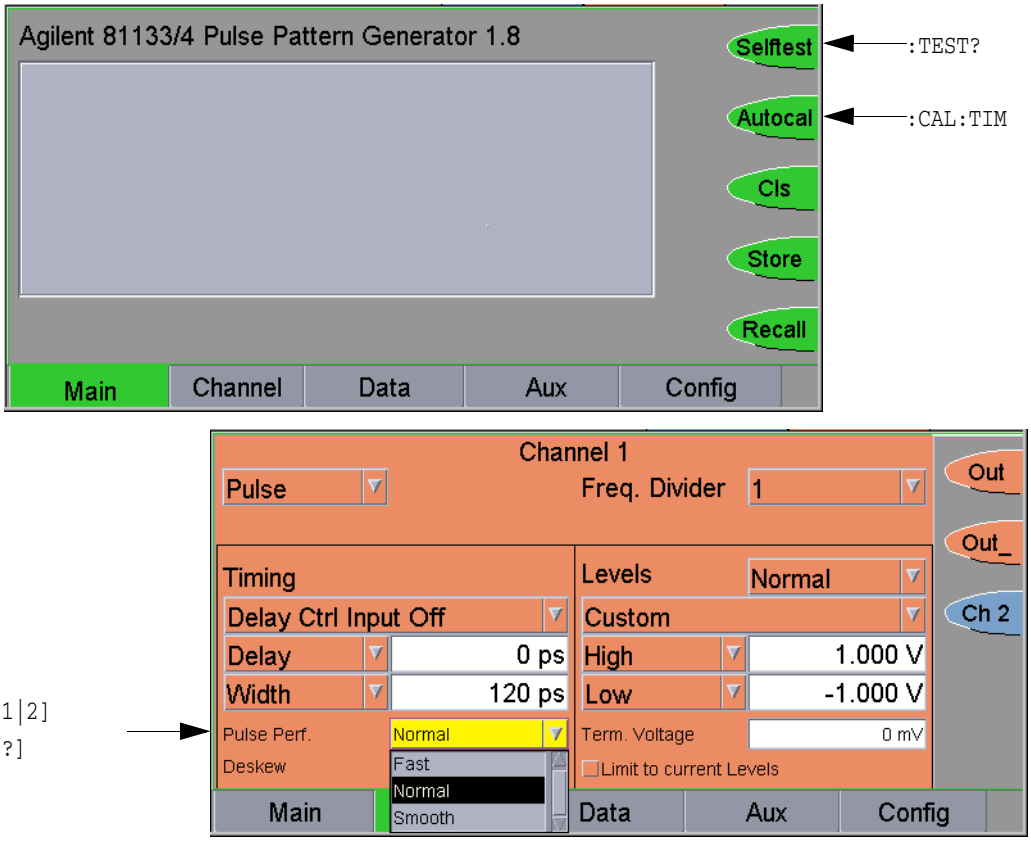
DIAGnostic Commands

The following table shows the Agilent 81133A/81134A Pulse Generator DIAGnostic Commands.

Command	Parameter	Description
:CAL:TIM		Calibrates the timing system of the instrument
:DIAG :CHANnel[1|2] :PPERformance[?]	NORMal | FAST | SMOoth	Sets/reads channel peak performance

Commands in the User Interface

The following figure shows how the DIAGnostic commands are implemented in the 81133A/81134A user interface.



:CAL:TIM

Syntax :CAL:TIM

Form Set

Description Calibrates the timing system of the instrument.

NOTE Execution of this command can take about 15 minutes.

***RST value** –

:DIAG:CHANnel[1|2]:PPERformance

Syntax :DIAG:CHAN[1|2]:PPER[?]

Form Set & Query

Description This command is used to modify the specified transition time of the signal.

For the specified transition time, please refer to the *Technical Specification* delivered on the product CD.

Parameter NORMal|FAST|SMOoth

- Normal

  Produces pulses with the standard transition time specified for the instrument.

  For the specified transition time, please refer to the *Technical Specification* delivered on the product CD.

- Fast

  Reduces the transition time. This leads to a higher slew rate but more overshoot.

- Smooth

  Produces a rounder output pulse, with lower slew rate and less overshoot.

***RST value** Normal

Example Set the Peak Performance to Fast.

```
:DIAG:CHANnel:PPER FAST
```

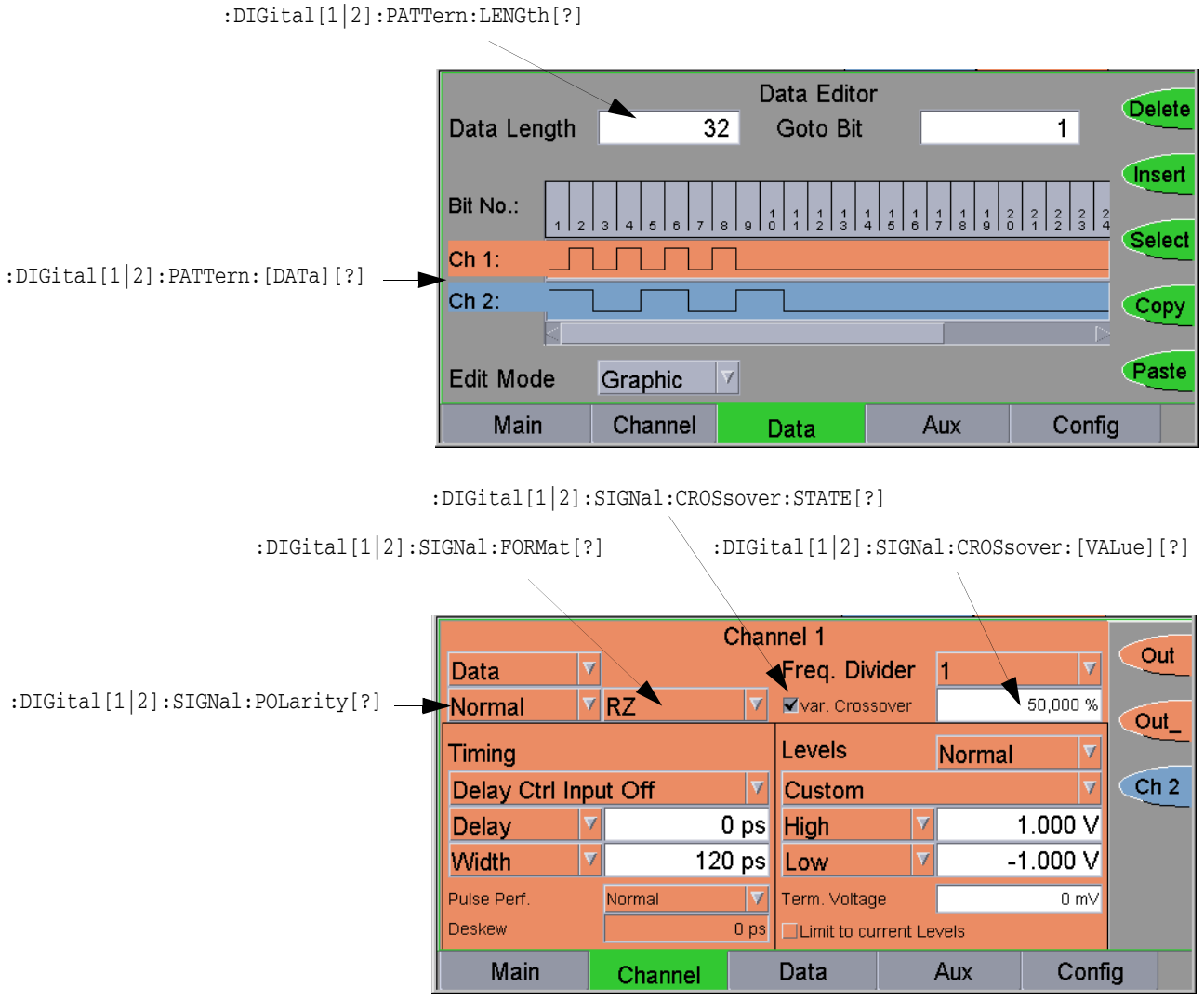
DIGital Commands

The following table shows the Agilent 81133A/81134A Pulse Generator DIGital commands:

Command	Parameter	Description
:DIGital[1|2]		
[:STIMulus]		
:PATTern		
[:DATa][?]	<data>, [HEX | BIN | DUAL]	Sets/reads data in hexadecimal (default), binary or dual format; this command is for data patterns with maximum 8192 bits
:LDATa	<data>	Sets data in hexadecimal format; this command is for data patterns larger than 8192 bits
:LENGth[?]	Numeric	Sets/reads data pattern length in bits (32 ... 8192) in steps of 32
:SIGNal		
:FORMat[?]	RZ | NRZ | R1	Sets/reads the signal mode
:POLarity[?]	NORMal | COMPlement | INVerted	Sets/reads data polarity
:CROSsover :[VALUE][?]	Numeric	Sets/reads crossover
:STATe[?]	ON | OFF | 1 | 0	Switches crossover on/off

Commands in the User Interface

The following figures show how the DIGital commands are implemented in the 81133A/81134A user interface.



:DIGital[1|2][:STIMulus]:PATTern[:DATa]

Syntax :DIG[1|2][:STIM]:PATT[:DAT][?]

Form Set & Query

Description This command is used to set or read the pattern data of one of the channels. The minimum length of these patterns is 32 bits, the maximum length is 8192 bits, the granularity is 32 bits. For patterns larger than 8192 bits, see *":DIGital[1|2][:STIMulus]:PATTern:LDATa" on page 29*.

The data can be written in either hexadecimal, binary or dual format. The query returns the data in hexadecimal format.

For the command, the format is specified by the format parameter; HEX is the default.

- Hexadecimal

  With the hexadecimal format, the characters passed as the data pattern will be interpreted as hexadecimal values.

  The MSB of the first character becomes bit 0 of the data pattern.

- Binary

  With the binary format, the ASCII values of the characters passed are used to build the data pattern.

- Dual

  With the dual format, you can use "0" and "1" to build the data pattern.

Parameter <data>, [HEX | BIN | DUAL]

The <data> is an arbitrary block of program data as defined in IEEE 488.2 7.7.6.2, for example:

```
#181CF1011E, HEX
#          Start of block
1          Length of the length of the data
8          Length of the data (in bytes)
1CF1011E   32 bits of pattern data
HEX        Data in hex format

#232010010010010010101001010101001l0, DUAL
#          Start of block
2          Length of the length of the data
32         Length of the data (in bytes)
010...110  32 bits of pattern data
DUAL       Data in dual format

#14@@@@, BIN
#          Start of block
1          Length of the length of the data
4          Length of the data (in bytes)
@@@@       32 bits of pattern data
BIN        Data in binary format
```

***RST value** 4 bytes with the binary value 00010001

Example The examples above would be sent as follows:

```
:DIG:PATT #181CF1011E[, HEX]
:DIG:PATT?
>#181CF1011E

:DIG:PATT #232010010010010010101001010101001l0, DUAL
:DIG:PATT?
>#1849254AA6

:DIG:PATT #14@@@@, BIN
:DIG:PATT?
>#1840404040
```

:DIGital[1|2][:STIMulus]:PATTern:LDATa

Syntax :DIG[1|2][:STIM]:PATT:LDAT

Form Set

Description This command is used to program long data patterns in hexadecimal format. The minimum length of these patterns is 128 bits, the maximum length is 12 Mbits, the granularity is 128 bits. Patterns generated with this command are subject to various restrictions (see below). See also *":DIGital[1|2][:STIMulus]:PATTern[:DATa]" on page 27*.

Parameter <data>

The <data> is an arbitrary block of hex program data as defined in IEEE 488.2 7.7.6.2, for example:

```
#532768AB03CDAD......
#          Start of block
5          Length of the length of the data
32768      Length of the data
AB03CD...  32768 bytes of data for pattern
           length of 131072 bits
```

Restrictions This command is subject to the following restrictions:

- Only pattern up to 12 Mbits and a granularity of 128 bits are allowed.
- Because of hardware restrictions, a complete 12 Mbit pattern can only be sent at higher frequency ranges:

Range	Maximum Pattern Length
60 – 3360 Mhz	12 Mbit
30 – 60 Mhz	6 Mbit
15 – 30 Mhz	3 Mbit

- The frequency divider of a two-channel instrument also restricts the maximum pattern. The following equation is valid:

$$\text{Max. pattern size} = \frac{\text{Max. pattern size (@ freq)}}{\text{Frequency divider}}$$

- The complete pattern is stored temporarily in the instrument. If there are no restrictions to the maximum pattern length regarding frequency and frequency divider, the complete pattern will always be continually emitted. If there are restrictions to the maximum pattern length, the stored pattern will be emitted up to the maximum pattern length and then repeated from the beginning.
- The channel mode must be data mode.
- The main mode of the instrument must be Pulse/Pattern. Burst and RBurst mode are not allowed.
- The extended pattern is lost by any of the following actions:
  - Modifying the pattern in the pattern editor of the user interface
  - Sending another pattern by :DIG:PATT:DATA or :DIG:PATT:LDAT
  - Changing channel mode or main mode of the instrument
  - Recalling a setting or resetting the instrument
  - Restarting the instrument

  In all these cases, the pattern length is set to the actual data length.

***RST value** –

Example The above example would be sent as:

```
:DIG:PATT:LDAT #532768AB03CDAD......
```

:DIGital[1|2][:STIMulus]:PATTern:LENGth

Syntax `:DIG[1|2][:STIM]:PATT:LENG[?]`

Form Set & Query

Description Defines the length of the data pattern. If the current pattern is longer than the new value for :LENGth, the pattern is truncated. If the current pattern is shorter than the new value for :LENGth, the pattern is lengthened and the new bits are set to '0'.

Parameter Numeric

Valid values are:
32 ... 8192 in steps of 32.

***RST value** 32

Example Define a data pattern length of 64 bits.

```
:DIG:PATT:LENG 64
```

:DIGital[1|2][:STIMulus]:SIGNal:FORMat

Syntax `:DIG[1|2][:STIM]:SIGN:FORM[?]`

Form Set & Query

Description This command is used to program the signal format for data and PRBS signals:

Parameter RZ|NRZ|R1

- RZ
  Return to Zero. A pulse of 50% duty cycle is generated for each 1.
- NRZ
  Non-Return to Zero. A pulse of 100% duty cycle is generated for each 1.
- R1
  Return to One. A pulse of 100% duty cycle is generated for each 0.

***RST value** NRZ

Example Set data format to R1.

```
:DIG:SIGN:FORM R1
```

:DIGital[1|2][:STIMulus]:SIGNal:POLarity

Syntax `:DIG[1|2][:STIM]:SIGN:POL[?]`

Form Set & Query

Description This command is used to program the data polarity for Data and PRBS signals. The 32-bit data pattern is logically inverted, that is, 1 s are replaced with 0 s and vice versa.

NOTE This is not the same as the `[:SOURce][:PULSe]:POLarity[1|2]` command, which physically inverts the signal by swapping the OUTPUT and $\overline{\text{OUTPUT}}$ signals.

Parameter NORMal|COMPlement|INVerted

INVerted are synonyms (INVerted is included for backwards compatibility).

***RST value** NORMal

Example Logically invert the 32-bit data.

```
:DIG:SIGN:POL INV
```

:DIGital[1|2][:STIMulus]:SIGNal: CROSsover:[VALue]

Syntax `:DIG[1|2][:STIM]:SIGN:CROS[?]`

Form Set & Query

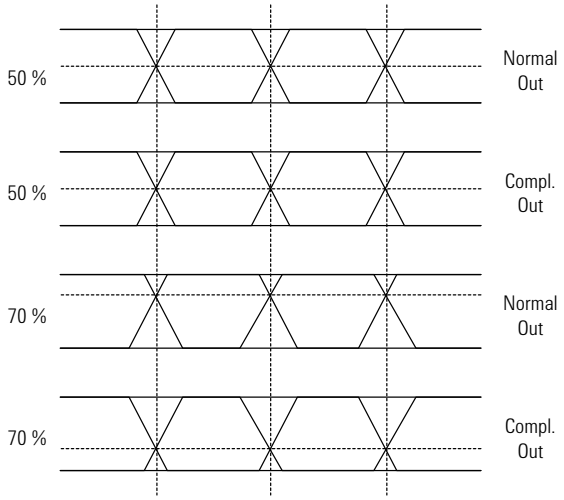
Description If variable crossover mode is enabled, this command specifies a value to adjust the crossover point of the NRZ signal in PRBS or data mode, individually for each channel.

To enable the variable crossover mode, use *":DIGital[1|2][:STIMulus]:SIGNal: CROSsover:[STATe]" on page 34*.

The variable crossover is used to artificially close the eye pattern, which simulates distortion.

NOTE This parameter has no influence if the delay control input for the channel is switched on (`:PM[1|2] ON`).

The figure below shows the normal and complement output with the crossover point set to 50% and 70% respectively.



Parameter Numeric values (in %) in the range of 20 ... 80.

***RST value** 50

Example Set the variable crossover point to 70%.

```
:DIG:SIGN:CROS 70
```

:DIGital[1|2][:STIMulus]:SIGNal: CROSsover:[STATe]

Syntax `:DIG[1|2][:STIM]:SIGN:CROS:STAT[?]`

Form Set & Query

Description For each channel, the crossover mode of the NRZ signal in PRBS or data pattern mode can be enabled. This is used to artificially close the eye pattern, simulating distortion.

If you enabled the variable crossover mode, specify the variable crossover point with *":DIGital[1|2][:STIMulus]:SIGNal: CROSsover:[VALue]" on page 32*.

Parameter ON|OFF|1|0

***RST value** OFF|0

Example Enable the variable crossover mode.

```
:DIG:SIGN:CROS:STAT ON
```

DISPlay Commands

The following table shows the Agilent 81133A/81134A Pulse Generator DISPlay commands.

Command	Parameter	Description
:DISPlay [:WINDow] [:STATe][?]	ON | OFF | 1 | 0	Sets/reads front panel display state

:DISPlay[:WINDow][:STATe]

Syntax :DISP[:WIND][:STAT][?]

Form Set & Query

Description This command is used to turn the front panel display on and off. Switching off the display improves the programming speed of the instrument.

NOTE The display is switched back on if a key on the instrument is pressed. The command *RST switches the display back on. Use :SYSTem:PRESet to perform a *RST without switching the display back on.

Parameter ON|OFF|1|0

***RST value** ON

Example Switch off the front panel display.

```
:DISP OFF
```

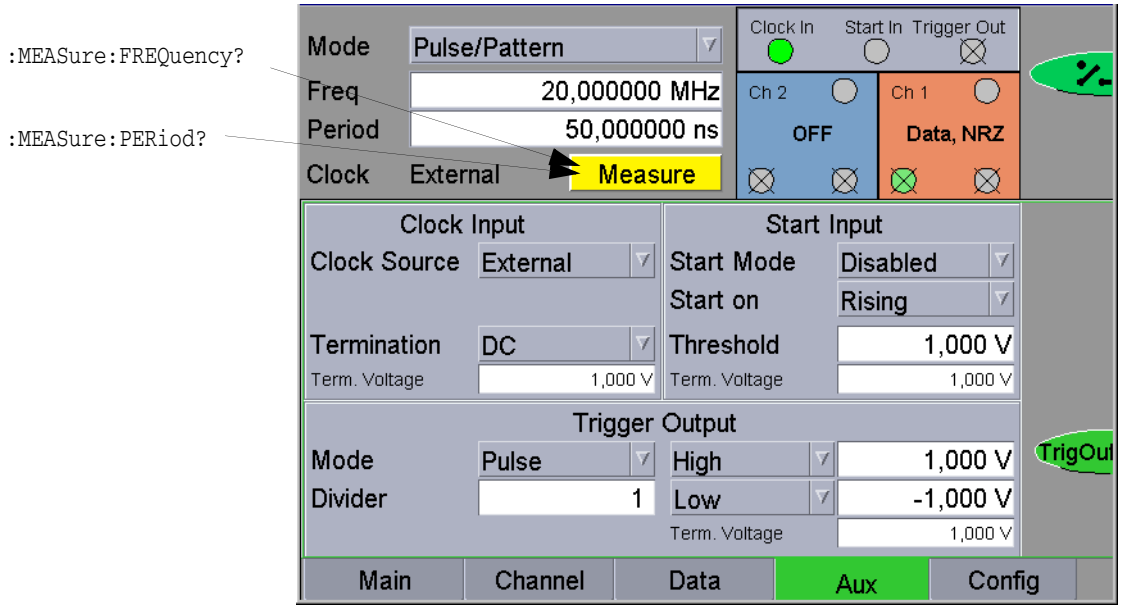
MEASure Commands

The following table shows the Agilent 81133A/81134A Pulse Generator MEASure commands:

Command	Parameter	Description
:MEASure		
:FREQuency?		Read time base frequency
:PERiod?		Read time base period

Commands in the User Interface

The following figure shows how the MEASure commands are implemented in the 81133A/81134A user interface.



:MEASure:FREQuency?

Syntax :MEAS:FREQ?

Form Query

Description This command is used to measure the operating frequency of the instrument.

In internal mode (:TRIGger:SOURce IMMediate) the frequency returned is the measured internal clock frequency (not the programmed value).

In external mode (:TRIGger:SOURce EXTernal) the frequency returned is that measured at the Clock Input connector. If an invalid signal, or no signal, is present at the Clock Input connector, a value of zero is returned.

The query does not return a value immediately, but waits for the internal frequency counter to complete its next measurement cycle. This can take about half a second.

NOTE The instrument is stopped when this command is executed. Thus, during the measurement, no signals will be output.

NOTE When working in an automated test system, if the clock frequency is known, it is better to set it directly instead of reading it from the instrument. This is because:

- This method is faster since it eliminates the measurement time.
- The instrument is not stopped.

Parameter –

***RST value** –

Example :MEAS:FREQ?

:MEASure:PERiod?

Syntax :MEAS:PER?

Form Query

Description This command is used to read the operating period of the instrument.

In internal mode (:TRIGger:SOURce IMMediate) the period returned is the internal clock period.

In external mode (:TRIGger:SOURce EXTernal) the period returned is that measured at the Clock Input connector. If an invalid signal, or no signal, is present at the Clock Input connector, a value of zero is returned.

The query does not return a value immediately, as it waits for the internal frequency counter to complete its next measurement cycle. This can take about half a second.

NOTE The instrument is stopped when this command is executed. Thus, during the measurement, no signals will be output.

NOTE When working in an automated test system, if the clock frequency is known, it is better to set it directly instead of reading it from the instrument. This is because:

- This method is faster since it eliminates the measurement time.
- The instrument is not stopped.

Parameter –

***RST value** –

Example :MEAS:PER?

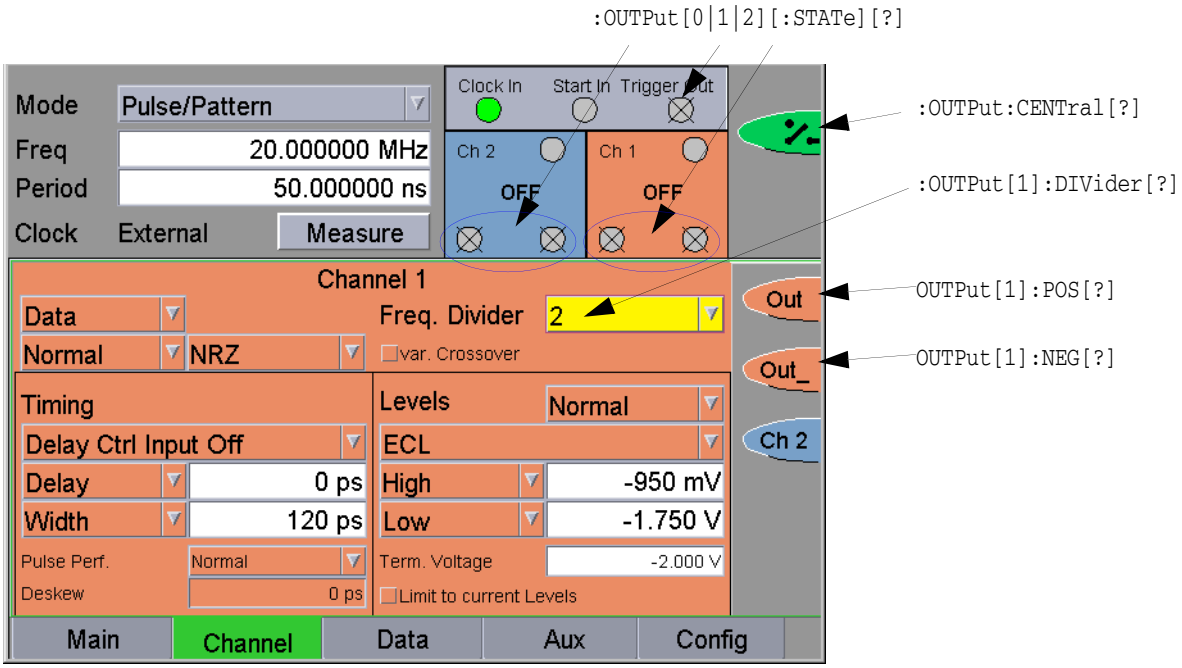
OUTPut Commands

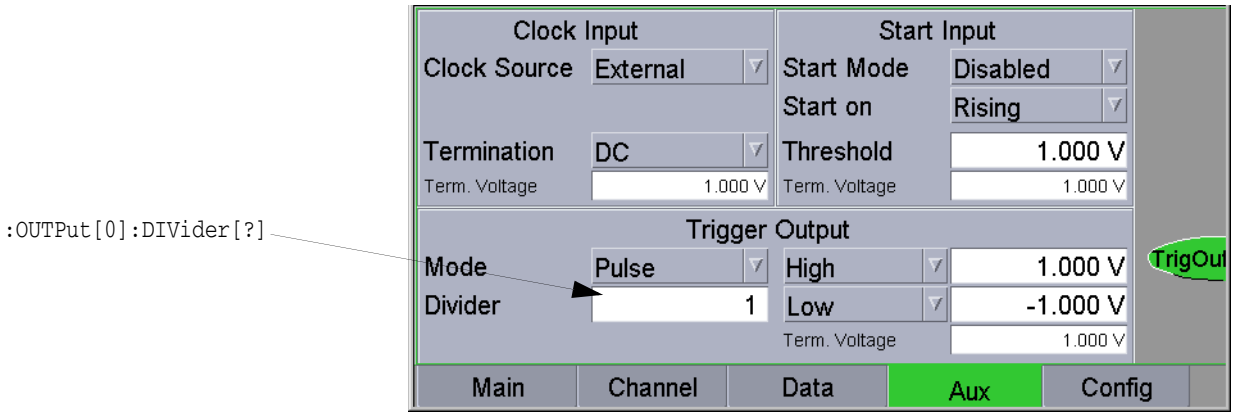
The following table shows the Agilent 81133A/81134A Pulse Generator OUTPut commands.

Command	Parameter	Description
:OUTPut		
[0|1|2]:DIVider[?]	Numeric | MIN | MAX	Set/read channel frequency divider
[0]:SOURce[?]	PERiodic | BITStream	Set/read trigger source mode
[0|1|2][:STATe][?]	ON | OFF | 1 | 0	Set/read channel outputs on and off
[1|2]:NEG[?]	ON | OFF | 1 | 0	Set/read negative channel output on and off
[1|2]:POS[?]	ON | OFF | 1 | 0	Set/read positive channel output on and off
:CENTral[?]	ON | OFF | 1 | 0	Set/read central output settings

Commands in the User Interface

The following figures show how the DIAGnostic commands are implemented in the 81133A/81134A user interface.





:OUTPut[0|1|2]:DIVider

Syntax `:OUTP[0|1|2]:DIV[?]`

Form Set & Query

Description This command is used to program the frequency divider parameters of the trigger output (0) and the channel outputs (1, 2).

The trigger output frequency is divided only when the trigger output is in *Pulse* mode (`:OUTPut0:SOURce PERiodic`).

You can program the divider in *Data* mode (`:OUTPut0:SOURce BITstream`) but it will have no effect until you select the trigger output to pulse mode.

The channel output frequency is divided in square and pulse pattern mode only (`[SOURce]:FUNCtion:MODe[1|2] SQUare|PULSe`).

You can program the divider in data and PRBS pattern mode (`[SOURce]:FUNCtion:MODe[1|2] DATa|PRBS`), but it will have no effect until you select the square or pulse pattern mode.

Parameter Numeric|MIN|MAX

***RST value** 1

Specified Limits For trigger output (channel 0): 1 ... $2^{31} - 1$

For channels 1 and 2: 1, 2, 4, 8, 16, 32, 64, 128

Example Set Trigger Output Divider to 8.

```
:OUTP0:DIV 8
```

:OUTPut0:SOURce

Syntax `:OUTP0:SOUR[?]`

Form Set & Query

Description This command programs the trigger output source mode.

- PERiodic

  This corresponds to *Pulse* mode on the front panel.

  The trigger source is the internal clock, and a trigger pulse is generated every clock period, unless the divider parameter has been set to a value other than 1. The trigger signal always has 50% nominal duty cycle.

- BITStream

  This corresponds to the *Data* mode on the front panel.

  If the clock source is external, the trigger is always synchronized to the clock with a fixed delay (± a few picoseconds over the frequency range). PERiodic means that a trigger pulse is generated for every *X* clocks, BITStream means that *X* is set to the data length.

  The trigger divider does not take the frequency divider of the channels into account. For a frequency divider of *n*, *n* trigger pulses are generated for each data packet, starting with the first edge of bit 0 of the data packet.

  To get one trigger pulse per data packet when the channel divider factor is not equal to 1, the trigger mode must set to *Trigger on pulse* and the divider to n x X, where n is the frequency divider and X is the data length. For example, if the data length = 32 bits and the frequency divider of channel 1 = 2, the frequency divider of the trigger output has to be 64.

Parameter PERiodic|BITStream

***RST value** PERiodic

Example Synchronize the trigger output signal to the data.

```
:OUTP0:SOUR BITS
```

:OUTPut[0|1|2][:STATe]

Syntax `:OUTP[0|1|2][:STAT][?]`

Form Set & Query

Description Switches the trigger output and channel outputs on or off, where 0 is the trigger output.

For the two channel outputs, both OUTPUT and $\overline{\text{OUTPUT}}$ are switched simultaneously. In query form, OFF is returned only if both OUTPUT and $\overline{\text{OUTPUT}}$ are off. They can be controlled separately from the front panel, or by adding `:POS` or `:NEG` to the command.

Parameter ON|OFF|1|0

***RST value** OFF

Example Switch on the channel 1 outputs.

```
:OUTP1 ON
```

:OUTPut[1|2]:NEG

Syntax `:OUTP[1|2]:NEG[?]`

Form Set & Query

Description Switches the specified channel $\overline{\text{OUTPUT}}$ on or off.

Parameter ON|OFF|1|0

***RST value** OFF

Example Switch off the channel 1 $\overline{\text{OUTPUT}}$.

```
:OUTP1:NEG OFF
```

:OUTPut[1|2]:POS

Syntax `:OUTP[1|2]:POS[?]`

Form Set & Query

Description Switches the specified channel OUTPUT on or off.

Parameter ON|OFF|1|0

***RST value** OFF

Example Switch off the channel 1 OUTPUT.

```
:OUTP1:POS OFF
```

:OUTPut:CENTral

Syntax `:OUTP:CENT[?]`

Form Set & Query

Description Sets or reads the central output settings.

The OFF command forces all outputs (trigger output and channel outputs) to be switched off, the ON command switches on every output that is set to on by the other `:OUTPut` commands.

Parameter ON|OFF|1|0

***RST value** 1

Example Switches off all output channels.

```
:OUTP:CENT OFF
```

SOURce Commands

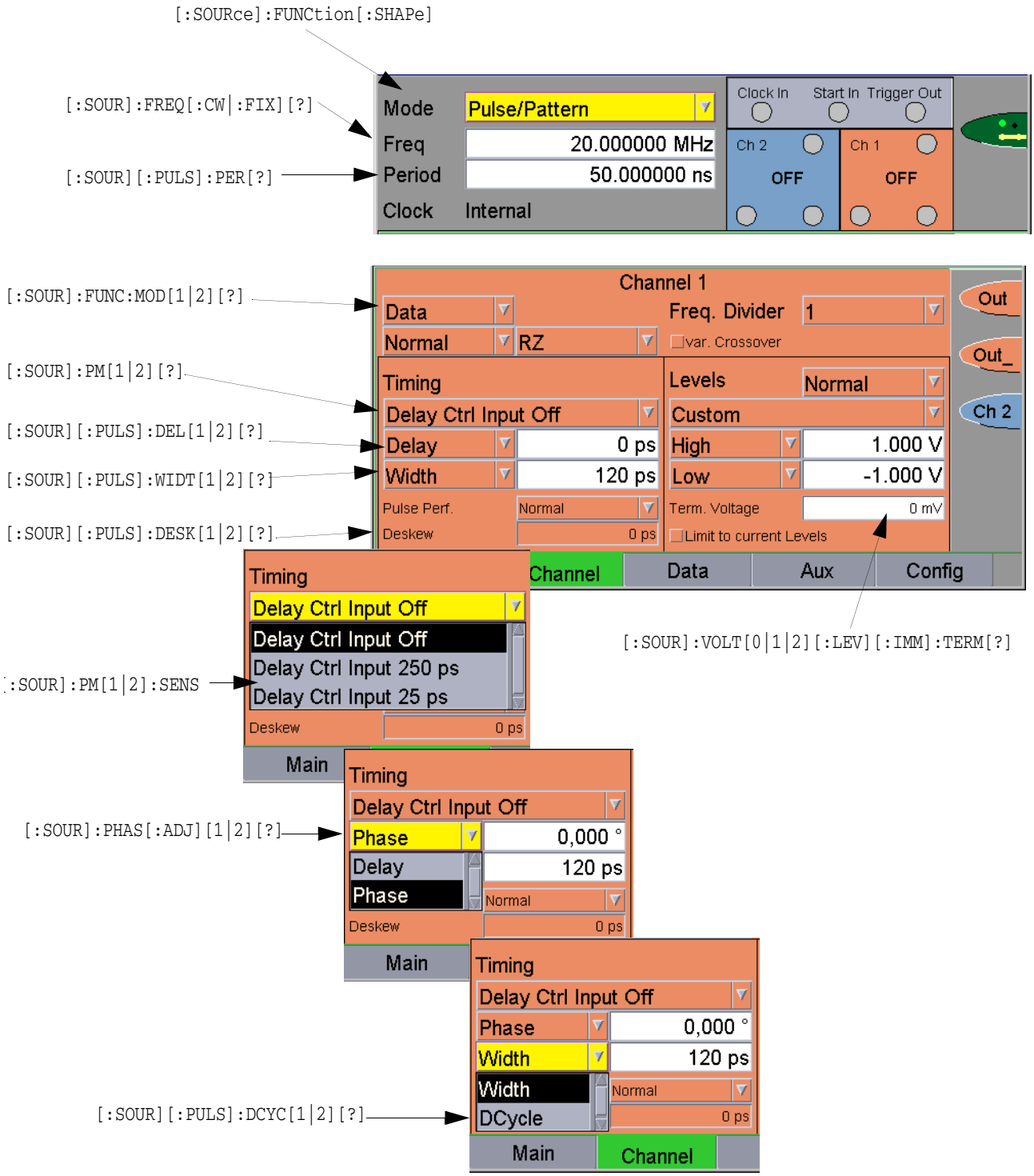
The following table shows the Agilent 81133A/81134A Pulse Generator SOURce commands:

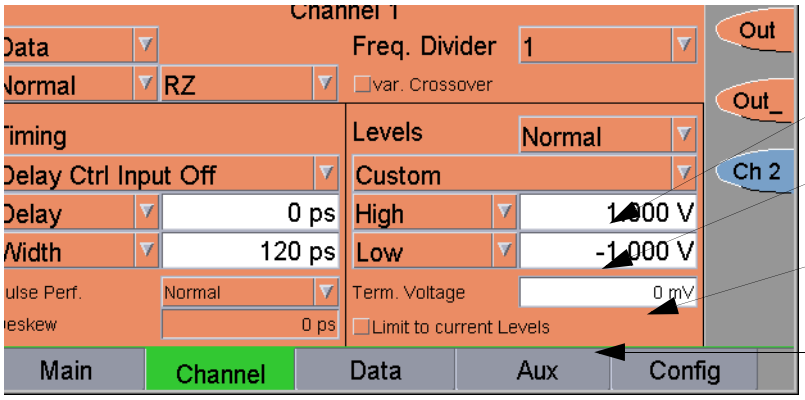
Command	Parameter	Description
[:SOURce]		
:FUNCtion		
[:SHAPe][?]	PATTern| BURSt, <numeric>| RBURSt, <numeric>, <numeric>	Sets/reads instrument mode
:MODE[1|2][?]	PULSe|SQUare| DATa|PRBS <numeric>	Sets instrument main mode
:FREQuency		
[:CW|:FIXed][?]	Numeric [GHz|MHz|kHz|Hz] | MIN|MAX	Sets/reads internal clock frequency
:PHASe		
[:ADJ][1|2][?]	Numeric|MIN|MAX	Sets/reads channel phase
[:PULSe]		
:DCYCle[1|2][?]	Numeric|MIN|MAX	Sets/reads channel duty cycle
:DELay[1|2][?]	Numeric [ps|ns|us|ms|s] | MIN|MAX	Sets/reads channel delay
:DESKew[1|2][?]	Numeric [ps|ns|us|ms|s]| MIN|MAX	Sets/reads channel deskew
:DHOLd[1|2][?]	DELay|PHASe	Holds Delay|Phase fixed with varying frequency
:PERiod[?]	Numeric [ps|ns|us|ms|s] | MIN|MAX	Sets/reads internal clock period
:POLarity[1|2][?]	NORMal| COMPlement| INVerted	Sets/reads channel polarity
:WIDTh[1|2][?]	Numeric [ps|ns|us|ms|s] | MIN|MAX	Sets/reads channel width

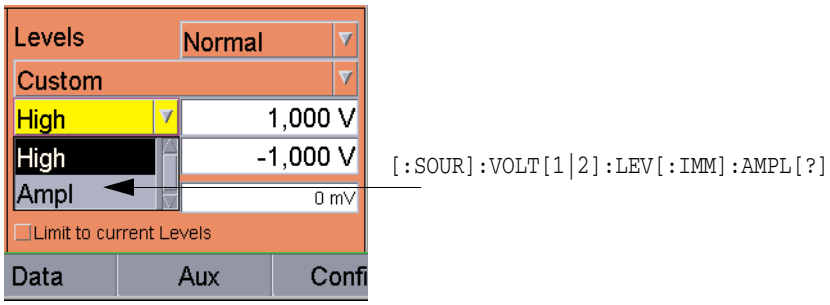
Command	Parameter	Description
PM[1|2][?]	OFF|ON	Sets/reads jitter modulation
:SENSitivity	25ps|250ps	Sets jitter modulation sensitivity
:VOLTage		
[0|1|2][:LEVel]		
[:IMMediate]		
[:AMPLitude][?]	Numeric [uV|mV|V] |MIN|MAX	Sets/reads channel amplitude
:OFFSet[?]	Numeric [uV|mV|V] |MIN|MAX	Sets/reads channel offset
:HIGH[?]	Numeric [uV|mV|V] |MIN|MAX	Sets/reads channel high-level
:LOW[?]	Numeric [uV|mV|V] |MIN|MAX	Sets/reads channel low-level
:TERM[?]	Numeric [uV|mV|V]	Sets/reads termination voltage
[1|2]:LIMit		
[:AMPLitude]?		Reads channel amplitude limit
:OFFSet?		Reads channel offset limit
:HIGH?		Reads channel high-level limit
:LOW?		Reads channel low-level limit
:STATe[?]	ON|OFF|1|0	Sets/reads limited output mode on and off

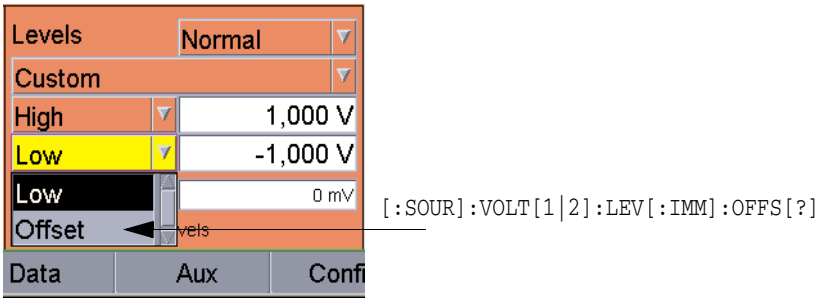
Commands in the User Interface

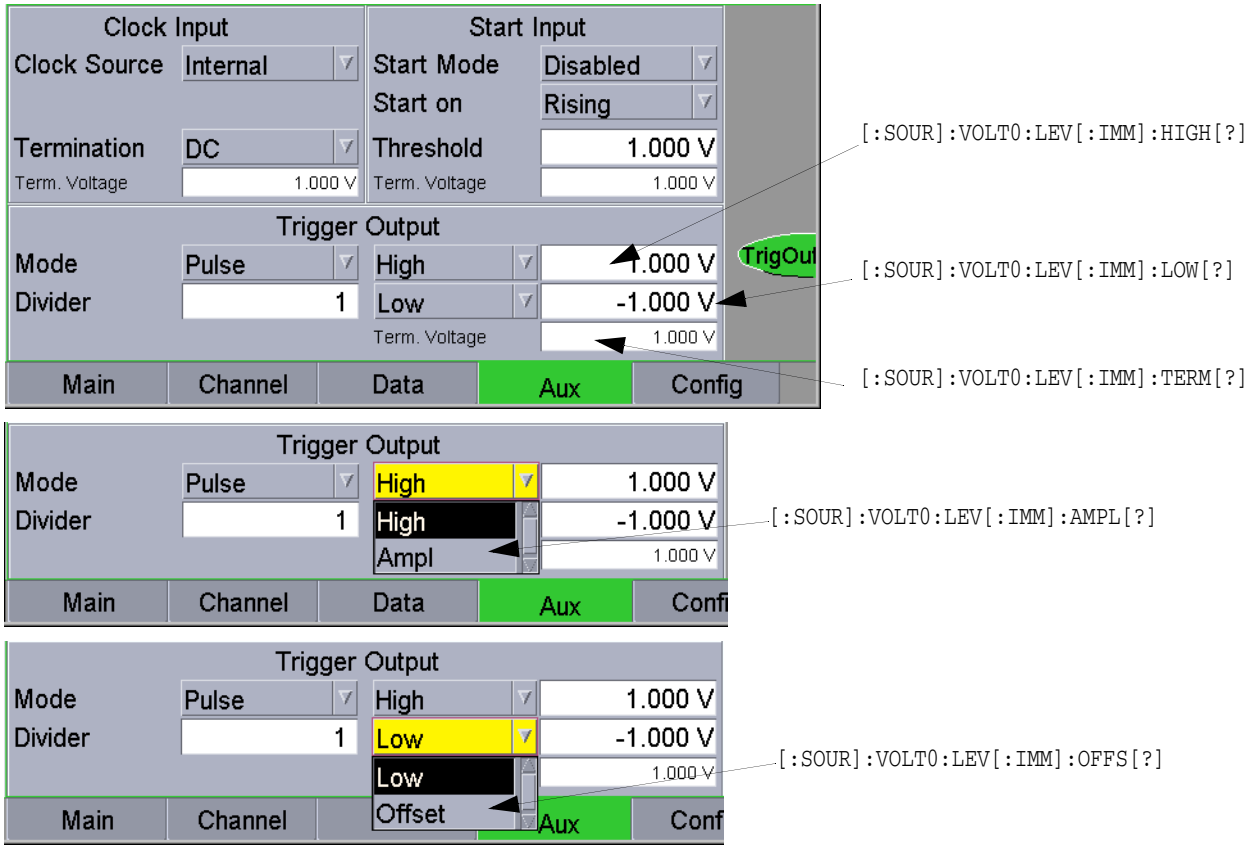
The following figures show how the DIAGnostic commands are implemented in the 81133A/81134A user interface.




Channel 1
Data
Freq. Divider
1
Normal
RZ
var. Crossover
Timing
Levels
Normal
Delay Ctrl Input Off
Custom
Delay
0 ps
High
Width
120 ps
Low
ulse Perf.
Normal
Term. Voltage
0 mV
eskew
0 ps
Limit to current Levels
Out
Out_
Ch 2
Main
Channel
Data
Aux
Config
[:SOUR]:VOLT[1|2]:LEV[:IMM]:HIGH[?]
[:SOUR]:VOLT[1|2]:LEV[:IMM]:LOW[?]
[:SOUR]:VOLT[1|2]:LEV[:IMM]:TERM[?]
[:SOUR]:VOLT[1|2]:LIM:STAT[?]


Levels
Normal
Custom
High
1,000 V
High
-1,000 V
Ampl
0 mV
Limit to current Levels
Data
Aux
Confi
[:SOUR]:VOLT[1|2]:LEV[:IMM]:AMPL[?]


Levels
Normal
Custom
High
1,000 V
Low
-1,000 V
Low
0 mV
Offset
vels
Data
Aux
Confi
[:SOUR]:VOLT[1|2]:LEV[:IMM]:OFFS[?]


Clock Input
Clock Source
Internal
Termination
DC
Term. Voltage
1.000 V
Start Input
Start Mode
Disabled
Start on
Rising
Threshold
1.000 V
Term. Voltage
1.000 V
Trigger Output
Mode
Pulse
Divider
1
High
1.000 V
Low
-1.000 V
Term. Voltage
1.000 V
TrigOut
Main
Channel
Data
Aux
Config
[:SOUR]:VOLT0:LEV[:IMM]:HIGH[?]
[:SOUR]:VOLT0:LEV[:IMM]:LOW[?]
[:SOUR]:VOLT0:LEV[:IMM]:TERM[?]
Trigger Output
Mode
Pulse
Divider
1
High
1.000 V
High
-1.000 V
Ampl
1.000 V
Main
Channel
Data
Aux
[:SOUR]:VOLT0:LEV[:IMM]:AMPL[?]
Trigger Output
Mode
Pulse
Divider
1
High
1.000 V
Low
-1.000 V
Low
1.000 V
Offset
Main
Channel
Aux
[:SOUR]:VOLT0:LEV[:IMM]:OFFS[?]

[:SOURce]:FUNCtion[:SHAPe]

Syntax `[:SOUR]:FUNC[:SHAP][?]`

Form Set & Query

Description Defines the main mode of the signal to be generated (pulse/pattern, burst, or repetitive burst mode).

Parameter PATTern|BURSt, <numeric>|RBURSt, <numeric>, <numeric>

- PATTern

  In this mode, each channel can be set independently to generate:

  – Square waves of fixed width

  – Pulses with selectable width or duty cycle

  – Data in either RZ, R1 or NRZ format

  – Pseudo random bit stream (PRBS) polynomials

  To generate these signals, use *"[:SOURce]:FUNCtion:MODe[1|2]" on page 51*.

- BURSt, <number of repeated data>

  This mode enables you to generate a burst consisting of data repeated n times followed by continuous zero data.

  *RST value: 1

- RBURSt, <number of repeated data>, <p>

  This mode enables you to generate a repeated burst consisting of data repeated n times. A pause of zeros is inserted between two successive bursts. The pause of zeros is calculated by:

  Length of the pause = Burst Length × p

  *RST values: 4, 4

***RST value** PATT

Example Generate a burst of data repeated 5 times:

```
:FUNC BURSt, 5
```

[:SOURce]:FUNCtion:MODe[1|2]

Syntax `[:SOUR]:FUNC:MOD[1|2][?]`

Form Set & Query

Description Use this command to set the pattern mode for each channel. The pattern modes specify pulses, clocks, data patterns or PRBS signals.

Parameter PULSe | SQUare | DATa | PRBS, <numeric>

- SQUare

  Generates a square wave (clock) of fixed width (50% duty cycle). The frequency of the square wave can optionally be divided by 1, 2, 4, ..., 128 with `:OUTPut[1|2]:DIVider`.

- PULSe

  Generates pulses with selectable width or duty cycle. The frequency of the pulses can optionally be divided by 1, 2, 4, ..., 128 with `:OUTPut[1|2]:DIVider`.

- DATa

  Generates data in either RZ, R1 or NRZ format as specified with `:DIG[1|2][:STIM]:SIGN:FORM[?]`.

  In RZ and R1 mode, the pulse width can be set. Set the pulse width with `[:SOUR][:PULS]:WIDT[1|2][?]`.

  The frequency of the data can optionally be divided by 1, 2, 4, ..., 128 with `:OUTPut[1|2]:DIVider`.

- PRBS, <numeric>

  Generates a PRBS polynomial of selectable type in either RZ, R1 or NRZ format. In RZ and R1 mode, the pulse width can be set. Set the pulse width with `[:SOUR][:PULS]:WIDT[1|2][?]`.

  The frequency of the PRBS signals can optionally be divided by 1, 2, 4, ..., 128 with `:OUTPut[1|2]:DIVider`.

  Valid values are: 2^5-1 ... $2^{31}-1$

***RST value** PULSe

Example Generate a PRBS signal of 2^5-1 on channel 1:

1. Set the Pulse/Pattern mode:

   ```
   [:SOUR]:FUNC[:SHAP] PATT
   ```

2. Set the PRBS signal:

   ```
   [:SOUR]:FUNC:MOD[1|2] PRBS, 31
   ```

[:SOURce]:FREQuency[:CW|:FIXed]

Syntax `[:SOUR]:FREQ[:CW|:FIX][?]`

Form Set & Query

Description This command programs the internal clock frequency, and also selects the internal clock as time base if it is not already selected.

Parameter Numeric [GHz|MHz|kHz|Hz] | MIN|MAX

Value coupling Period = 1 / Frequency

***RST value** 15.0E6 Hz

Specified limits 15E6 ... 3.35E9 Hz, with overclocking up to 3.35E9 Hz

Example Select the clock with frequency 1.2 GHz.

```
:FREQ 1.2GHz
```

[:SOURce]:PHASe[:ADJ][1|2]

Syntax `[:SOUR]:PHAS[:ADJ][1|2][?]`

Form Set & Query

Description This command programs the pulse phase for a channel.

Parameter Numeric | MIN|MAX

Parameter Suffix DEG or RAD. A parameter without suffix is interpreted as degrees.

Functional coupling Programming the pulse phase also executes `[:SOURce][:PULSe]:HOLD PHASe` so that the pulse phase is held constant when the signal frequency is changed.

Value coupling Delay = (Phase / 360) * Period

Range coupling Deskew

***RST value** 0.0

Specified limits –6000° ... +279000°, constrained by delay and period limits.

Absolute limits –6000° ... +279000°, constrained by delay and period limits.

Example Set channel 1 phase delay to –180°.

```
:PHAS1 -180
```

[:SOURce][:PULSe]:DCYCle[1|2]

Syntax `[:SOUR][:PULS]:DCYC[1|2][?]`

Form Set & Query

Description This command programs the duty cycle for a channel.

NOTE The duty cycle cannot be set:

- In direct mode. To query the clock source, see *":TRIGger:SOURce" on page 87*.
- If signal mode is set to NRZ. To query the signal mode, see *":DIGital[1|2][:STIMulus]:SIGNal:FORMat" on page 31*.

Parameter Numeric|MIN|MAX

Functional coupling Programming the pulse duty cycle also executes `[:SOURce][:PULSe]:HOLD DCYCLE` so that the pulse duty cycle is held constant when the signal frequency is changed.

Value coupling Width = (duty cycle / 100) * Period

Range coupling Frequency, Period

***RST value** 50% (derived from WIDth and PERiod)

NOTE The DCYCle command holds the PERiod and WIDth values in proportion (if one value is increased 50 %, the other value is also increased 50 %). Its limits are therefore dependent on the limits of PERiod and WIDth.

Example Set channel 1 duty cycle to 66%.

```
:DCYC1 66
```

[:SOURce][:PULSe]:DELay[1|2]

Syntax `[:SOUR][:PULS]:DEL[1|2][?]`

Form Set & Query

Description This command programs the pulse delay for a channel.

Parameter Numeric [ps|ns|us|ms|s]|MIN|MAX

Functional coupling Programming the pulse delay also executes the `[:SOURce][:PULSe]:DHOLD DELay`s so that the pulse delay is held constant when the signal frequency is changed.

Value coupling Phase = (Delay / Period) * 360

Range coupling Deskew

***RST value** 0.0

Specified limits –5 ns ... + 230 ns

Absolute limits –5 ns <= <*value in absolute delay time*> + <*Deskew*> <= 230 ns

Example Set Channel 1 Delay to 500 ps.

```
:DEL1 500PS
```

[:SOURce][:PULSe]:DESKew[1|2]

Syntax `[:SOUR][:PULS]:DESK[1|2][?]`

Form Set & Query

Description This command programs the deskew for a channel. The deskew allows you to move the zero-point of the delay (and phase) parameter by ± 10 ns.

The final delay at the output is Delay + Deskew.

Parameter Numeric [ps|ns|us|ms|s]|MIN|MAX

Range coupling Delay, Phase

***RST value** 0.0

Specified limits –10E–9 ... 10E–9, but deskew and delay must be within the delay limits.

Absolute limits –10E–9 ... 10E–9

Example Set Channel 1 deskew to –155 ps.

```
:DESK1 -155PS
```

[:SOURce][:PULSe]:DHOLd[1|2]

Syntax `[:SOUR][:PULS]:DHOL[1|2][?]`

Form Set & Query

Description Defines whether the pulse delay or the pulse phase of a channel is held constant when the signal frequency is changed.

Parameter DELay|PHASe

***RST value** DELay

Example Hold Channel 1 Delay fixed when frequency varies.

```
:DHOL1 DEL
```

[:SOURce][:PULSe]:PERiod

Syntax `[:SOUR][:PULS]:PER[?]`

Form Set & Query

Description This command programs the internal clock period, and also selects the internal clock time base if it has not already been selected.

Parameter Numeric [ps|ns|us|ms|s]|MIN|MAX

Functional coupling Programming the signal period, or frequency, also executes `:TRIGger:SOURce IMMediate` to select the internal clock.

Value coupling Frequency = 1 / Period

Range coupling Width, Dutycycle, Phase and Pulse/Data mode selection.

***RST value** 66.6 ns

Specified limits 299E–12 sec ... 66.6E–9 sec

Instrument limits 297.61905E–12 sec ... 66.66667E–9 sec

Example Select internal clock with period 750 ps.

```
:PER 750PS
```

[:SOURce][:PULSe]:POLarity[1|2]

Syntax `[:SOUR][:PULS]:POL[1|2][?]`

Form Set & Query

Description This command programs the output polarity of a channel.

Parameter NORMal|COMPlement|INVerted

COMPlement and INVerted are synonyms (INVerted is included for backwards compatibility).

***RST value** NORMal

Example Invert the Channel 1 outputs.

```
:POL1 INV
```

NOTE This is not the same as the `:DIGital[1|2][:STIMulus]:SIGNal:POLarity` command, which logically inverts the 32-bit data on the channels by swapping 1s with 0s and vice-versa.

[:SOURce][:PULSe]:WIDTh[1|2]

Syntax `[:SOUR][:PULS]:WIDT[1|2][?]`

Form Set & Query

Description Programs the pulse width for a channel.

Parameter Numeric [ps|ns|us|ms|s]|MIN|MAX

Functional coupling Programming the pulse width also executes `[:SOURce][:PULSe]:HOLD WIDTh` so that the pulse width is held constant when the signal frequency is changed.

Value coupling Dutycycle = (Width / Period) * 100

Range coupling Frequency, Period

***RST value** 50% of Period

Specified limits 100E–12 ... (Period – 100E–12) sec

Absolute limits 100 ps <= *<value>* <= *<Period value>* –100 ps && *<value>* < 10 ns

Example Set Channel 1 pulse width to 1 ns.

```
:WIDT1 1NS
```

[:SOURce]:PM[1|2]

Syntax `[:SOUR]:PM[1|2][?]`

Form Set & Query

Description Enables the jitter modulation.

Parameter OFF|ON

***RST value** OFF

Example Enable the jitter modulation on channel 1.

```
PM1 ON
```

[:SOURce]:PM[1|2]:SENSitivity

Syntax `[:SOUR]:PM[1|2]:SENS`

Form Set

Description Sets jitter modulation sensitivity.

You now have to apply an external source (–0.5 V ... 0.5 V max) for jitter modulation to the Delay Control Input at the instrument's front panel:

- If you apply a source of 500 mV, the signal delay will be increased by 250 ps/25 ps.
- If you apply a source of –500 mV, the signal delay will be decreased by 250 ps/25 ps.

Between –500 mV and +500 mV, the signal delay increases/decreases lineally to the Delay Control Input, for example, a source of +200 mV results in a delay of 250 ps/V * 200 mV = 50 ps.

Parameter 25ps|250ps

Jitter modulation is turned on with fixed sensitivity of 50 ps/V or 500 ps/V.

***RST value** 25 ps

Example Set the delay control input to 25 ps.

```
[:SOUR]:PM[1|2]:SENS 25ps
```

[:SOURce]:VOLTage[0|1|2][:LEVel] [:IMMediate][:AMPLitude]

Syntax `[:SOUR]:VOLT[0|1|2][:LEV][:IMM][:AMPL][?]`

Form Set & Query

Description Programs the amplitude of the output signal for the trigger output and the channels.

Parameter Numeric [uV|mV|V] |MIN|MAX

Value coupling High = Offset + (Amplitude / 2)

Low = Offset – (Amplitude / 2)

Range coupling Offset

***RST value** Trigger output (0): 100 mV

Channels 1 and 2: 100 mV

Specified limits Trigger output (0): 50 mV... 2.0 V

Channels 1 and 2: 50 mV... 2.0 V

Absolute limits Trigger output (0): 2.0 V

Channels 1 and 2: 2.0 V

Example Set Trigger Output amplitude to 1 V.

```
:VOLT0 1V
```

[:SOURce]:VOLTage[0|1|2][:LEVel] [:IMMediate]:OFFSet

Syntax `[:SOUR]:VOLT[0|1|2][:LEV][:IMM]:OFFS[?]`

Form Set & Query

Description Programs the offset of the output signal for the trigger output and the channels.

Parameter Numeric [uV|mV|V] |MIN|MAX

Value coupling High = Offset + (Amplitude / 2)

Low = Offset – (Amplitude / 2)

Range coupling Amplitude

***RST value** Trigger output (0): 0 V

Channels 1 and 2: 0 V

Specified limits Trigger channel (0): –1.975 V ... +2.975 V

Channels 1 and 2: –1.975 V ... +2.975 V

Example Set Trigger Output offset to –100 mV.

```
:VOLT0:OFFS -100MV
```

[:SOURce]:VOLTage[0|1|2][:LEVel][:IMMediate]:HIGH

Syntax `[:SOUR]:VOLT[0|1|2][:LEV][:IMM]:HIGH[?]`

Form Set & Query

Description Programs the high-level of the output signal for the trigger output and the channels.

Parameter Numeric [uV|mV|V] |MIN|MAX

Value coupling Amplitude = High – Low

Offset = (High – Low) / 2

Range coupling Low-level

***RST value** Trigger channel (0): 0.1 V

Channels 1 and 2: 0.1 V

Specified limits Trigger channel (0): –1.95 ... +3.0 V

Channels 1 and 2: –1.95 ... +3.0 V

Absolute limits Trigger output (0): –2.2 ... +3.2 V

Channels 1 and 2: –2.2 ... +3.2 V

Example Set Channel 1 high-level to –200 mV.

```
:VOLT1:HIGH -200MV
```

[:SOURce]:VOLTage[0|1|2][:LEVel][:IMMediate]:LOW

Syntax `[:SOUR]:VOLT[0|1|2][:LEV][:IMM]:LOW[?]`

Form Set & Query

Description Programs the low-level of the output signal for the trigger output and the channels.

Parameter Numeric [uV|mV|V] |MIN|MAX

Value coupling Amplitude = High – Low

Offset = (High – Low) / 2

Range coupling High-level

***RST value** Trigger channel (0): 0.0 V

Channels 1 and 2: 0.0 V

Specified limits Trigger channel (0): –2.0 ... +2.95 V

Channels 1 and 2: –2.0 ... +2.95 V

Absolute limits Trigger output (0): –2.2 ... +3.2 V

Channels 1 and 2: –2.2 ... +3.2 V

Example Set Channel 1 low-level to –1 V.

```
:VOLT1:LOW -1V
```

[:SOURce]:VOLTage[0|1|2][:LEVel][:IMMediate]:TERM

Syntax `[:SOUR]:VOLT[0|1|2][:LEV][:IMM]:TERM[?]`

Form Set & Query

Description Programs the termination voltage of the output signal for the trigger output and the channels.

Parameter Numeric [uV|mV|V]

***RST value** Trigger output (0): 0.0 V

Channels 1 and 2: 0.0 V

Specified limits Trigger output (0): –2.0 V ... +3.0 V

Channels 1 and 2: –2.0 V ... +3.0 V

Example Set Channel 1 termination voltage to 1 V.

```
:VOLT1:TERM 1V
```

[:SOURce]:VOLTage[1|2]:LIMit[:AMPLitude]?

Syntax `[:SOUR]:VOLT[1|2]:LIM[:AMPL]?`

Form Query

Description Reads the current setting of the amplitude limit. The result is only valid if the “Limit to current levels” output mode is currently on (`[:SOURce]:VOLTage[1|2]:LIMit:STATe ON`).

***RST value** 100 mV

Example Read Channel 1 amplitude limit.

```
:VOLT1:LIM?
```

[:SOURce]:VOLTage[1|2]:LIMit:OFFSet?

Syntax `[:SOUR]:VOLT[1|2]:LIM:OFFS?`

Form Query

Description This command reads the current setting of the offset limit. The result is only valid if "Limit to current levels" output mode is currently on (`[:SOURce]:VOLTage[1|2]:LIMit:STATe ON`).

***RST value** 0 mV

Example Read Channel 1 offset limit.

```
:VOLT1:LIM:OFFS?
```

[:SOURce]:VOLTage[1|2]:LIMit:HIGH?

Syntax `[:SOUR]:VOLT[1|2]:LIM:HIGH?`

Form Query

Description This command reads the current setting of the high-level limit. The result is only valid if Limited output mode is currently on (`[:SOURce]:VOLTage[1|2]:LIMit:STATe ON`).

***RST value** 100 mV

Example Read Channel 1 high-level limit.

```
:VOLT1:LIM:HIGH?
```

[:SOURce]:VOLTage[1|2]:LIMit:LOW?

Syntax `[:SOUR]:VOLT[1|2]:LIM:LOW?`

Form Query

Description This command reads the current setting of the low-level limit. The result is only valid if "Limit to current values" mode is currently on (`[:SOURce]:VOLTage[1|2]:LIMit:STATe ON`).

***RST value** 0 V

Example Read Channel 1 low-level limit.

```
:VOLT1:LIM:LOW?
```

[:SOURce]:VOLTage[1|2]:LIMit:STATe

Syntax `[:SOUR]:VOLT[1|2]:LIM:STAT[?]`

Form Set & Query

Description Switches the “Limit to current values” output mode on or off. When you switch on Limited output mode the current high-level and low-level parameters are taken as limit values restricting the available ranges of all output-level parameters. You cannot program the output-levels beyond these temporary limits, until you switch off Limited output mode. The limits apply whether you program high/low levels or amplitude/offset levels.

Parameter ON|OFF|1|0

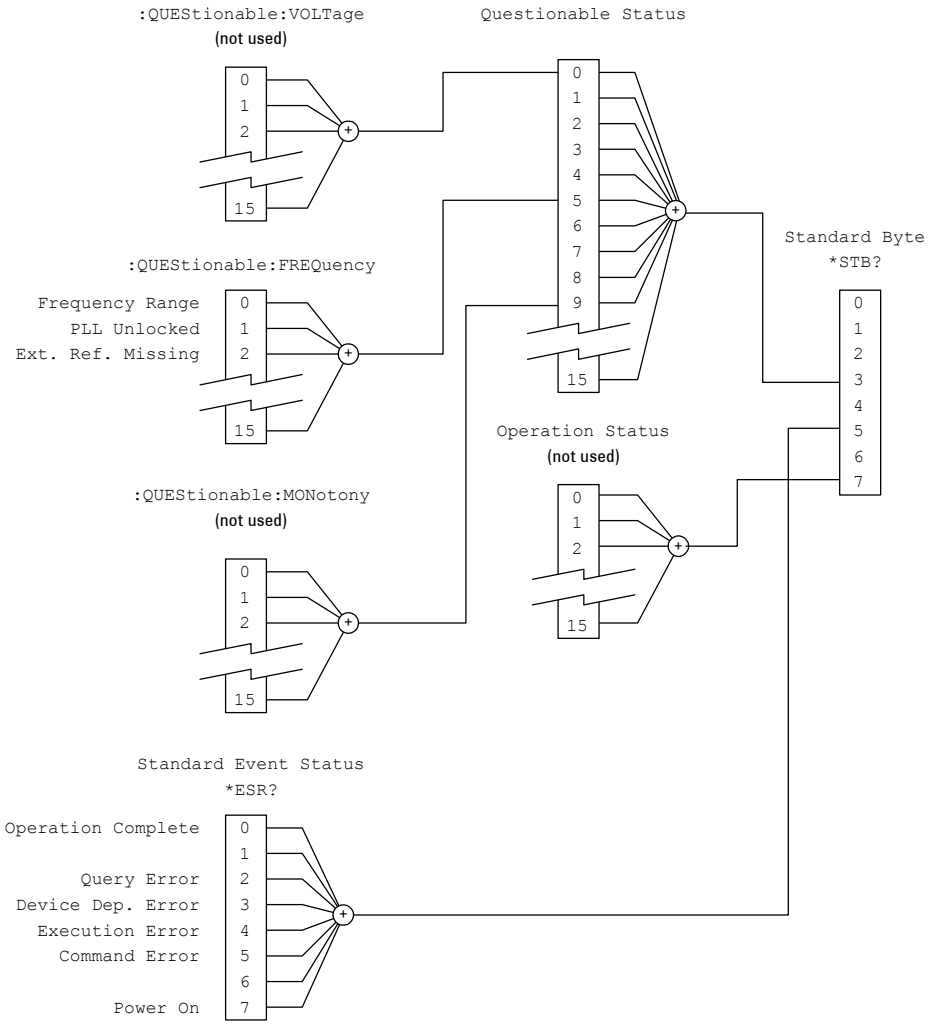
***RST value** OFF

Example Switch on Channel 1 Limited output mode.

```
:VOLT1:LIM:STAT ON
```

Status Handling Commands

The IEEE 488.2 specification requires status registers that contain information about the instrument's hardware and firmware. For the Agilent 81133A/81134A Pulse Generator, the status registers have the following structure:



The following table shows the Agilent 81133A/81134A Pulse Generator Status Handling Commands:

Command	Parameter	Description
:STATus		
:OPERation		
[:EVENt]?		Reads operation event register
:CONDition?		Reads operation condition register
:ENABle[?]	Numeric	Sets/reads operation enable register
:NTRansition[?]	Numeric	Sets/reads operation negative-transition filter
:PTRansition[?]	Numeric	Sets/reads operation positive-transition filter
:PRESet		Clears and presets status groups
:QUEStionable		
[:EVENt]?		Reads questionable event register
:CONDition?		Reads questionable condition register
:ENABle[?]	Numeric	Sets/reads questionable enable register
:NTRansition[?]	Numeric	Sets/reads questionable negative-transition filter
:PTRansition[?]	Numeric	Sets/reads questionable positive-transition filter
:VOLTage		
[:EVENt]?		Reads questionable voltage event register
:CONDition?		Reads questionable voltage condition register
:ENABle[?]	Numeric	Sets/reads questionable voltage enable register
:NTRansition[?]	Numeric	Sets/reads questionable voltage negative-transition register
:PTRansition[?]	Numeric	Sets/reads questionable voltage positive-transition register
:FREQuency		
[:EVENt]?		Reads questionable frequency event register
:CONDition?		Reads questionable frequency condition register
:ENABle[?]	Numeric	Sets/reads questionable frequency enable register

Command	Parameter	Description
:NTRansition[?]	Numeric	Sets/reads questionable frequency negative-transition register
:PTRansition[?]	Numeric	Sets/reads questionable frequency positive-transition register
:MONotony		
[:EVENt]?		Reads questionable monotony event register
:CONDition?		Reads questionable monotony condition register
:ENABle[?]	Numeric	Sets/reads questionable monotony enable register
:NTRansition[?]	Numeric	Sets/reads questionable monotony negative-transition register
:PTRansition[?]	Numeric	Sets/reads questionable monotony positive-transition register

:STATus:OPERation

This command tree accesses the OPERation status group.

The OPERation status group is not used by the Agilent 81133A/81134A Pulse Generator, therefore this command tree is redundant.

:STATus:PRESet

Syntax `:STAT:PRES`

Form Event

Description This command

- clears all status group event-registers
- clears the error queue
- presets the status group enable, PTR, and NTR registers as follows:

Status Group	Register	Preset Value
OPERation	ENABle	0000000000000000
	PTR	0111111111111111
	NTR	0000000000000000
QUEStionable	ENABle	0000000000000000
	PTR	0111111111111111
	NTR	0000000000000000
QUEStionable:VOLTage	ENABle	0111111111111111
	PTR	0111111111111111
	NTR	0000000000000000
QUEStionable:FREQuency	ENABle	0111111111111111
	PTR	0111111111111111
	NTR	0000000000000000
QUEStionable:MONotony	ENABle	0111111111111111
	PTR	0111111111111111
	NTR	0000000000000000

Parameter –

***RST value** –

:STATus:QUEStionable

This command tree accesses the QUEStionable status group.

The QUEStionable status group contains the summary bits from the QUEStionable:VOLTage, :FREQuency and MONotony status group.

The following commands are used to access the registers within the status group.

:STATus:QUEStionable[:EVENt]?

Syntax `:STAT:QUES[:EVEN]?`

Form Query

Description Reads the event register in the QUEStionable status group.

Parameter –

***RST value** –

:STATus:QUEStionable:CONDition?

Syntax `:STAT:QUES:COND?`

Form Query

Description Reads the condition register in the QUEStionable status group.

NOTE The Agilent 81133A/81134A Pulse Generator does not use this condition register, therefore, this command is redundant.

Parameter –

***RST value** –

:STATus:QUEStionable:ENABle

Syntax `:STAT:QUES:ENAB[?]`

Form Set & Query

Description Sets or queries the enable register in the QUEStionable status group.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:NTRansition

Syntax `:STAT:QUES:NTRansition[?]`

Form Set & Query

Description Sets or queries the negative-transition register in the QUEStionable status group.

NOTE The Agilent 81133A/81134A Pulse Generator does not use the transition registers of the QUEStionable status group, therefore, this command is redundant.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:PTRansition

Syntax `:STAT:QUES:PTRansition[?]`

Form Set & Query

Description Sets or queries the positive-transition register in the QUEStionable status group.

NOTE The Agilent 81133A/81134A Pulse Generator does not use the transition registers of the QUEStionable status group, therefore, this command is redundant.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:VOLTage

This command tree accesses the QUEStionable:VOLTage status group.
The QUEStionable:VOLTage status group monitors the currently programmed output voltage levels against their specified ranges.

The following commands are used to access the registers within the status group.

:STATus:QUEStionable:VOLTage[:EVENt]?

Syntax `:STAT:QUES:VOLT[:EVEN]?`

Form Query

Description Reads the event register in the QUEStionable:VOLTage status group.

Parameter –

***RST value** –

:STATus:QUEStionable:VOLTage:CONDition?

Syntax `:STAT:QUES:VOLT:COND?`

Form Query

Description Reads the condition register in the QUEStionable:VOLTage status group.

Parameter –

***RST value** –

:STATus:QUEStionable:VOLTage:ENABle

Syntax `:STAT:QUES:VOLT:ENAB[?]`

Form Set & Query

Description Sets or queries the enable register in the QUEStionable:VOLTage status group.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:VOLTage:NTRansition

Syntax `:STAT:QUES:VOLT:NTR[?]`

Form Set & Query

Description Sets or queries the negative-transition register in the QUEStionable:VOLTage status group.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:VOLTage:PTRansition

Syntax `:STAT:QUES:VOLT:PTR[?]`

Form Set & Query

Description Sets or queries the positive-transition register in the QUEStionable:VOLTage status group.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:FREQuency

This command tree accesses the QUEStionable:FREQuency status group.

The QUEStionable:FREQuency status group monitors the currently programmed frequency against the specified range, detects if the PLL in unlocked and indicates if there is a valid signal at the time base external input.

The following commands are used to access the registers within this status group.

:STATus:QUEStionable:FREQuency[:EVENt]?

Syntax `:STAT:QUES:FREQ[:EVEN]?`

Form Query

Description Reads the event register in the QUEStionable:FREQency status group.

Parameter –

***RST value** –

:STATus:QUEStionable:FREQuency:CONDition?

Syntax `:STAT:QUES:FREQ:COND?`

Form Query

Description Reads the condition register in the QUEStionable:FREQency status group.

Parameter –

***RST value** –

:STATus:QUEStionable:FREQuency:ENABle

Syntax `:STAT:QUES:FREQ:ENAB[?]`

Form Set & Query

Description Sets or queries the enable register in the QUEStionable:FREQency status group.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:FREQuency:NTRansition

Syntax `:STAT:QUES:FREQ:NTR[?]`

Form Set & Query

Description Sets or queries the negative-transition register in the QUEStionable:FREQency status group.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:FREQuency:PTRansition

Syntax `:STAT:QUES:FREQ:PTR[?]`

FormForm Set & Query

Description Sets or queries the positive-transition register in the QUEStionable:FREQency status group.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:MONotony

This command tree accesses the QUEStionable:MONotony status group.

The QUEStionable:MONotony status group monitors the frequency, width, delay and amplitude parameters. The range of these parameters is made up of several internal ranges and when the parameter moves from one internal range to the next a discontinuity can occur. For example, increasing the frequency at a range boundary could cause the actual output frequency to decrease slightly. The range boundaries also vary with temperature, and a significant temperature change could cause the instrument to move to the next range in order to maintain the current parameter within specification.

When a parameter range change occurs, the corresponding bit in the QUEStionable:MONotony status event register is set to indicate that the output signal may not vary monotonically with the programmed parameter value.

The following commands are used to access the registers within the status group.

:STATus:QUEStionable:MONotony[:EVENt]?

Syntax `:STAT:QUES:MON[:EVEN]?`

Form Query

Description This command reads the event register in the QUEStionable:MONotony status group.

Parameter –

***RST value** –

:STATus:QUEStionable:MONotony:CONDition?

Syntax `:STAT:QUES:MON:COND?`

Form Query

Description This command reads the condition register in the QUEStionable:MONotony status group.

NOTE The Agilent 81133A/81134A Pulse Generator does not use the condition register of the QUEStionable:MONotony status group, therefore, this command is redundant.

Parameter –

***RST value** –

:STATus:QUEStionable:MONotony:ENABle

Syntax `:STAT:QUES:MON:ENAB[?]`

Form Set & Query

Description Sets or queries the enable register in the QUEStionable:MONotony status group.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:MONotony:NTRansition

Syntax `:STAT:QUES:MON:NTR[?]`

Form Set & Query

Description Sets or queries the negative-transition register in the QUEStionable:MONotony status group.

NOTE The Agilent 81133A/81134A Pulse Generator does not use the transition registers of the QUEStionable:MONotony status group, therefore, this command is redundant.

Parameter Numeric

***RST value** –

Specified limits 0 ... 32767

:STATus:QUEStionable:MONotony:PTRansition

Syntax `:STAT:QUES:MON:PTR[?]`

Form Set & Query

Description This command sets or queries the positive-transition register in the QUEStionable:MONotony status group.

NOTE The Agilent 81133A/81134A Pulse Generator does not use the transition registers of the QUEStionable:MONotony status group, therefore, this command is redundant.

Parameter Numeric

***RST value** –

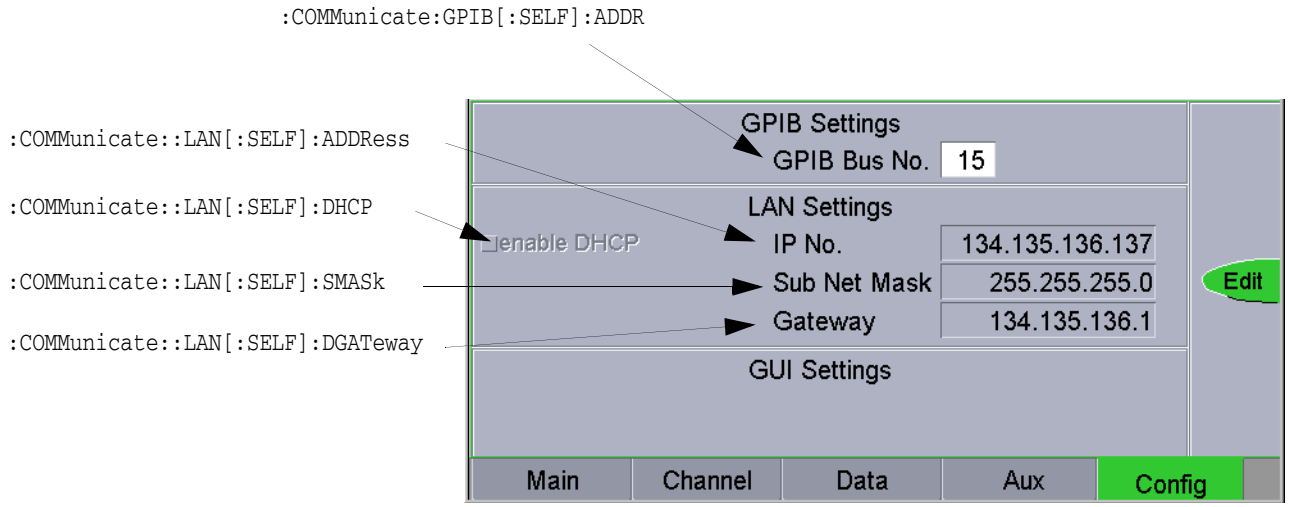
Specified limits 0 ... 32767

SYSTem Commands

The following table shows the Agilent 81133A/81134A Pulse Generator SYSTem Commands:

Command	Parameter	Description
:SYSTem		
:ERRor?		Reads error queue
:KEY[?]	Numeric	Simulates key press or reads key queue
:PRESet		*RST without changing display state
:SET[?]	Block data	Sets/reads complete instrument setting
:VERSion?		Reads SCPI compliance version
:COMMunicate		
:LAN		
[:SELF]		
:ADDRess	String	Sets/reads instrument's LAN address. String format is A.B.C.D with A,B,C,D is number between 1 and 255
:SMASk	String	Sets/reads subnet mask of current LAN. String format is similar to :ADDR
:DGATeway	String	Sets/reads default Gateway for the instrument. String format is similar to :ADDR

Command	Parameter	Description
:HADDRess	String	Set/read host LAN address for the instrument. This setting is important for FTP transfer
:NAME	alphanumeric	Sets the LAN name for the instrument
:DHCP	1|0|ON|OFF	Enables/disables DHCP configuration
:GPIB		
[:SELF]		
:ADDR	Numeric	Sets/reads GPIB bus No. A value between 0 to 30 is required



:SYSTem:ERRor?

Syntax :SYST:ERR?

Form Query

Description This command is used to read the Agilent 81133A/81134A Pulse Generator error queue. The Agilent 81133A/81134A Pulse Generator error queue can store up to 32 error codes on a first-in-first-out basis. When you read the error queue, the error number and associated message are put into the instrument's output buffer.

If the error queue is empty, the value 0 is returned, meaning No Error. If the queue overflows at any time, the last error code is discarded and replaced with –350 meaning Queue Overflow.

Refer to *"Troubleshooting" on page 97* for a list of the error messages.

Parameter –

***RST value** –

:SYSTem:PRESet

Syntax :SYST:PRES

Form Event

Description This command is equivalent to *RST, except that there is no change to the :DISPlay[:WINDow][:STATe]. Use this command instead of *RST if you want the display to remain switched off during program execution.

Parameter –

***RST value** –

:SYSTem:SET

Syntax :SYST:SET[?]

Form Set & Query

Description In query form, the command reads a block of data containing the instrument's complete setup. The setup information includes all parameter and mode settings, but does not include the contents of the instrument setting memories, the status group registers or the :DISPlay[:WINDow][:STATe]. The data is in a binary format, not ASCII, and cannot be edited.

In set form, the block data must be a complete instrument setup that was produced with the query form of this command.

Parameter Block data

:SYSTem:VERSion?

Syntax :SYST:VERS?

Form Query

Description This command reads the SCPI revision to which the instrument complies.

***RST value** –

:SYSTem:COMMunicate:LAN[:SELF]:DHCP

Syntax `:SYST:COMM:LAN[:SELF]:DHCP`

Form Set

Description Enables/disables the DHCP.

- DHCP enabled

  If DHCP is enabled, the instrument will request its own LAN settings from the network. You only need to specify the LAN name with *":SYSTEM:COMMunicate:LAN[:SELF]:NAME" on page 83*.

- DHCP disabled

  If DHCP is disabled, you have to set the instrument IP address, subnet mask, gateway and host address with:

  – *":SYSTem:COMMunicate:LAN[:SELF] :ADDRess" on page 84*

  – *":SYSTem:COMMunicate:LAN[:SELF] :SMASk" on page 84*

  – *":SYSTem:COMMunicate:LAN[:SELF] :DGATeway" on page 85*

Parameter 1|0|ON|OFF

***RST value** –

Example Disable DHCP:

```
:SYST:COMM:LAN[:SELF]:DHCP OFF
```

:SYSTEM:COMMunicate:LAN[:SELF]:NAME

Syntax `:SYST:COMM:LAN[:SELF]:NAME <LAN name>`

Form Set

Description Sets the LAN name. This command is only necessary for DHCP.

Parameter alphanumeric

***RST value** –

Example

```
:SYST:COMM:LAN:NAME PP81134A01
```

:SYSTem:COMMunicate:LAN[:SELF] :ADDRess

Syntax `:SYST:COMM:LAN[:SELF]:ADDR[?]`

Form Set & Query

Description Sets the instrument's IP address.

NOTE This parameter must only be set if DHCP is not available. See *":SYSTem:COMMunicate:LAN[:SELF]:DHCP" on page 83*.

Parameter String <no>.<no>.<no>.<no> in quotes, where <no> is in the range 1 ... 255.

***RST value** –

Example `:SYST:COMM:LAN:ADDR "150.215.17.9"`

:SYSTem:COMMunicate:LAN[:SELF] :SMASk

Syntax `:SYST:COMM:LAN[:SELF]:SMAS[?]`

Form Set & Query

Description Sets the instrument's subnet mask.

NOTE This parameter must only be set if DHCP is not available. See *":SYSTem:COMMunicate:LAN[:SELF]:DHCP" on page 83*.

Parameter String <no>.<no>.<no>.<no> in quotes, where <no> is in the range 0 ... 255.

***RST value** –

Example `:SYST:COMM:LAN:SMAS "255.255.240.000"`

:SYSTem:COMMunicate:LAN[:SELF] :DGATeway

Syntax `:SYST:COMM:LAN[:SELF]:DGAT[?]`

Form Set & Query

Description Sets the instrument's gateway.

NOTE This parameter must only be set if DHCP is not available. See *":SYSTem:COMMunicate:LAN[:SELF]:DHCP" on page 83*.

Parameter String <no>.<no>.<no>.<no> in quotes, where <no> is in the range 1 ... 255.

***RST value** –

Example `:SYST:COMM:LAN:DGAT "150.215.001.001"`

:SYSTem:COMMunicate:GPIB[:SELF]:ADDR

Syntax `:SYST:COMM:GPIB[:SELF]:ADDR[?]`

Form Set & Query

Description Sets/reads the instrument's GPIB bus number.

Parameter Numeric

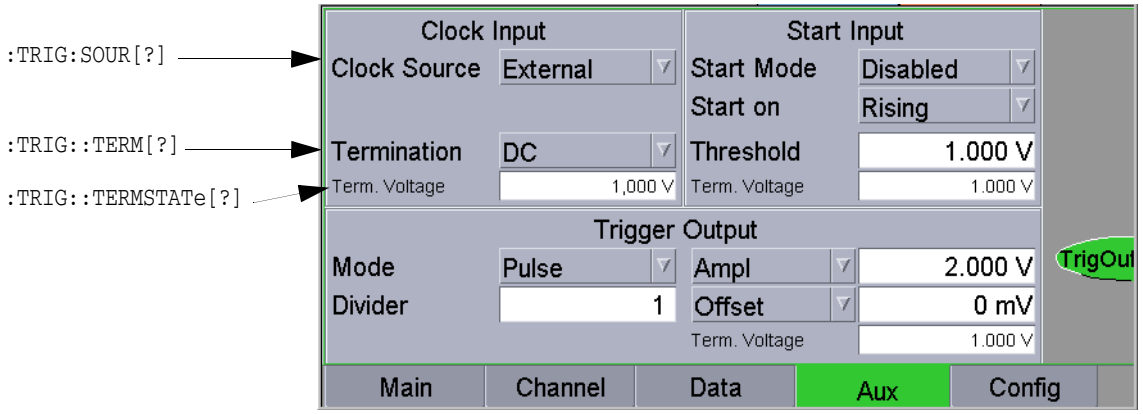
***RST value** 13

Example `:SYST:COMM:GPIB[:SELF]:ADDR 15`

TRIGger Commands

The following table shows the Agilent 81133A/81134A Pulse Generator TRIGger Commands:

Command	Parameter	Description
:TRIGger		
:SOURce[?]	IMMediate|EXTernal | REFerence| IDIRect| EDIRect	Sets/reads timebase mode internal, external, external 10 MHz reference, internal direct and external direct
:TERM[?]	Numeric [uV|mV|V]	Sets/reads termination voltage
:TERMSTATe[?]	ON|OFF	Sets/reads termination state. Defines whether the external clock input (Clock In) connector is AC or DC terminated



:TRIGger:SOURce

Syntax `:TRIG:SOUR[?]`

Form Set & Query

Description This command is used to switch the timebase mode between Internal (`IMMediate`) and External (`EXTernal`).

With `:TRIGger:SOURce IMMediate` (internal timebase) the frequency (or period) is controlled with the `[:SOURce]:FREQuency` (or `[:SOURce][:PULSe]:PERiod`) command.

With `:TRIGger:SOURce EXTernal`, the frequency (or period) is controlled by the external signal applied to the external clock input connector (Clock In).

Parameter IMMediate|EXTernal|REFerence|IDIRect|EDIRect

- IMMediate

  The clock is derived from the internal oscillator.

- EXTernal

  Enables the external clock input (*Clock In*) to accept an external clock signal that forms the time base. The frequency is measured once by selecting the Measure function from the user interface or as a remote SCPI command (`:MEASure:FREQuency?`).

  This value is then used to calculate frequency-dependent values, like the pulse width or the phase (available at the Channel page).

- REFerence

  Enables the external clock input (*Clock In*) to apply a 10 MHz reference clock. This clock is used as a reference for all timing parameters.

- IDIRect|EDIRect

  The direct modes allow changes of frequency without dropouts in the range of 1:2. They are used for applications (precise clock source), where dropouts would make a measurement impossible, for example, PLL frequency sweep and micro processor clock sweep.

  – IDIRect

    Allows you to vary the clock derived from the internal oscillator in the range of one octave.

– EDIRect

Allows you to vary the frequency of the external clock signal in the range of one octave.

Range switching occurs at the following frequency values:

- 1680 MHz
- 840 MHz
- 420 MHz
- 210 MHz
- 105 MHz
- 51.5 MHz
- 25.75 MHz

These values are based on 1680 MHz, subject to the frequency divider.

***RST value** IMMediate

Example Select the external timebase mode.

```
:TRIG:SOUR EXT
```

:TRIGger:TERM

Syntax `:TRIG:TERM[?]`

Form Set & Query

Description Sets/reads the termination voltage for the external clock input. See *":TRIGger:SOURce" on page 87*.

NOTE The termination voltage can only be specified if the Clock In connector is DC terminated.

Parameter Numeric [uV|mV|V]

***RST value** 0 mV

Specified limits –2.0 V ... 3.0 V

Example Set the termination voltage of the signal applied to the external clock input to 1V.

```
:TRIG:TERM 1V
```

:TRIGger:TERM:STATE[?]

Syntax `:TRIG:TERM:STATE[?]`

Form Set & Query

Description Defines whether the external clock input connector (Clock In) is AC or DC terminated.

Parameter ON|OFF where ON = DC and OFF = AC terminated.

***RST value** OFF

Example Set the coupling of the external clock input connector to DC.

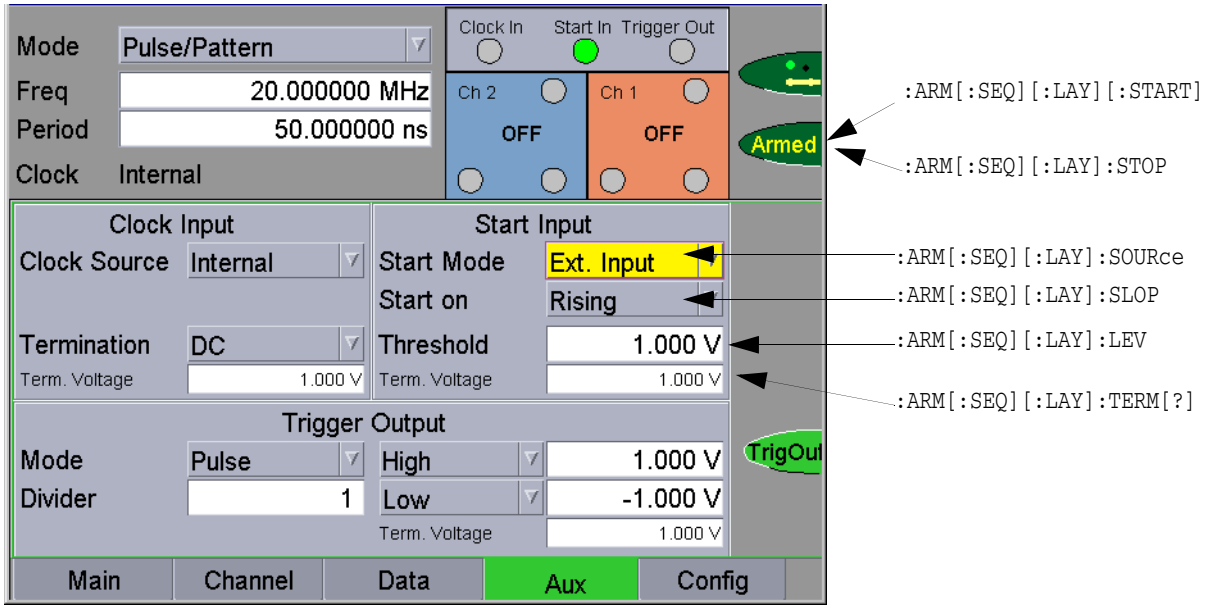
```
:TRIG:TERM:STATE ON
```

ARM Commands

The following table shows the Agilent 81133A/81134A Pulse Generator ARM Commands:

Command	Parameter	Description
:ARM		External start input
[:SEQuence]		
[:LAYer]		
:LEVel	Numeric[mV|V]	Sets the trigger threshold
:SLOPe	POSitive|NEGative	Trigger set to leading/trailing edge of external signal
:SOURce	IMMediate|MANual|EXTernal	Sets the start input to disabled/manual (by key)/external started
:TERM[?]	Numeric [mV|V]	Sets/reads termination voltage
[:STARt]		Starts signal output
:STOP		Stops signal output

The following figure shows how the ARM commands are implemented in the 81133A/81134A user interface.



:ARM[:SEQuence][:LAYer]:LEVel

Syntax `:ARM[:SEQ][:LAY]:LEV`

Form Set & Query

Description Specifies the threshold voltage for the external start signal.

NOTE The threshold can only be specified if the external start mode is selected (`":ARM:SOURce EXT"`).

Parameter Numeric [mV|V]

Specified limits –2 V ... 3.0 V

Absolute limits –2.0 V ... 3.0 V

***RST value** 100 mV

Example Sets the threshold voltage to 2.0 V.

```
:ARM:LEV 2.0 V
```

:ARM[:SEQuence][:LAYer]:SLOPe

Syntax `:ARM[:SEQ][:LAY]:SLOP`

Form Set & Query

Description Specifies whether the signal is generated at the rising or falling edge of the external start signal.

NOTE The threshold can only be specified if the external start mode is selected (`":ARM:SOURce EXT"`).

Parameter POSitive|NEGative

- POSitive

  The signal is generated at the rising edge.

- NEGative

  The signal is generated at the falling edge.

***RST value** POS

Example Specifies that the signal is generated at the falling edge.

```
:ARM:SLOP NEG
```

:ARM[:SEQuence][:LAYer]:SOURce

Syntax `:ARM[:SEQ][:LAY]:SOUR`

Form Set & Query

Description Specifies when the generated signal is output immediately, by manual start or depending on an external signal at the Start In connector.

Parameter IMMediate|MANual|EXTernal

The instrument provides the following start modes:

- IMMediate

  The generated signal is always available at the outputs (assumed that the outputs are enabled).

- MANual

  The generated signal is output after *Start* at the instrument panel is pressed.

- EXTernal

  Select this start mode to send the generated signal to the outputs depending on the external signal applied at the *Start In* connector.

  You can define the following parameters that the external signal must meet:

  – Threshold (voltage)

    Set the threshold with *":ARM[:SEQuence][:LAYer]:LEVel" on page 91*.

  – Termination voltage

    Set the termination voltage with *":ARM[:SEQuence][:LAYer]:TERM" on page 94*.

  – Edge (rising/falling)

    Set the edge with *":ARM[:SEQuence][:LAYer]:SLOPe" on page 92*.

***RST value** `IMMediate`

Example Set the start mode to external.

```
:ARM:SOUR EXT
```

:ARM[:SEQuence][:LAYer]:TERM

Syntax :ARM[:SEQ][:LAY]:TERM[?]

Form Set & Query

Description Sets/reads the termination voltage for the start input signal.

Parameter Numeric [mV|V]

Specified limits –2.0 V ... 3.0 V

***RST value** 0 mV

Example Set the termination voltage to 1 V.

```
:ARM:TERM 1V
```

:ARM[:SEQuence][:LAYer][:STARt]

Syntax :ARM[:SEQ][:LAY][STAR]

Form Event

Description Puts the instrument in armed mode. This means, that the instrument waits for the selected edge to appear.

NOTE The instrument can only be put in armed mode if the external start mode is selected (":ARM:SOURce EXT").

Parameter –

***RST value** –

Example Set the armed mode:

```
:ARM
```

:ARM[:SEQuence][:LAYer]:STOP

Syntax :ARM[:SEQ][:LAY]:STOP

Form Event

Description Deactivates the armed mode for the instrument.

Parameter –

***RST value** –

Example Deactivate the armed mode:

```
:ARM:STOP
```

Troubleshooting

This chapter provides basic troubleshooting tips that you can use if the instrument is not performing as expected.

Error Generated

The instrument generates error messages as follows:

```
-221:Settings conflict:String describing the error
```

The string describing the error could be one of the following:

- "**divider of channel [1|2] leads to a frequency below minimum frequency**"

  This occurs if the frequency below one of the channels is set below the minimum (15 MHz). This could happen if the frequency is decreased or if the frequency divider is increased.

- "**pulsewidth of channel [1|2] is too small**"

  May happen:

  – If the frequency is decreased and the channel is in duty mode, or

  – if the signal mode is switched from NRZ to RZ or R1, or

  – if clock mode is switched from internal/external direct to a none direct mode

- "**pulse width of channel [1|2] is too large**"

  May happen:

  – If the frequency is increased and the channel is in duty mode, or

  – if the signal mode is switched from NRZ to RZ or R1, or

  – if clock mode is switched from internal/external direct to a none direct mode.

- "**delay of channel [1|2] below minimum**"

  May happen:

  – If the frequency is decreased and the channel is in phase mode, or

  – if the square mode is switched from square to another mode, or

  – if clock mode is switched from internal/external direct to a none direct mode.

- "**delay of channel [1|2] above maximum**"

  May happen:

  – If the frequency is increased and the channel is in phase mode, or

  – if the channel mode is switched from SQUARE to another mode, or

  – if clock mode is switched from internal/external direct to a none direct mode.

- "**amplitude of trigger is too small**"

  May happen:

  – If the trigger high level is decreased, or

  – if the low level is increased.

- "**amplitude of channel [1|2] is too small**"

  May happen:

  – If the channel high level is decreased, or

  – if the low level is increased.

- "**amplitude of trigger is too large**"

  May happen:

  – If the trigger high level is increased, or

  – if the low level is decreased.

- "**amplitude of channel [1|2] is too large**"

  May happen

  – If the channel high level is increased, or

  – if the low level is decreased.

- "**low level of trigger is below minimum**"

  May happen:

  - If the trigger amplitude is increased, or
  - if the offset is decreased.
- "**low level of channel [1|2] is below minimum**"

  May happen:
  - If the channel amplitude is increased, or
  - if the offset is decreased.
- "**high level of trigger exceeds maximum**"

  May happen:
  - If the trigger amplitude is increased, or
  - if the offset is increased.
- "**high level of channel [1|2] exceeds maximum**"

  May happen:
  - If the channel amplitude is increased, or
  - if the offset is increased.
- "**high level of trigger is lower than low level**"

  May happen:
  - If the trigger amplitude is set to a negative value, or
  - if high level and low level are set in parallel by list of semicolon- separated SCPI commands.
- "**high level of channel [1|2] is lower than low level**"

  May happen:
  - If the channel amplitude is set to a negative value, or
  - if high level and low levels are set in parallel by list of semicolon- separated SCPI commands.

Instrument not Operable via LAN

If the instrument cannot be programmed via LAN, make sure that you have installed the latest Agilent I/O library on your computer.

Visit the Agilent web site for the newest version.

Differences between the 8133A and the 81133A/81134A

This chapter shows you how to adapt a program written for the 8133A 3 GHz Pulse Generator to the new 81133A/81134A instrument.

New Commands

The following tables list all commands new for the 81133A/81134A pulse/pattern generator:

DIAGnostic Commands
":DIAG:CHANnel[1|2]:PPERformance" on page 24

DIGital Commands
":DIGital[1|2][:STIMulus]:PATTern:LENGth" on page 31
":DIGital[1|2][:STIMulus]:SIGNal: CROSsover:[VALue]" on page 32
":DIGital[1|2][:STIMulus]:SIGNal: CROSsover:[STATe]" on page 34

OUTPut Commands
":OUTPut:CENTral" on page 44

SOURce Commands
"[:SOURce]:FUNCtion:MODe[1|2]" on page 51
"[:SOURce]:PM[1|2]" on page 59
"[:SOURce]:PM[1|2]:SENSitivity" on page 59
"[:SOURce]:VOLTage[0|1|2][:LEVel] [:IMMediate]:TERM" on page 64

SYSTem Commands
":SYSTem:COMMunicate:LAN[:SELF]:DHCP" on page 83
":SYSTEM:COMMunicate:LAN[:SELF]:NAME" on page 83
":SYSTem:COMMunicate:LAN[:SELF] :ADDRess" on page 84
":SYSTem:COMMunicate:LAN[:SELF] :SMASk" on page 84
":SYSTem:COMMunicate:LAN[:SELF] :DGATeway" on page 85
":SYSTem:COMMunicate:GPIB[:SELF]:ADDR" on page 85

TRIGger Commands
":TRIGger:TERM" on page 88
":TRIGger:TERM:STATE[?]" on page 89

ARM Commands
":ARM[:SEQuence][:LAYer]:LEVel" on page 91
":ARM[:SEQuence][:LAYer]:SLOPe" on page 92
":ARM[:SEQuence][:LAYer]:SOURce" on page 93
":ARM[:SEQuence][:LAYer]:TERM" on page 94
":ARM[:SEQuence][:LAYer][:STARt]" on page 94
":ARM[:SEQuence][:LAYer]:STOP" on page 95

Same Commands with Internal Change

The following tables list all commands that have been changed internally due to new 81133A/81134A functionality:

Common Commands (see *"Common Commands" on page 19*)
*RCL
*SAV

DIGital Commands
":DIGital[1|2][:STIMulus]:PATTern[:DATa]" on page 27

SOURce Commands
"[:SOURce]:FUNCtion[:SHAPe]" on page 50

Obsolete Commands

The following tables list all commands that no longer exist for the 81133A/81134A pulse/pattern generator:

DIAGnostic Commands
:DIAG:CHANnel[1|2]:CABLecomp
:DIAG:CHANnel[1|2]:SMOothshape
:DIAG:TEMPCAL

MEASure Commands
:MEASure:TEMPerature?

SOURce Commands
[:SOURce]:FUNCtion:SOURce[?]
[:SOURce]:FUNCtion:BURSTcount
[:SOURce]:FUNCtion:RBURSTcount

SYSTem Commands
:SYSTem:KEY[?]

TRIGger Commands
:TRIGger:[START]
:TRIGger:STOP

Printed in Germany March 2007

5988-7402EN

Agilent Technologies