HMC-T2200 Synthesized Signal Generator Family

HMC-T2220/40/70
HMC-T2220B

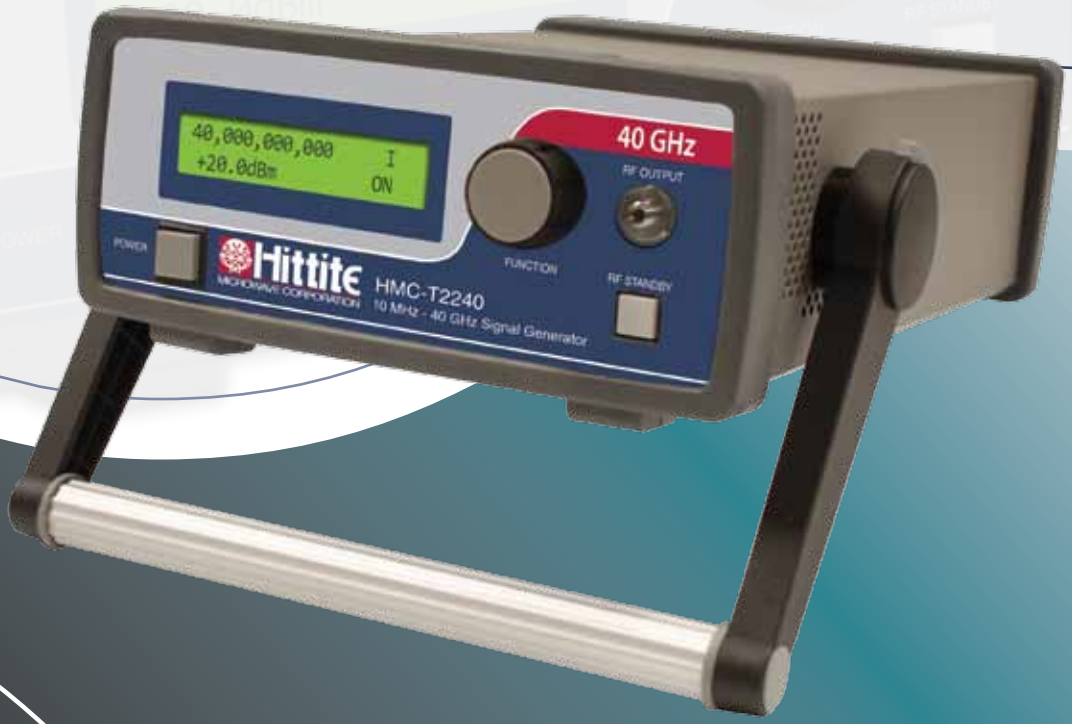
131547 Rev G - v06.0512

Analog, Digital & Mixed-Signal ICs, Modules, Subsystems & Instrumentation

ECN# CP120624

Programmer's Manual

Installation, Operation & Maintenance Guide for HMC-T2220, HMC-T2220B, HMC-T2240, and HMC-T2270



HMC-T2220/40/70

Order On-Line at: www.tm-hittite.com
Receive the latest product releases - click on "My Subscription"

2 Elizabeth Drive Chelmsford, MA 01824
978-250-3343 tel • 978-250-3373 fax • TE@hittite.com

Hittite MICROWAVE CORPORATION

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

1.0 Introduction

The HMC-T2200s are SCPI based signal generators supporting CW and stepped sweep operation through USB, GPIB and Ethernet.

Note: The programming capability outlined in this manual is for the HMC-T2200 Synthesized Signal Generator Family. For simplicity only the HMC-T2240 will be referenced throughout the document.

2.0 Quick Start

2.1 USB Quick Start

The HMC-T2240 presents its USB port as a serial port. The serial port interface is easy to control from virtually any programming language or environment and requires no drivers other than the ones distributed with Windows and a Hittite provided .INF to use. Any application which can access a serial port (ie. Windows COM port) can talk directly to the HMC-T2240.

2.1.1 HyperTerminal

You can use an application like HyperTerminal to send SCPI commands directly to it to try things out and for debug. HyperTerminal is no longer available on Windows 7®. See the section below on Tera Term Pro or use any terminal program you prefer.

HyperTerminal is part of the standard Windows distribution (prior to Windows 7®) and can be found under the Start button:

Start -> Programs -> Accessories -> Communications -> HyperTerminal

The serial port settings that Windows and HyperTerminal present (baud rate, parity, stop bits) are unused. However, you may want to **"Echo typed characters locally"** so you can see what you are typing.

From within HyperTerminal:

File -> Properties

On the Properties dialogue box, select the **Settings** tab.

Click the **ASCII Setup**… button.

Check the **Echo typed characters locally** box.

You may also want to check the **Send line ends with line feeds** and **Wrap lines that exceed terminal width** boxes. (The HMC-T2240 doesn't need them, but they make the output more readable.)

2.1.2 Tera Term Pro Terminal Emulator

Tera Term Pro is a free terminal emulation program which can connect to the HMC-T2240 over USB, Telnet, and RS-232.

It can be downloaded from many websites, including:

http://forceforge.jp/projects/ttssh2/

While the above website did not claim Windows 7® support when this document was written, Tera Term Pro was used under Windows 7® to generate some of the examples.

A strictly minimal installation ("Tera Term & Macro") is sufficient.

Once Tera Term (version 4.67 for this example) is running, use

Setup -> Terminal...

and verify the following:

Receive:	**LF**	
Transmit:	**CR+LF**	
Local echo	**Checked**	**(Unchecked for Telnet)**
Coding	**UTF-8**	
Terminal size	**80 x 24**	
Term size = win size	**Checked**	
Terminal ID:	**VT100**	
Answerback:		
Auto switch	**Unchecked**	
locale:	**american**	
CodePage:	**65001**	

also:

Setup -> Serial port...

and verify the following:

Port:	**<Don't Care>**
Baud rate:	**115200**
Data:	**8 bit**
Parity:	**none**
Stop:	**1 bit**
Flow Control:	**none**

To select a connection:

File -> New Connection...

For USB, select Serial, then Port: should have an option for:

COMn: HMC-T2240 Signal Generator (COMn)

If you do not see the option listed, you may already be connected.

File -> Disconnect

You should now be able to type commands like "***IDN?**" and see the response.

Once it is working, save the settings using:

Setup -> Save setup...

Hittite Microwave Corporation is not involved in the development or distribution of Tera Term Pro and cannot guarantee it or provide technical support for it; it is merely identified as a possible alternative to HyperTerminal. See the Tera Term Pro software and websites for terms of license and documentation.

2.1.3 HMCT2XXXGUI

The HMC-T2200 install CD contains a graphical interface called HMCT2XXXGUI which allows you to program Frequency, Power, RF Output, and sweep related parameters.

The display has a pull-down list of synthesizers which can be selected. Note that if another application has a synthesizer open, you will not be able to open it from HMCT2XXGUI.

There is a right click menu which allows you to

- **Refresh** the display without having to go up to the button at the top.
- **Remap Hardware** to update the pull-down list of signal generators when units have been added or removed.
- **Close** a HMC-T2240 so it will be available for other applications to use.
- **Show Sweep Parameters** can be unchecked to make the display smaller if you aren't sweeping.
- **About**… shows revision information.
- **LIST** Mode brings up a window to make it easy to look at FREQ, POW, OUTP, and DWEL LIST points, as well as LIST:SEQuence.

You can bring up multiple copies of HMCT2XXXGUI if you have multiple synthesizers to control.

HMCT2xxxGUI only works over USB.

2.2 GPIB

The HMC-T2240 IEEE 488.2 (GPIB) interface supports all of the applicable 488.2 and SCPI commands supported over USB and Ethernet plus a number of IEEE 488.1 specific features including service requests, serial polls, parallel polls, group execute trigger, and high speed operation.

2.2.1 GPIB Quick Start

The HMC-T2240 GPIB Address (SW Ver < 2.3) can be set by pressing and holding the front panel knob for 2 seconds. When the "GPIB Addr:" screen appears, turn the knob to change the value, then press and hold the knob again to get the screen with the serial number (SN), hardware version (HW VER), and software version (SW VER), and press and hold the knob again to get back to the normal display. The GPIB Address can also be set through any of the programming interfaces.

The HMC-T2240 GPIB Address (SW Ver ≥ 2.3) can be set by pressing and holding the front panel knob for 2 seconds. When the next menu screen is displayed, the "GPIB Addr" menu option should be selected by rotating the control knob. Press and release the knob to enter the GPIB address menu. Select the address field by rotating the knob, and then depress and release the knob. Turn the knob to change the value, then press and release the knob again to go back to the previous menu, or press and hold the knob for 2 seconds to go back to the main frequency & power display. The GPIB Address can also be set through any of the programming interfaces.

No drivers beyond the ones that come with your GPIB controller should be necessary.

2.2.2 GPIB Feature Set and Other Implementation Information

This section includes various information related to GPIB (IEEE 488.1(E):2003/IEC 60488-1(E):2004) standard compliance.

The HMC-T2240 supports the following GPIB functions:

Function	Symbol
Source Handshake	SH1
Acceptor Handshake	AH1
Talker, no "talk only" mode	T6
Listener, no "listen only" mode	L4
Service Request	SR1
Remote/Local	RL1
Parallel Poll, Remote configuration	PP2
Device Clear	DC1
Device Trigger	DT1
No Controller capability	C0
Source Handshake, Extended	SHE1
Acceptor Handshake, Extended	AHE1
Configuration	CF1
Three State Drivers (open collector during parallel poll)	E2

The HMC-T2240's hardware input buffer holds 16 bytes. The maximum input string length is 256 bytes for SW Ver <2.5; unlimited for SW Ver >=2.5. New input beyond what will fit in the hardware buffer is not accepted until all the commands in the current string have been parsed (translated into an executable form) but not necessarily executed. The IEEE 488.2 defined INTERRUPTED and UNTERMINATED errors will be issued if new commands are sent before the output of previous queries has been read or if a read is started before a complete command line has been sent.

For best performance, read output as soon as it becomes available. MAV (Message Available) does not get set until output is ready to send. If there are multiple queries on one line, MAV may be dropped between queries, particularly if there is a slow operation between them. Queries generally collect their information in the execution phase to preserve FIFO order relative to the commands that go before or come after them, but error messages are put into the error queue in the order they are detected. See **SCPI Tips**, section 3.2, for query message length information.

LIST write operations may violate this FIFO ordering. LIST operations access the LIST memory directly. Attempting to access the LIST memory while it is being used (during a LIST mode sweep) will result in an error. Reading LISTs during a sweep or other time critical operation is likely to disrupt timing and is not recommended, even if an error message is not generated.

The HMC-T2240 accepts input lines terminated with EOI (a GPIB signal which can be asserted with the last character of a string,) newline (10, 0x0A, \n,) or both. The carriage return character (13, 0x0D, \r,) is not recognized by the IEEE 488 standard and is treated as whitespace.

The HMC-T2240 output lines are terminated with both EOI and newline.

Serial Polls and Parallel Polls can be used at "any" time, including while ***WAI** and ***OPC?** are in effect, and are unlikely to introduce significant jitter.

(As the unit is addressed and unaddressed over GPIB, small delays are incurred, but they will not affect sweep timing, for instance.)

Normal commands (such as ***STB?**) are blocked by ***WAI** and ***OPC?** until the sweep is completed. Commands which affect the RF output (**FREQuency, POWer, OUTPut**) may interfere with sweep timing.

Note: Local Lockout (LLO) does not lock out the USB and Ethernet interfaces. This is contrary to the IEEE 488 specs which consider anything other than GPIB to be a "local" control.

If Local Lockout is active and the user presses and holds the front panel knob, the URQ (User Request) bit in the ESR will be set, and, if ***ESE** and ***SRE** are set appropriately, a service request can be generated so the controlling program can restore local control when it is safe to do so.

The GPIB Address is in the range 0 to 30. Attempting to program it outside this range will result in an error. The HMC-T2240 does not use secondary addressing. ***AAD** and ***DLF**, which allow automatic address assignment, are not supported. It is not necessary to power cycle the HMC-T2240 after changing the GPIB Address. However, changing it during a GPIB operation may result in undesirable behavior.

The HMC-T2240 goes into its ***RST** state after power on. The ***PSC** command is not supported at this time.

The following commands are coupled: **STARt, STOP, CENTer, SPAN** (frequency and power.)

2.3 Ethernet

The HMC-T2240 supports Socket and Telnet connections over Ethernet with the same functionality as USB. The Socket interface is a programming interface; Telnet is for typing commands in by hand for setup or debug. Any of the SCPI commands can be sent from any interface, so you can use USB or Telnet to set the Socket Port number, for instance. In order to use a Socket or Telnet, the address of the HMC-T2240 must be configured. Ethernet addresses can be configured from the front panel with SW Ver ≥ 2.3.

No drivers beyond the ones that come with normal networked computers should be necessary.

2.3.1 Ethernet Addressing

The goal of configuring the HMC-T2240's Ethernet settings is to get it to work on *your* network. Consulting your network administrator about how to configure this device can save a lot of time.

The easiest way to configure the HMC-T2240's Ethernet address is with DHCP. If your network is configured for DHCP (Dynamic Host Configuration Protocol), you should[1] be able to connect to the HMC-T2240 to the network and access it through its host name like any other computer.

The HMC-T2240's host name (machine name) is:

HMCT2240-<serial number>

where <serial number> is the 6 digit serial number shown when the unit turns on. You can also find the serial number by pressing and holding the front panel knob twice; the first press and hold takes you to the GPIB address screen and the second shows the serial number and firmware revisions. Press and hold the knob again to go back to the main display.

Reasons you might not want to use DHCP include:

1. A name server must also be set up. Some network configurations will support DHCP but not looking up devices by name.

- You are connecting a PC directly with a cable, and most PCs are not set up as DHCP servers. (The HMC-T2240 has auto-MDIX; crossover cables not required.)
- Routers or other network hardware do not support or are not configured for DHCP or name service.
- You have multiple equipment racks, each of which has its own router. Rather than refer to the instruments by name (which is different for each unit, or may not work) the IP Address is the same for each instrument within a rack so the same software with hard coded IP Addresses can be run on any rack. (This is commonly done with GPIB addresses. Note that IP Addresses must be unique within a subnet. The router within the rack defines each subnet. The subnets can be connected to the building network through the router or a second network connection in a PC, depending on the level of isolation desired for the equipment rack.)

2.3.1.1 Ethernet Addressing: Turning DHCP ON

To turn DHCP ON, send the following command through the USB or GPIB interfaces (assuming the Ethernet connection isn't working.)

SYSTem:COMMunicate:ETHernet:DHCP ON

You must power cycle the HMC-T2240 before this setting will take effect. You may also want to set the socket port number (discussed below) before power cycling.

You will also need to power cycle the HMC-T2240 when you move it from one network to another to re-start the DHCP address assignment process.

2.3.1.2 Ethernet Addressing: Static IP Address Assignment

If your address strategy is to use static IP addresses, turn DHCP off and set the ADDRess, NETMask, and GATeway. (You will need to substitute the correct values for your network.)

```
syst:comm:eth:DHCP      OFF
syst:comm:eth:ADDRess   192.168.1.105
syst:comm:eth:NETMask   255.255.255.0
syst:comm:eth:GATeway   192.168.1.1
```

You must power cycle the HMC-T2240 before these settings will take effect. You may also want to set the socket port number (discussed below) before power cycling.

2.3.1.3 Telnet Example

The following example comes from running the Windows telnet client from the command line. Replace the **000000** with the actual serial number from your unit. You can also type in the IP Address instead of **hmcT2240-000000**.

```
C:\>telnet hmcT2240-000000
*=====================================================*
*          Hittite T2240 Embedded Telnet Server                *
*=====================================================*
*idn?
Hittite,HMC-T2240,000000,1.8 5.3
```

syst:comm:eth:dhcp?
1
syst:comm:eth:port?
56789
syst:comm:eth:addr?
10.0.0.3
syst:comm:eth:netm?
255.255.255.0
syst:comm:eth:gat?
10.0.0.1

(Press Ctrl-C to get out of a command line Telnet session.)

You can also control the RF hardware with any of the normal SCPI commands.

See Also: **Example: Network configuration with DHCP OFF, Example: Network Configuration with DHCP ON**

2.3.3 Sockets

The HMC-T2240 Socket interface is the connection to use from a program. It has the same capability as a USB connection.

In addition to the Host Name or IP Address, you need to know the socket port number.

SYSTem:COMMunicate:ETHernet:PORT?
56789
SYSTem:COMMunicate:ETHernet:PORT 65432

The range of legal PORT values is 0 to 65535. Pick a value consistent with your existing applications. If you do not already have a standard port number, pick one from the range 49152-65535. Lower number ports may already be in use (23 = Telnet) or be used in the future (80 = HTTP, 111 = VXI-11, 5044 = LXI, for example) or may be used by common network protocols. (See http://www.iana.org/assignments/port-numbers for details.)

You must power cycle the HMC-T2240 to make this change take effect.

2.3.3.1 Example: Sockets with VISA drivers

The following code opens a socket connection to an HMC-T2240 and reads the ***IDN?** string. (This code was tested with LabWindows 8.1.)

```
#include <ansi_c.h>
#include <cvirte.h>
#include <visa.h>

int main (int argc, char *argv[])
{
   char cmd[] = "*idn?\n";
   char readbuf[256];
   int byt;
   int err;
   ViSession vihdl = 0;
   int Rsrc=0;
```

```
    char constring[128];
    unsigned SerialNum = 0;
    unsigned short PortNum = 54321;
    char IPAddr[] = "10.0.0.2";

    viOpenDefaultRM (&Rsrc);
//    sprintf(constring, "TCPIP::HMCT2240-%06u::%u::SOCKET", SerialNum, PortNum);
    sprintf(constring, "TCPIP::%s::%u::SOCKET", IPAddr, PortNum);
    viOpen(Rsrc, constring, 4, 0, &vihdl);
    viSetAttribute(vihdl, VI_ATTR_TERMCHAR_EN, VI_TRUE);
    viSetAttribute(vihdl, VI_ATTR_TMO_VALUE, 10000);
    viSetAttribute(vihdl, VI_ATTR_TERMCHAR, 0x0A);
    viSetAttribute(vihdl, VI_ATTR_SEND_END_EN, VI_TRUE);
    viSetAttribute(vihdl, VI_ATTR_IO_PROT, VI_PROT_NORMAL);

    err = viWrite(vihdl, cmd, sizeof(cmd)-1/*ignore NULL*/, &byt);
    err = viRead (vihdl, readbuf, 255, &byt);
    readbuf[byt] = 0;                          // input not null terminated
    printf("%s\n", readbuf);
    viClose (vihdl);
    return err;  // set breakpoint here to prevent output window from going away
}
```

2.3.4 Ethernet Security

The HMC-T2240 is intended to be used within a "safe" network environment such as a company network or a private subnet within an equipment rack.

The HMC-T2240 does NOT have password protection or other mechanisms to prevent unauthorized access.

The HMC-T2240 does not run Windows or any UNIX variant and does not support firmware updates over Ethernet so it is not considered vulnerable to infection by viruses.

3.0 SCPI Command Reference

3.1 SCPI Conformance Information

The HMC-T2240 command set is based on the SCPI standard, version 1999.0, but is not SCPI compliant because of the following:

- **SYSTem:VERSion?** - SCPI version - not compliant → not implemented
- **UNIT:POWer** – Only dBm supported.
- This document does not meet all SCPI documentation requirements.

3.2 SCPI Tips

The following tips may be helpful for understanding the SCPI instrument behavior.

- SCPI commands are not case sensitive. ***IDN?** and ***idn?** are considered the same command.
- CAPitalized letters indicate the "short form" of a command. **FREQ** is the short form of **FREQuency.**
- Words in [brackets] are optional. OUTPut[:STATe] means you can use OUTPut:STATe if you want to, but OUTPut or OUTP is sufficient. The leading [SOURce:] in front of **FREQuency**, **POWer**, and **SWEep** is also optional.
- Commands ending in question marks (**?**) are called *queries* and return results.
- Separate multiple commands on a line with semicolons (;). **FREQ 3e9;POW -3.3**; **OUTP ON** is a legal command. Multiple queries on the same line result in all the query results coming back on the same line, separated by semicolons. **FREQ?;POW?;OUTP?** returns **3000.00e6;-3.3;1**. (The trailing "1" corresponds to Boolean "true" for **ON**; 0 means **OFF**.)

  When you specify a "**:**" for a command, subsequent commands on the same line are expected to start at the same level of the command tree as the previous command. This allows you to say **FREQ:STAR 1e9;STOP 2e9;STEP 10e6** without repeating FREQ every time. However, to change from a command with one command path ([SOURce:]FREQuency) to another (SWEep), it is necessary to start the following command with a ":" to indicate that the next command is starting as if it were the first command. **FREQ:MODE SWE; SWE:DWEL 0.1s** results in the error **-113,"Undefined header"** because there is no command "FREQ:SWE:DWEL." **FREQ:MODE SWE; :SWE:DWEL 0.1s** works as expected.
- If you send a command and it does not affect the hardware or return a value, check for errors. **SYST:ERR?** will return an error message or **0,"No error"** if there are no errors left to read back. You can read back multiple errors at one time with **SYST:ERR:ALL?**.
- Input lines are limited to 256 characters, including the carriage return and/or line feed, for SW Ver <=2.2. SW Ver >=2.5, arbitrarily long input lines are supported.
- Output from any one query is limited to 256 characters, including the carriage return and/or line feed, or the ";" separating it from the output of another query on the same line. If multiple queries are on the same line, the total output on one line can exceed 256 characters, up 256 characters per query. An *important exception* is **SYST:ERR:ALL?** which can return 10 error messages of up to 256 characters each plus **-350, "Queue overflow"** for a total of 2583 characters from one query. Another **important exception** is LIST queries, which can return arbitrarily long strings.
- The only "overlapped" operation supported by the HMC-T2240 is sweeping. All other operations must complete their execution (but not necessarily hardware settling or having their results read back) before the next command can execute. If a command takes time to execute and you need to synchronize with other instruments, use ***OPC?** or ***OPC**. To delay the next command (sent only to the HMC-T2240) until a sweep completes, use ***WAI**.
- Expressions (**FREQ 3GHz** + **10 kHz**, for example) are not supported.

3.3 SCPI Command Reference

This section describes the commands supported by the HMC-T2240.

3.3.1 ABORt - Stop Sweeping

ABORt stops a running sweep.

If **INITiate:CONTinuous ON**, the sweep will restart (**INITiate**) immediately; turn **INITiate:CONTinuous OFF** before issuing **ABORt** to prevent sweep from restarting.

ABORt will not interrupt ***OPC?** or ***WAI**. See **SDC**.

ABORt is an event and does not have a query form.

3.3.1.1 Example: Shut Off a Continuous Sweep

INIT:CONT OFF;:ABOR

3.3.2 *CLS - Clear Status

*CLS clears the following status registers:

- **ESR** (488.2 Standard Event Status Register)
- **STATus:OPERation:EVENt**
- **STATus:QUEStionable:EVENt**

It also empties the Error Queue and clears the flag set by ***OPC**.

See Also: ***RST, STATus:PRESet, SDC, *ESR?**.

***CLS** is an event and does not have a query form.

3.3.2.1 Example: Reset Status

To "fully" reset the instrument according to SCPI-99 Vol 2-20.2 (PRESet)

***CLS;*SRE 0; *ESE 0; STATus:PRESet**

Note that ***RST** is required if you also want to reset hardware state.

Note that this does not affect LISTs. To clear all of the lists use:

LIST:FREQ;POW;OUTP;DWEL;SEQ

3.3.3 DCL - Device Clear - See SDC

DCL is a GPIB (488.1) command used to gain control of the instruments on the bus. See **SDC**.

3.3.4 DIAGnostic:BATTery:CYCLes? - Battery Charge/Discharge Cycles

Reads back the number of battery charge/discharge cycles, 0 to 65,534. This information may be used to determine when to replace a battery.(T22XXB Only). Software Version ≥ 2.3.

3.3.4.1 Example: DIAGnostic:BATTery:CYCLes?

diag:batt:cycl?
12,13 (Number of Charge/Discharge Cycles 12 - Battery 1, 13 - Battery 2)

3.3.5 DIAGnostic:BATTery:PCHarge? - Battery Percent Charge

Reads back the percent charge for each battery, 0 to 100. (T22XXB Only). Software Version ≥ 2.3. If only one battery is installed, the missing battery is reported as 0.

3.3.5.1 Example: DIAGnostic:BATTery:PCHarge?

diag:batt:pch?
100,98 (Percent Charge 100% Battery 1, 98% Battery 2)

3.3.6 DIAGnostic:BATTery:SNUMber? - Battery Serial Number

Reads back the serial number for each battery. (T22XXB Only). Software Version ≥ 2.3. If only one battery is installed, the missing battery is reported as 0.

3.3.6.1 Example: DIAGnostic:BATTery:SNUMber?

diag:batt:snum?
210,0 (Serial Number Battery 1 = 210, Battery 2 = missing)

3.3.7 DIAGnostic:BATTery:TCHArge? - Battery Time To Charge

Reads back the estimated time to charge the battery or batteries. (T22XXB Only). When the battery controller switches between charging and discharging, it will typically take several seconds for this data to stabilize. This data is only available when the unit is charging the battery. Software Version ≥ 2.3.

3.3.7.1 Example: DIAGnostic:BATTery:TCHArge?

diag:batt:tch? (While charging batteries)
6,30,00 (6 hours, 30 minutes, 0 seconds)

diag:batt:tch? (While battery is being discharged)
0,00,00

3.3.8 DIAGnostic:BATTery:TREMaining[?] - Battery Time Remaining

Reads back the estimated operating time remaining while operating from battery power. (T22XXB Only). When the battery controller switches between charging and discharging, it will typically take several seconds for this data to stabilize. This data is only available when the unit is operating from the battery. Software Version ≥ 2.3.

3.3.8.1 Example: DIAGnostic:BATTery:TREMaining

diag:batt:trem? (While operating from batteries)
4,00,00 (4 hours, 0 minutes, 0 seconds)

diag:batt:trem? (While battery is being charged)
99,59,59

3.3.9 DIAGnostic:COMPatibility[?] - Compatibility Mode

DIAGnostic:COMPatibility OFF|T2100 allows an HMC-T2200 series synthesizer to emulate an HMC-T2100 synthesizer.

The following are effected:

- **FREQuency** - minimum, maximum, and resolution
- **POWer** - minimum, maximum
- **SWEep:DWELI** - minimum

You must power cycle the unit to cause changes to **DIAG:COMP** to take effect.

DIAGnostic:COMPatibility? is the query form; it returns OFF or T2100.

DIAG:COMP is supported by the HMC-T2220/HMC-T2240 SW Version 2.2 and later.

See Also: **FREQuency:RESolution**.

3.3.9.1 Example: DIAGnostic:COMPatibility

diag:comp off
<<< power cycle >>>
***idn?**
Hittite,HMC-T2240,000010,2.2 4.4
freq? min; freq? max
10000000;40000000000
pow? min; pow? max
-60.0;30.0
swe:dwel? min; dwel? max
0.000300;4294.967044
freq:resolution?
1
diag:comp T2100
<< power cycle >>
***idn?**
Hittite,HMC-T2100,000010,2.2 4.4
freq? min; freq? max
10.00e6;20000.00e6
pow? min; pow? max
-36.0;27.0
swe:dwel? min; dwel? max
0.000100;4294.967044
freq:resolution?
0.01e6
diag:compatibility?
T2100

3.3.10 DIAGnostic:MACaddress?

Read the MAC Address from the unit.

MAC addresses are used for Ethernet communication. The MAC Address is programmed at the factory, and cannot be changed, but it may be helpful to read back for debug purposes.

Your router may also have a utility which will tell you the MAC address of everything connected to it.

See Also:

SYSTem:COMMunicate:ETHernet:ADDRess,SYSTem:COMMunicate:ETHernet:DHCP

3.3.10.1 Example: Read theMAC Address from a HMC-T2240

DIAG:MAC?
00242a987654

3.3.11 DISPlay:ENABle[?] - LCD ON/OFF

DISPlay:ENABle OFF causes the following to appear on the front panel LCD:

DISPLAY IS
CURRENTLY OFF

The display can be turned back on with **DISP: ENAB ON**.

The query form is **DISPlay:ENABle?** which returns 1 or 0.

The reset (***RST**) state is ON. Because the **SECurity** commands are not implemented, the display will go to a different state whenever ***RST** is issued, per SCPI-99. However the reset state of **DISP:TEXT:STAT** is not defined by **SCPI,** so ***RST** will not change it, and any text written with **DISP:TEXT** will remain visible even if ***RST** is used.

Turning the display off does not particularly improve performance.

See Also: **DISPlay:[WINDow:]TEXT[:DATA], DISPlay:[WINDow:]TEXT:STATe**

3.3.11.1 Example: Preventing the display from changing at *RST

DISP: ENAB OFF

DISP: TEXT:STAT ON

3.3.12 DISPlay:[WINDow:]TEXT:CLEar

DISPlay:TEXT:CLEar causes the front panel LCD to go blank.

There is no query form.

There is no effect from ***RST**.

See Also: **DISPlay:[WINDow:]TEXT[:DATA]**

3.3.13 DISPlay:[WINDow:]TEXT[:DATA]

DISP:TEXT "message" displays a message on the front panel LCD.

The LCD is 20 rows wide and 2 rows high. Text which goes beyond the end of the line will be wrapped. To show text on the second row, without overwriting the first row, use **DISPlay:[WINDow:]TEXT:LOCate**.

The query form **DISPlay:[WINDow:]TEXT[:DATA]?** may give incomplete or out of date results if the display has not had an opportunity to update, or is in the middle of an update.

The text on the LCD is not affected by ***RST** when **DISP:TEXT:STAT ON**.

3.3.13.1 Example: Front Panel LCD control

```
DISP:ENAB OFF
DISP:TEXT?
        DISPLAY IS              CURRENTLY OFF
DISP:ENAB ON
DISP:TEXT?
10,005,000,000 R    I-60.0dBm*              OFF
DISP:TEXT:CLEar
DISP:TEXT?
10,005,000,000 R    I-60.0dBm*              OFF
```

Display stays in normal mode unless you turn TEXT on.

```
DISP:TEXT:STAT ON
DISP:TEXT?

DISP:TEXT "LO Source, Pin 7"
DISP:TEXT?
LOSource, Pin 7
DISP:TEXT:LOC 2,8
DISP:TEXT "Line 2"
DISP:TEXT?
LO Source, Pin 7            Line 2
DISP:TEXT "over"
DISP:TEXT?'
LO Source, Pin 7            Over 2
DISP:TEXT:LOC 1,1
DISP:TEXT "Line 1"
DISP:TEXT?
Line 1rce, Pin 7            Over 2
```

It is generally more efficient to write the desired text with additional padding to overwrite any previous text than to **CLEar** and then write.

3.3.14 DISPlay:[WINDow:]TEXT:LOCate

DISPlay:[WINDow:]TEXT:LOCate <row>,<col> positions the cursor for the next write with **DISP:TEXT**.

<row> **row number, 1 or 2**

<col> **column number, 1 to 20**

The cursor is not moved by **DISP:TEXT**.

The query form is **DISPlay:[WINDow:]TEXT:LOCate?**, which returns **<row>,<col>**.

The **TEXT** mode cursor position is not affected by ***RST**.

See Also: Example: Front Panel LCD control.

3.3.15 DISPlay:[WINDow:]TEXT:STATe

DISPlay:[WINDow:]TEXT:STATe {ON|OFF} changes the front panel LCD to text mode so the output of **DISPlay:[WINDow:]TEXT or DISPlay:[WINDow:]CLEar** is visible or not.

The value of **DISP:TEXT:STATe** is not affected by ***RST**.

The query form, **DISPlay:[WINDow:]TEXT:STATe?**, returns 1 or 0.

See Also: **DISPlay:[WINDow:]TEXT**, Example: Front Panel LCD Control.

3.3.16 *ESE[?] - Event Status Enable

***ESE <mask>** determines which of the bits in the Standard Event Status Register (**ESR**) are summarized by bit 5 of the Status Byte (STB.)

<mask> is in the range 0 to 255.

Query Form: **ESE?**.

***ESE?** is affected by **FORMat:SREGister.**

Reset Value: 0

See Also: ***STB?**, ***ESR?, *SRE**

3.3.16.1 Example: Errors Propagate to STB

Set standard Event Status Enable and read it back.

```
*ese #h3c
*ese?
60
form:sreg hex
*ese?
#H3C
```

Verify STB (standard STatus Byte) is clear.

```
*stb?
#H00
```

Generate an error and verify it shows up in STB.

```
oops
*stb?
#H24
```

Reading the error message clears the Error/Event Queue bit (0x04.).

```
syst:err:all?
-113, "Undefined header; oops"
*stb?
#H20
```

The ESR summary bit will remain set until ESR is read..

```
*esr?
```

The msb of the ESR is Power On. This will be set after the unit is turned on.

```
#HA0
```

Reading ESR clears ESR and the ESR summary bit..

***esr?**
#H00
***stb?**
#H00

3.3.17 *ESR? – Event Status Register

***ESR?** reads the Standard Event Status Register and clears it.

The **ESR** is "sticky" in that it keeps its bits high until they are cleared even if the condition that caused them to be set is no longer true.

The result is in the range 0 to 255.

There is no command to set the ESR.

Reading the ESR clears it. To clear the ESR without reading its contents, see ***CLS**.

***ESR?** is affected by **FORMat:SREGister**.

See Also: ***ESE**, ***STB?**, **FORMat:SREGister**

3.3.17.1 The bits in the ESR:

(0 is least significant)

0. **Operation Complete (See *OPC)**
1. **Unused (The HMC-T2240 cannot be a GPIB controller.)**
2. **Query Error**
3. **Device Dependent Error**
4. **Execution Error**
5. **Command Error**
6. **User Request (Front panel knob pressed and held, whether in local lockout/KLOCk or not.)**
7. **Power On (Set when unit turns on. Cleared by first *ESR? or *CLS.)**

The "Error" bits indicate that errors, broken down by SCPI category, have happened. Read the actual error from the error queue using **SYSTem:ERRor[:NEXT]?**.

3.3.17.2 Example: Detect an error

a_bad_command
***esr?**
32

^^--- Bit 5 → Error. Find out which one:

SYSTem:ERRor?
-113 "Undefined header; a_bad_command"

This is the SCPI defined error for an unrecognized command.

3.3.18 FORMat:SREGister[?] – Decimal/Hex/Binary

FORMat:SREGister <format> selects the output format for status registers.

<format> can be:

- **ASCii** – Base 10 (Decimal, Reset value)
- **HEXadecimal** – Base 16 (#H0123ABCD)
- **BINary** – Base 2 (#B010101)

The query form, **FORMat:SREGister?**, returns **ASC**, **HEX**, or **BIN**.

It has no effect on Frequency, Power, or "Boolean" values.

3.3.18.1 Example: FORMat:SREGister

```
form:sreg hex
*ese?
#H3C
form:sreg bin
*ese?
#B111100
```

3.3.19 FREQuency – See [SOURce:]FREQuency

SCPI specifies the optional "**SOURce**" keyword before **FREQuency**.

3.3.20 GET – Group Execute Trigger (GPIB only)

GET allows multiple instruments to be triggered by the same trigger command over GPIB.

GET is encoded with the ATN signal asserted, not sent as a string. See your GPIB Controller software documentation for the best way to send **GET**.

GET does not have to go through the SCPI parser so it can be processed with much less latency than ***TRG** or **TRIGger**. However, in order to comply with the GPIB FIFO order requirement for commands, the HMC-T2240 must not be busy when **GET** is sent. If it is busy, **GET** will be placed in the input buffer similar to a normal string command and the trigger will be delayed relative to when it would normally be expected.

To ensure the HMC-T2240 is ready for **GET** to be sent, send ***OPC?** and wait for the **1** to come back before sending **GET**. This ensures any previous commands have been received, processed, and completed.

The **TRIGger:SOURce** must be set to **BUS** for **GET** to be received properly.

GET is an event so there is no query form.

See Also: ***TRG, Example: GPIB Parallel Poll with GET**, **TRIGger:SOURce**

3.3.21 *IDN? – Identify

*IDN? produces an identification string including the manufacturer's name (Hittite), the model (HMC-T2240), the serial number, followed by SW revision, followed by HW (FPGA) revision, and last (optional) is the battery controller revision.

3.3.21.1 Example: *IDN?

***IDN?**

Hittite,HMC-T2240,000006,X.X X.X X.X

3.3.22 INITiate:CONTinuous[:ALL][?] – Start Sweeping

INITiate:CONTinuous[:ALL] {ON|OFF} enables sweeping so that sweeping will continue until this command, ***RST**, **SYSTem:PRESet**, **FREQ:MODE** or **POW:MODE** is used to turn it off again.

Turning **INIT:CONT OFF** will allow the currently running sweep to terminate normally. To stop the current sweep immediately, use **ABORt** or change the **MODE**. Note that if **INIT:CONT** is left **ON** after **ABORt**, the sweep will restart immediately.

Executing ***OPC?** or ***WAI** with **INITiate:CONTinuous ON** will cause the unit to appear to hang. It will continue to sweep, but any commands after ***OPC?** or ***WAI** will be ignored. See **SDC**.

The query form, **INITiate:CONTinuous[:ALL]?** returns 0 if continuous sweeping is **OFF** and 1 if it is **ON**.

If **TRIGger:SOURce** is not **IMMediate**, each sweep will wait for a trigger.

If **SWEep:COUNt** is N, each trigger will result in N sweeps. (Note: Triggers which happen while a sweep is in progress are ignored or result in errors.) If **INIT:CONT OFF** is used, the total number of sweeps will be a multiple of N.

FREQ:MODE must be **SWEep** for the unit to begin sweeping frequency or **POW:MODE** must be **SWEep** to begin sweeping power.

3.3.22.1 Example: Start Continuous Sweep

freq:star 1e9;stop 2e9;step 100e6;mode swe
swe:dwel 0.1
init:cont on
<let sweep run>
init:cont off

3.3.23 INITiate[:IMMediate][:ALL] – Start a Single Sweep

INITiate starts a sweep.

If **TRIGger:SOURce** is **IMMediate**, the sweep begins when this command is issued. Otherwise, it waits for a trigger.

If **SWEep:COUNt** is N, each trigger or **INIT** will result in N sweeps. (Note: Triggers which happen while a sweep is in progress are ignored or result in errors.)

FREQ:MODE or **POWer:MODE** must be **SWEep** or an error will be generated.

Sweep parameters, such as the **FREQ:STARt** and **FREQ:STOP**, are sampled at INIT time. If **INIT:CONTinuous ON**, each time the sweep re-starts, the parameters are re-sampled. An **ABORt** forces a continuous sweep to re-**INIT**, if you don't want to wait for the sweep to complete normally.

INITiate is an event; there is no query form.

3.3.23.1 Example: Start Sweep with Trigger

This initiates a sweep, but the sweep does not actually begin until the ***trg** command is received.

```
freq:mode sweep
trig:source bus
init
*trg
```

3.3.24 *IST? - Individual Status

*IST? reads the IEEE 488.1 (GPIB) parallel poll Individual Status flag.

1 *ist* is true

0 *ist* is false

There is no command to set *ist* directly; use ***PRE** to select the bits *ist* monitors.

***IST?** can be used from any interface, GPIB or otherwise.

See Also: ***PRE, PPE, *STB?**

3.3.25 LIST - See [SOURce:]LIST

SCPI specifies the optinal **"SOURce"** keyword before **LIST**.

3.3.26 OUTPut:BLANking[?] – Disable RF Output While Frequency Changing

OUTPut:BLANking {ON|OFF} controls whether the output is attenuated while the output frequency is changing. This may reduce undesirable spectral output when executing any of the commands which change frequency or during a frequency sweep. This does, however, result in more changes to the RF attenuators, which may increase power settling time. It does not affect frequency settling time.

OUTPut:BLANking ON Attenuate RF while changing frequency

OUTPut:BLANking OFF Do not cause extra changes to RF signal path while changing frequency.

OUTPut:BLANking? Returns 0 for OFF and 1 for ON.

Blanking is only available for SW Ver ≥ 2.3 and FPGA Ver ≥ 4.6.

SW Ver 2.4 and later: OUTPut:BLANking is a non-volatile parameter. Once it is set, it will stay set until it is changed by another OUTP:BLANking command or from the front panel. It is not affected by *RST.

SW Ver 2.3: OUTPut:BLANking is set to OFF by *RST or by power cycling the unit.

This is a Hittite extension and is not portable. Do not use if compatibility with other instruments is required.

3.3.26.1 Example: Blanking

```
OUTP:BLAN ON
OUTP:BLAN?
1
OUTP:BLAN OFF
OUTP:BLAN?
0
```

3.3.27 OUTPut[:STATe][?] – RF Output On/Off

OUTput **{OFF|ON}** controls the RF Output ON/OFF function. Note that enabling and disable the RF Output affects attenuators and/or frequency in order to minimize RF leakage.

OUTPut[:STATe]? returns **0** for **OFF** and **1** for **ON**.

Specify OUTput next to FREQuency and POWer on the same line to enable optimization.

3.3.27.1 Example: RF On/Off State

To Set and Query the RF output state:

OUTPut ON
OUTP?
1
OUTP OFF
OUTP?
0

3.3.27.2 Example: Output with Frequency and Power

Specifying Frequency and Power with Output allows some optimization as all three commands affect the attenuators. Order is not important, but they should not be separated by other commands.

freq 16.384GHz;pow 17.2dBm;outp on

3.3.28 *OPC – Operation Complete Command

*OPC sets a flag which causes the ESR Operation Complete bit (bit 0) to be set when a sweep is done. If no sweep is initiated, the ESR Operation Complete bit is set immediately.

ABORt, ***CLS**, ***RST**, **SDC**, and **STATus:PRESet** clear this operation whether the sweep completes or not.

***OPC** does not have a query form that reads back the ***OPC** flag. See ***OPC?**.

See Also: ***ESR?**, ***OPC?**, ***WAI**

3.3.28.1 Example: Operation Complete

***ESR?**
#H00

Status is clear

***OPC**
***ESR?**
#H01

Status is Operation is Complete

INIT:CONT ON
***OPC**
***ESR?**
#H00

Status is clear – sweep still running

```
INIT:CONT OFF
*ESR?
#H01
```

Status is Operation is Complete – It took long enough to type "***ESR?**" that the last sweep completed.

3.3.29 *OPC? – Operation Complete Query

***OPC?** returns a "**1**" when the current operation is complete. Since sweeps are the only "overlapped" operations, ***OPC?** returns immediately unless a sweep has been initiated.

Since ***OPC?** blocks other operations from executing until the sweep completes, issuing ***OPC?** when **INIT:CONTinuous ON** will cause the HMC-T2240 to appear to hang. (The sweep will continue, but commands which are sent will be queued up waiting for ***OPC?** to complete.) To break this deadlock, see **SDC**.

3.3.29.1 Example: Monitoring Sweep Status

```
swe:dwel 0.1
freq:star 1e9;stop 2e9;step 100e6;mode swe
init;:stat:oper:cond?;*opc?;:stat:oper:cond?
8;1;0
```

The leading "8" came out noticeably before the ";1;0", where the "1" is from ***OPC?** and the "0" is from **STATus:OPERation:CONDition?**.

3.3.30 POWer – See [SOURce:]POWer

POWer is specified under the SCPI optional keyword **SOURce**.

3.3.31 PPE - Parallel Poll Enable (GPIB Only)

PPE is a GPIB encoded command for configuring Parallel Polls. It is not available on any interface other than GPIB; see ***IST?** if you want to read *ist* over USB or Ethernet.

Parallel Polls allow the GPIB controller to read status from multiple instruments simultaneously. Instruments are configured to write their *ist* information to a particular bit of the 8 bit wide bus which operates in a wired-OR manner so that multiple instruments can contribute to each bit.

Unlike normal queries, parallel polls and serial polls do not significantly slow down the HMC-T2240 or add jitter during sweeps.

To disable parallel poll capability, send 0x70 with ATN asserted.

To enable parallel poll capability, send 0x60 ORed with **S** and **P** while ATN is asserted.

S is 0x00 or 0x08 and controls the Sense or polarity of the signal read from the GPIB.

S	ist	Value on GPIB
0	0	Low Voltage - Logic 1
0	1	Unasserted - Logic 0
1	0	Unasserted - Logic 0
1	1	Low Voltage - Logic 1

P is the bit on the bus to write *ist* to in the range 0 to 7. 0 is the least significant bit and 7 is the most significant bit.

You will need to consult your software or controller documentation for how to set up parallel polls, but an example for the National Instruments™ "ib" functions follows.

There is no query form of **PPE**.

See Also: ***PRE, *IST?, *STB?**

3.3.31.1 Example: GPIB Parallel Poll with GET

```
char ppr = 0;       // Parallel Poll Response

// Set up a triggered sweep
const char ccc[] =
    "trig:sour bus;"
    ":freq:star 10e6;stop 1.01e9;step 1e6;mode swe;"
    "*PRE #h200;"   // Parallel Poll ist - Waiting for Trigger
    ":init"
    ;
// Will stay in "waiting for trigger" state until send *TRG or TRIG
// Send SCPI command to T2240
ibwrt(Device, ccc, sizeof(ccc)-1/*strip NULL*/);

// set up parallel poll on lsb (bit 1 for "1 oriented" GPIB)
// 0x60 --> PPE + configure
// 0x08 --> positive logic
// 0x00 --> lsb
ibppc(Device, 0x60 | 0x08 | 0x00);
// Run parallel poll
ibrpp(Device, &ppr);
if (0x01 != ppr) {
    printf(
        "Parallel poll for Waiting for Trigger failed: Got 0x%02x; expected 0x01; ibsta = 0x%02x\n",
        ppr, ibsta);
}

// Send Trigger: GET – Group Execute Trigger
ibsta_val = ibtrg(Device);

// Should no longer be waiting for trigger.
// Parallel poll to verify
ibrpp(Device, &ppr);
if (0x00 != ppr) {
    printf(
      "Parallel poll for NOT Waiting for Trigger failed: Got 0x%02x; expected 0x00; ibsta = 0x%02x\n",
        ppr, ibsta);
}
```

3.3.32 *PRE[?] - Parallel Poll Register Enable

***PRE <mask>** (Parallel Poll Register Enable Command) selects the bits to summarize in ist for IEEE 488.1 (GPIB) parallel polls, similar to ***SRE** for service requests.

<mask> is in the range 0 to 65535. The low 8 bits come from STB. The high 8 bits are instrument specific.

0. not currently used
1. not currently used
2. Error Queue summary
3. **STATus:QUEStionable** summary
4. MAV – Message Available
5. Event Status Register (***ESR?**) summary
6. RQS – Service Request
7. **STATus:OPERation** summary
8. **STATus:OPERation:CONDition** SWEeping
9. **STATus:OPERation:CONDition** Waiting for TRIGger

10-15. not currently used

***PRE** is cleared on power cycle and by writing 0 into it.

***PRE?** is the readback form. It is affected by **FORMat:SREGister**.

The GPIB bit that ist is written to during parallel polls, and whether it is inverted or not, is determined by **PPE**. (GPIB PP1 – remote Parallel Poll configuration.)

See Also: ***IST?, PPE, *STB?, *SRE**

3.3.33 *RST - Reset

***RST** puts the instrument into a consistent state.

Command	Reset Value
DISPLay:ENABLE	ON
FORMat:SREGister	ASCii
INITiate:CONTinuous	OFF
OUTPut[:STATe]	OFF
OUTPut:BLANking	OFF
***OPC**	Clears flag to set Operation Complete when sweep done
[SOURce:]FREQuency:CENTer	MID BAND
[SOURce:]FREQuency[:FIXed|:CW]	10.005 GHz (mid band)
[SOURce:]FREQuency:MODe	CW (sweep off)
[SOURce:]FREQuency:STEP	10 kHz (min)
[SOURce:]FREQuency:SPAN	MAX
[SOURce:]FREQuency:STARt	10 MHz (min)
[SOURce:]FREQuency:STOP	MAX
[SOURce:]LIST:COUNt	1
[SOURce:]LIST:DIRection	UP
[SOURce:]LIST:GENeration	DSEQuence

Command	Reset Value
[SOURce:]POWer:CENTer (SW Ver < 2.3)	-15 dBm
[SOURce:]POWer:CENTer (SW Ver ≥ 2.3)	-60 dBm
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]	-60 dBm
[SOURce:]POWer:MODe	FIXed (sweep off)
[SOURce:]POWer:STEP	0.1 dB (min)
[SOURce:]POWer:SPAN (SW Ver < 2.3)	90 dB
[SOURce:]POWer:SPAN (SW Ver ≥ 2.3)	0 dB
[SOURce:]POWer:STARt	-60 dBm
[SOURce:]POWer:STOP (SW Ver < 2.3)	+30 dBm
[SOURce:]POWer:STOP (SW Ver ≥ 2.3)	-60 dBm
[SOURce:]SWEep:MODE	AUTO
SWEep:COUNt	1
SWEep:DIRection	UP
SWEep:DWELl	3ms
TRIGger:DELay	0
TRIGger:SLOPE	POSitive
TRIGger:SOURce	IMMediate

3.3.34 SDC – Selected Device Clear – Control Code

SDC is a GPIB (488.1) command used to gain control of an instrument. It breaks out of situations like ***WAI** and ***OPC?** waiting for a sweep which will never terminate because it will never be triggered, or is set to sweep continuously, or which will take longer than the user is willing to wait.

For a control interface other than GPIB, **SDC** is encoded as Ctrl-D, or decimal 4, rather than as a character string, and must be sent on a line by itself. For GPIB, it is encoded on the bus by the controller using control lines.

SDC clears the input and output FIFOs for the interface it is received on and exits any ***OPC?** or ***WAI** commands which may be executing. It does not stop any sweeps which may be in progress or otherwise change instrument state. Status for ***OPC** and ***OPC?** is cleared.

SDC is an event; there is no query form.

When first taking control of a bus, **SDC** is usually followed by ***RST**.

3.3.34.1 Example: Using Ctrl-D to Regain Control

Set dwell time to 1 hour and issue ***OPC?**

swe:dwel 3600
freq:start 10MHz;stop 20GHz;step 10kHz;mode swe
init
***opc?**
***STB?**

Not responding…

<Ctrl-D><Enter>
***STB?**

0

Response came back right away.

stat:oper:cond?
8

Sweep is still running

abor
stat:oper:cond?
0

Sweep is stopped

3.3.35 [SOURce:]FREQuency:CENTer[?]

FREQuency:CENTer <freq> is used with **FREQuency:SPAN**, **FREQuency:STARt**, or **FREQuency:STOP** to specify the range of frequencies to sweep over.

<freq> is a frequency in **Hz** which can be written in scientific notation (1.23456e9) or with the standard units (**GHz, MHz, kHz, Hz**) as well as by writing out all the digits down to Hz (20000000000.0 for 20GHz) as an integral or floating point value. The text strings **MINimum** and **MAXimum** are also accepted.

Specifying **CENTer** with **FREQuency:SPAN**, **FREQuency:STARt**, or **FREQuency:STOP** next to it on the same line allows the sweep frequencies to be set without going through intermediate states which could result in extraneous errors or confusing behavior.

The query form returns a value in Hz. It accepts **MINimum** and **MAXimum** as arguments.

3.3.35.1 Example: Center and Span

Center and Span:

FREQ:CENT 3e9;SPAN 2e9
FREQ:STARt?;STOP?
2000000000;4000000000

3.3.35.2 Example: Center and Start

FREQ:CENT 3e9;STAR 1e9
FREQ:STARt?;STOP?
1000000000;5000000000

The results of putting these commands on separate lines are different:

***rst**
FREQ:STARt?;STOP?
10000000;20000000000
freq:cent 3e9
FREQ:STARt?;STOP?
10000000;20000000000
syst:err?
250,"FREQ-Sweep Calculation ERROR; Calc By[FREQ-Center 3000000000 FREQ-Span

39990000000] outside of range [10000000,40000000000]"

freq:start 1e9
FREQ:STARt?;STOP?
1000000000;20000000000
freq:cent?
10500000000

3.3.35.3 Example: Center Frequency Min/Max

freq:cent? min
10000000
freq:cent? max
40000000000

3.3.36 [SOURce:]FREQuency[:FIXed|:CW][?] – Output Frequency/frequency

FREQuency <freq> specifies the RF output when the unit is not in sweep mode. (**FREQ:MODE CW** or **FREQ:MODE FIXed**) Setting **FREQ** in sweep mode results in an error according the the SCPI standard.

<freq> is a frequency in Hz which can be written in scientific notation (1.23456e9) or with the standard units (GHz, MHz, kHz, Hz) as well as by writing out all the digits down to Hz (20000000000.0 for 20 GHz) as an integral or floating point value. The text strings **MINimum**, **MAXimum, UP**, and **DOWN** are also accepted.

The query form returns a value in Hz. It accepts **MINimum** and **MAXimum** as arguments.

See Also: **FREQuency:MODE**, **FREQuency:STEP**

3.3.36.1 Example: Frequency Commands

freq 3.14159e9
frequency 2GHz
source:frequency:cw 20e9
sour:freq:fix 10MHz
freq:cw 20000000000.000
freq?
20000000000
freq:step 100MHz
freq down
freq?
19900000000
freq up
freq?
20000000000
freq minimum
freq?
10000000
freq maximum
freq?

40000000000
freq? min
10000000
freq? max
40000000000

3.3.37 [SOURce:]FREQuency[:FIXed|:CW]:STEP[:INCRement][?] – Frequency Step

FREQuency:STEP <step> specifies the step size for sweeps and the UP and DOWN keywords when used with the FREQuency command.

<step> is a frequency in **Hz** which can be written in scientific notation (1.23456e9) or with the standard units (**GHz, MHz, kHz, Hz**) as well as by writing out all the digits down to Hz (20000000000.0 for 20 GHz) as an integral or floating point value. The text strings **MINimum** and **MAXimum** are also accepted.

The query form returns a value in Hz. It accepts **MINimum** and **MAXimum** as arguments.

3.3.37.1 Example: Frequency Step

freq:step?
100000000
freq:step? min
1
freq:step? max
39990000000
freq:step 1234.56MHz

3.3.38 [SOURce:]FREQuency:MODE – Sweep Enable

FREQuency:MODE {CW|FIXed|SWEep|LIST} controls whether the HMC-T2240 is configured for a stepped sweep, an arbitrary list of frequencies, or for single frequency **(CW** or **FIXed)** operation.

The query form returns **CW, SWE,** or **LIST.**

The ***RST** state is **CW.**

Enabling either **FREQ:MODE LIST** or **[SOURce:]POWer:MODE LIST** is sufficient to enable LIST mode which can affect both **FREQuency** and **POWer** (as well as **OUTPut[:STATe]** and **[SOURce:] SWEep:DWELI.) LIST** mode requires SW Ver >= 2.5.

See Also: **[SOURce:]SWEep:LIST:FREQuency**

3.3.38.1 Example: FREQuency:MODE[?]

***rst**
freq:mode swe
swe:dwel min
init

Copied the following in separately to introduce a delay:

freq?
22370000

Copied the following in separately to introduce a delay:

freq:mode cw
freq?
44760000

Note that leaving sweep mode terminated the sweep immediately. ***WAI** could have been used to wait until the sweep completed.

3.3.39 [SOURce:]FREQuency:RESolution[?]

FREQuency:RESolution <step> forces all frequencies to be a multiple of **<step>**.

The reset value for **FREQ:RES** is:

Synthesizer	Frequency Resolution
HMC-T2100	10 kHz
HMC-T2200	1 Hz

unless the Synthesizer is a HMC-T2200 in compatibility mode (**DIAG:COMP T2100**), in which case the reset value becomes the HMC-T2100 reset value.

Changing **FREQ:RES** to a value greater than 10 MHz will affect the minimum frequency the unit can be programmed to.

FREQ:RES is primarily tested at 1 Hz and 10 KHz but values from 1 Hz to 9.999999999e9 can be programmed. Values like 3.333333333 MHz will be rounded to the nearest 1 Hz. This may result in unexpected behavior, particularly when the front panel knob is used to set the frequency. This will also change the **MAXimum** and **MINimum** frequency values.

For frequency resolutions which are multiples of 10 kHz, frequencies read back from the unit will be in HMC-T2100 format.

The query form is **FREQ:RES?**.

FREQ:RES is supported on the HMC-T2200 series for SW Version 2.2 and later.

See Also: **DIAGnostic:COMPatibility**

3.3.39.1 Example: Frequency Resolution

diag:comp?
T2100
freq:res?
0.01e6
freq:res 1
freq:res?
1
freq:res 2.7182818284e9
freq:res?
2718281828
freq?
freq?
10873127312
freq 10e6

syst:err:all?
200,"FREQuency out of range; 0 outside of range [271828128,19027972796]:
freq:res? min; res? max
1;9999999999

3.3.40 [SOURce:]FREQuency:SPAN[?]

FREQuency:SPAN <freq> is used with **FREQuency:CENTer**, **FREQuency:STARt**, or **FREQuency:STOP** to specify the range of frequencies to sweep over.

<freq> is a frequency in **Hz** which can be written in scientific notation (1.23456e9) or with the standard units (**GHz, MHz, kHz, Hz)** as well as by writing out all the digits down to Hz (20000000000.0 for 20 GHz) as an integral or floating point value. The text strings **MINimum** and **MAXimum** are also accepted.

Specifying **SPAN** with **FREQuency:CENTer**, **FREQuency:STARt**, or **FREQuency:STOP** next to it on the same line allows the sweep frequencies to be set without going through intermediate states which could result in extraneous errors or confusing behavior. If more than two of these commands are specified on one line next to each other the last two take effect.

The query form returns a value in Hz. It accepts **MINimum** and **MAXimum** as arguments.

See Also: **FREQuency:CENTer**

3.3.41 [SOURce:]FREQuency:STARt[?]

FREQuency:STARt <freq> is used with **FREQuency:STOP**, **FREQuency:CENTer,** or **FREQuency:SPAN** to specify the range of frequencies to sweep over.

<freq> is a frequency in **Hz** which can be written in scientific notation (1.23456e9) or with the standard units (**GHz, MHz, kHz, Hz)** as well as by writing out all the digits down to Hz (20000000000.0 for 20 GHz) as an integral or floating point value. The text strings **MINimum** and **MAXimum** are also accepted.

Specifying START with **FREQuency:CENTer**, **FREQuency:SPAN**, or **FREQuency:STOP** next to it on the same line allows the sweep frequencies to be set without going through intermediate states which could result in extraneous errors or confusing behavior. If more than two of these commands are specified on one line next to each other the last two take effect.

The query form returns a value in Hz. It accepts **MINimum** and **MAXimum** as arguments.

See Also: **FREQuency:CENTer**

3.3.42 [SOURce:]FREQuency:STOP[?]

FREQuency:STOP <freq> is used with **FREQuency:STARt**, **FREQuency:CENTer**, or **FREQuency:SPAN** to specify the range of frequencies to sweep over.

<freq> is a frequency in **Hz** which can be written in scientific notation (1.23456e9) or with the standard units (**GHz, MHz, kHz, Hz)** as well as by writing out all the digits down to Hz (20000000000.0 for 20 GHz) as an integral or floating point value. The text strings **MINimum** and **MAXimum** are also accepted.

Specifying **STOP** with **FREQuency:CENTer**, **FREQuency:SPAN**, or **FREQuency:STARt** next to it on the same line allows the sweep frequencies to be set without going through intermediate states

which could result in extraneous errors or confusing behavior. If more than two of these commands are specified on one line next to each other the last two take effect.

See Also: **FREQuency:CENTer**

3.3.43 [SOURce:]LIST:COUNt[?]

LIST:COUNt <number> determines the number of sweeps that happen for each **TRIGger/*TRG,** or **INIT** if **TRIGger:SOURce IMMediate.**

<number> is in the range 1 to 4294967295 or the symbols **MIN** or **MAX**. The *RST value is 1.

Requires SW Ver >= 2.5.

3.3.43.1 Example:LIST:COUNt

list:count?
1
list:count 37
list:count?
37
list:coun? max
4294967295
list:coun? min
1

3.3.44 [SOURce:]LIST:DIRection[?]

LIST:DIRection UP|DOWN controls the order in which a **LIST** (frequency list) is processed.

UP processes values in the order they were entered in **LIST:FREQ, LIST:POWer, LIST:OUTP,** etc.

DOWN processes values from the end of the list to the start.

The query form returns the string**s UP** and **DOWN.**

The ***RST** value is **UP.**

Requires SW Ver >= 2.5.

3.3.45 [SOURce:]LIST:DWELI[?]

LIST:DWEL [<number>]{,<number>} allows zero or more dwell times to be specified for list mode. The number of dwell times in the list must be 0, 1, or match the number of points in the other **LIST** fields, such as **LIST:FREQuency.** If no **LIST:DWELI** values are specified, **LIST:DWELI** has no effect for **LIST** mode execution. If **LIST:DWELI** has exactly one value, that value is used for the whole list. **<number>** may be left blank to indicate "no change" or may be the symbols **MAX** or **MIN.**

Blank dwell times allow **LIST:SEQuence** to refer to the same point multiple times with **DWELI** times set by other points. This handling of blank or "no change" values means **"LIST:DWEL 1.0,,,"** a dwell time of one second followed by four "no change" values rather than **"LIST:DWEL 1.0"** which **SCPI-99** specifies as acting like a list having the same number of points as the other **LISTs** with all the values the same.

The total time for a list mode sweep is the sum of the dwell times plus the 250usec frequency change time per point.

If no value is specified for **LIST:DWEL**, then the value of **[SOURce:]SWEep:DWELI** is used instead.

LIST:DWEL? returns an empty string ("\n") if the list is empty.

LISTs are not affected by ***RST. LISTs** are empty after power cycling.

Requires SW Ver >= 2.5.

3.3.45.1 Example: LIST:DWELI

```
list:dwel:poin?
0
list:dwel 1e-3,2e-3,     ,3e-3,4e-3
list:dwel?
0.001000,0.002000,,0.003000,0.004000
list:dwel:poin?
5
list:dwel
list:dwel?
                                        <<< Note blank line here
list:freq?;pow?;outp?;dwel?
;;;
list:dwel:poin?
0
list:dwel:poin? max
2048
list:dwel:poin? min
0
```

3.3.46 [SOURce:]LIST:DWELI:POINts?

LIST:DWEL:POIN? reads back the number of dwell times in **LIST:DWEL.** Specifying the arguments **MIN** or **MAX** reads back the minimum or maximum number of points. **[SOURce:]LIST:SEQuence** affects the maximum number of points usable at one time.

This is a query only; there is no command form.

Requires SW Ver >= 2.5.

3.3.47 [SOURce:]LIST:FREQuency[?]

LIST:FREQuency [<number>]{,<number>} allows zero or more **FREQuencies** to be specified for **LIST** mode. The number of frequencies in the list must be 0, 1, or match the number of points in the other **LIST** fields, such as **[SOURce:]LIST:POWer**. If no **LIST:FREQuency** values are specified, **LIST:FREQuency** has no effect for LIST mode execution. If **LIST:FREQuency** has exactly one value, that value is used for the whole list. **<number>** may be left blank to indicate "no change" or may be the symbols **MAX** or **MIN.**

Blank frequencies allow **LIST:SEQuence** to refer to the same point multiple times with frequency set by other points. This handling of blank or "no change" values means **"LIST:FREQ** 1.0GHz,,,,", a frequency of 1GHz followed by four "no change" values, rather than **"LIST:FREQ** 1.0GHz", which **SCPI-99** specifies as acting like a list having the same number of points as the other **LISTs** with all the values the same.

The total time for a list mode sweep is the sum of the dwell times plus the 250usec frequency change time per point. Dwell times may be set by **[SOURce:]LIST:DWELI or [SOURce:]SWEep:DWELI.**

LIST:FREQ? returns an empty string ("\n") if the list is empty.

LISTs are not affected by ***RST**. LISTs are empty after power cycling.

Requires SW Ver >= 2.5.

3.3.47.1 Example: LIST:FREQuency

list:freq 3e9,4e9,5e9, ,max,min
list:freq?
3000000000,4000000000,5000000000,,20000000000,10000000
list:freq:points?
6
list:freq:points? max
2048
list:freq:points? min
0
list:freq
list:freq?
list:freq:poin?
0

3.3.48 [SOURce:]LIST:FREQuency:POINts?

LIST:FREQuency:POINts? reads back the number of frequencies in **[SOURce:]LIST:FREQuency.** Specifying the arguments **MIN** or **MAX** reads back the minimum or maximum number of points**. [SOURce:]LIST:SEQuence** affects the maximum number of points usable at one time.

This is a query only; there is no command form.

0

Requires SW Ver >= 2.5.

3.3.49 [SOURce:]LIST:GENeration[?]

LIST:GENeration DSEQuence|SEQuence selects between executing the **LIST** mode points in the order they are specified in the list **(DSEQuence)** and executing them in the order specified by **[SOURce:]LIST:SEQuence (SEQuence.)**

The ***RST** state is **DSEQuence.**

The query form returns **DSEQ** or **SEQ.**

Requires SW Ver >= 2.5.

3.3.50 [SOURce:]LIST:OUTPut[?]

LIST:OUTPut [<number>]{,<number>} allows zero or more **OUTPuts** to be specified for **LIST** mode. The number of output states in the list must be 0, 1, or match the number of points in the other **LIST** fields, such as **[SOURce:]LIST:DWELI**. If no **LIST:OUTPut** values are specified**, LIST:OUTPu**t has no effect for **LIST** mode execution. If **LIST:OUTPut** has exactly one value, that value is used for the

whole list. **<number>** may be left blank to indicate "no change" or may be the symbols **ON** or **OFF.**

Blank **OUTPuts** allow **LIST:SEQuence** to refer to the same point multiple times with the output state set by other points. This handling of blank or "no change" values means **"LIST:OUTP ON,,,,"**, enables the output at point 1, followed by four "no change" values, rather than **"LIST:OUTP ON"**, which **SCPI-99** specifies as acting like a list having the same number of points as the other **LISTs** with all the values the same, turning the output **ON** at every point.

The total time for a list mode sweep is the sum of the dwell times plus the 250usec frequency change time per point. Dwell times may be set by **[SOURce:]LIST:DWELI** or **[SOURce:]SWEep:DWELI.**

LIST:OUTP? returns an empty string ("\n") if the list is empty.

LISTs are not affected by ***RST**. LISTs are empty after power cycling.

Requires SW Ver >= 2.5.

3.3.50.1 Example: LIST:OUTPut

```
list:outp on,off,off,on,,,,,,,,off,on,off
source:list:output?
1,0,0,1,,,,,,,,0,1,0
list:outp:poin?
14
list:outp:points? max
2048
list:output:poin? min
0
list:outp on
list:outp?
1
list:outp
list:outp?

list:outp:poin?
0
```

3.3.51 [SOURce:]LIST:OUTPut:POINts?

LIST:OUTPut:POINts? reads back the number of states in **[SOURce:]LIST:OUTPut.** Specifying the arguments **MIN** or **MAX** reads back the minimum or maximum number of points. **[SOURce:] LIST:SEQuence** affects the maximum number of points usable at one time.

This is a query only; there is no command form.

Requires SW Ver >= 2.5.

3.3.52 [SOURce:]LIST:POWer[?]

LIST:POWer [<number>]{,<number>} allows zero or more POWers to be specified for **LIST** mode. The number of powers in the list must be 0, 1, or match the number of points in the other **LIST** fields, such as **[SOURce:]LIST:OUTPut**. If no **LIST:POWer** values are specified, **LIST:POWer** has no effect for **LIST** mode execution. If **LIST:POWer** has exactly one value, that value is used for the whole list. **<number>** may be left blank to indicate "no change" or may be the symbols **MAX** or **MIN.**

Blank powers allow **LIST:SEQuence** to refer to the same point multiple times with **POWer** set by other points. This handling of blank or "no change" values means **"LIST:POW 0,,,,", a POWer** of 0dBm followed by four "no change" values, rather than **"LIST:POW 0"**, which **SCPI-99** specifies as acting like a list having the same number of points as the othe**r LISTs** with all the values the same.

The total time for a list mode sweep is the sum of the dwell times plus the 250usec **POWer** change time per point. Dwell times may be set by **[SOURce:]LIST:DWELI** or **[SOURce:]SWEep:DWELI.**

LIST:POW? returns an empty string ("\n") if the list is empty.

LISTs are not affected by ***RST. LISTs** are empty after power cycling.

Requires SW Ver >= 2.5.

3.3.52.1 Example: LIST:POWer

```
list:pow 10, 5, 3, 0, -3, -5, -10
list:pow?
10.0,5.0,3.0,0.0,-3.0,-5.0,-10.0
list:pow:poin?
7
list:pow:poin? min
0
list:pow:poin? max
2048
pow:mode list
init
```

3.3.53 [SOURce:]LIST:POWer:POINts?

LIST:POWer:POINts? reads back the number of powers in **[SOURce:]LIST:POWer**. Specifying the arguments **MIN** or **MAX** reads back the minimum or maximum number of points. **[SOURce:] LIST:SEQuence** affects the maximum number of points usable at one time.

This is a query only; there is no command form.

Requires SW Ver >= 2.5.

3.3.54 [SOURce:]LIST:SEQuence[?]

LIST:SEQuence [<point#>]{,<point#>} allows the points defined by **LIST:FREQ, LIST:POW, LIST:OUTP**, and **LIST:DWEL** to be executed in any order and any number of times. This can be used to construct more complicated patterns or to optimize program execution time by loading all of the points of interest at startup time and then just referring to the points by number since sequence lists are faster to load. It is also possible to specify just frequencies in some **LIST** points and just powers in other **LIST** points then use the **SEQuence** to select any frequency with any power by specifying the points in order.

<point#> values are in the range 1 to the number of **LIST** points. Values outside this range will cause errors at **INITiate** time.

The **SEQuence** points are stored in the same memory as the **LIST** points. Each of the 2048 memory locations can hold one **FREQ/POW/OUTP/DWEL LIST** point or six **SEQuence** points. If you can specify all of your states in 100 points, you can create a sequence list over 12,000 points long. **[SOURce:]LIST:SEQuence:POINts?** will tell you how many **SEQuence** points can be used based

on the number of **LIST** points. **[SOURce:]LIST:FREQ:POINts?**, for instance, always returns 2048, regardless of how many sequence points are in use. (It is assumed that if you update the **LIST** points then the **SEQuence** will need to be updated, so **LIST** points always have precedence in memory conflicts.)

LIST:SEQ is not affected by ***RST.**

The query form, **LIST:SEQ?**, returns a comma separated list of points. If there are zero points in the **SEQuence**, an empty string is returned.

Requires SW Ver >= 2.5.

3.3.54.1 Example: LIST:SEQuence

```
sour:list:freq 3000000000,4000000000,5000000000,,6000000000
sour:list:pow 0,1,2,,3
sour:list:outp 1,,,0,1
sour:list:dwel 0.003,0.005,0.007,0.009,0.011
sour:list:seq 4,5,3,1,2,4
freq:mode list
list:dir down
init

list:pow:poin?;:list:freq:poin?;:list:outp:poin?;:list:dwel:poin?
5;5;5;5
list:seq:poin?
6

list:seq 12345
syst:err:all?
928,"Sequence list index out of range; 12345 outside of range [1,2048]"
list:seq 1234
init
syst:err:all?
928,"SOURce:LIST:SEQuence contains 1 or more invalid indexes; 1234 outside of range [1,5]".
```

3.3.55 [SOURce:]LIST:SEQuence:POINts[?]

LIST:SEQuence:POINts? reads back the number of points in **[SOURce:]LIST:SEQuence**. Specifying the arguments **MIN** or **MAX** reads back the minimum or maximum number of points based on the current number of **LIST** points defined.

This is a query only; there is no command form.

Requires SW Ver >= 2.5.

3.3.56 [SOURce:]POWer:CENTer[?]

POWer:CENTer <pow> is used with **POWer:SPAN**, **POWer:STARt**, and **POWer:STOP** to specify the range of power to sweep over.

<pow> is a power in dBm and can be written as a number with or without the dBm suffix. The text strings **MINimum** and **MAXimum** are also accepted.

POWer:CENTer? returns the center power in dBm.

Specifying **CENTer** with **POWer:SPAN**, **POWer:STARt**, or **POWer:STOP** next to it on the same line allows the sweep power values to be set without going through intermediate states which could result in extraneous errors or confusing behavior.

See Also: **[SOURce:]POWer:SPAN, [SOURce:]POWer:START, [SOURce:]POWer:STOP, [SOURce:]FREQuency:CENTer, [SOURce:]POWer:STEP**

3.3.56.1 Example: Setting Up a Power Sweep with Center and Span

```
pow:cent 0;span 20;step 0.1
pow:star?;stop?
-10.0;10.0

trig:sour imm
pow:mode swe
swe:dir down; dwel 3ms
init
```

3.3.57 [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude][?] - Output Power

POWer <value> sets the RF Output power in dBm into 50 Ohms.

<value> can be written as a number with or without a **dBm** suffix and also accepts the text strings **MINimum, MAXimum, UP,** and **DOWN**. The effect of **UP** and **DOWN** is controlled by **POWer:STEP**.

The query form returns a floating point number in dBm. It also accepts the strings MINimum and MAXimum.

3.3.58 [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]:RESolution?

POWer:RESolution? reads the smallest output power change supported by the hardware.

There is no command to change power resolution.

3.3.59 [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCRement][?]

POWer:STEP <value> sets the amount the output power will change when **POWer UP** or **DOWN** is used, and for power sweeps.

<value> can be a number or the text strings **MINimum** and **MAXimum**.

The query form returns a floating point number in dBm. It also accepts the strings **MINimum** and **MAXimum**.

See Also: **[SOURce:]POWer**

3.3.60 [SOURce:]POWer:MODE[?] {FIX|SWEep|LIST}

POWer:MODE {FIXed|SWEep|LIST} controls whether the HMC-T2240 is configured for a stepped sweep, an arbitrary list of frequencies, or for single frequency **(FIXed)** operation.

The query form returns **FIX, SWE,** or **LIST**.

The ***RST** state is **FIX**.

Enabling either **POW:MODE LIST** or **[SOURce:]FREQuency:MODE LIST** is sufficient to enable **LIST** mode which can affect both **FREQuency** and **POWer** (as well as **OUTPut[:STATe]** and **[SOURce:]SWEep:DWELI**.) **LIST** mode requires SW Ver >= 2.5.

See Also: **[SOURce:]SWEep:LIST:POWer**

3.3.61 [SOURce:]POWer:SPAN[?]

POWer:SPAN <pow> is used with **POWer:CENTer**, **POWer:STARt**, and **POWer:STOP** to specify the range of power to sweep over.

<pow> is a power in dBm and can be written as a number with or without the dBm suffix.

Specifying **SPAN** with **POWer:CENTer**, **POWer:STARt**, or **POWer:STOP** next to it on the same line allows the sweep power values to be set without going through intermediate states which could result in extraneous errors or confusing behavior.

See Also: **[SOURce:]POWer:CENTer, [SOURce:]POWer:START, [SOURce:]POWer:STOP**

3.3.62 [SOURce:]POWer:STARt[?]

POWer:STARt <pow> is used with **POWer:CENT**, **POWer:SPAN**, and **POWer:STOP** to specify the range of power to sweep over.

<pow> is a power in dBm and can be written as a number with or without the dBm suffix. The text strings **MINimum** and **MAXimum** are also accepted.

NOTE: POWer:STARt must be less than **POWer:STOP**.

POWer:STARt? returns the starting power in dBm.

Specifying **STARt** with **POWer:SPAN**, **POWer:CENTer**, or **POWer:STOP** next to it on the same line allows the sweep power values to be set without going through intermediate states which could result in extraneous errors or confusing behavior.

See Also: **[SOURce:]POWer:SPAN, [SOURce:]POWer:CENTer, [SOURce:]POWer:STOP, [SOURce:]FREQuency:CENTer, [SOURce:]SWEep:DIRection**

3.3.62.1 Example: Setting Up a Power Sweep with Start and Stop

```
pow:star -10;stop +20;step 0.1;mode swe
init
pow:cent?;span?
5.0;30.0
```

3.3.63 [SOURce:]POWer:STOP[?]

POWer:STOP <pow> is used with **POWer:CENTer**, **POWer:SPAN**, or **POWer:STARt** to specify the range of power to sweep over.

<pow> is a power in dBm and can be written as a number with or without the **dBm** suffix. The text strings **MINimum** and **MAXimum** are also accepted.

NOTE: POWer:STARt must be less than **POWer:STOP**.

POWer:STARt? returns the starting power in dBm.

Specifying **STOP** with **POWer:SPAN**, **POWer:STARt**, or **POWer:CENTer** next to it on the same line allows the sweep power values to be set without going through intermediate states which could result in extraneous errors or confusing behavior.

See Also: **[SOURce:]POWer:SPAN, [SOURce:]POWer:START, [SOURce:]POWer:CENTer, [SOURce:]FREQuency:CENTer, [SOURce:]SWEep:DIRection**

3.3.64 [SOURce:]POWer:TCOMpensation[?] – Temperature Compensation

Temperature Compensation adjusts the output power based on the synthesizer's internal temperature for steady state operation over the range of 0 to 35 ºC (See Datasheet). Temperature compensation is necessary for the output power to remain relatively stable across temperature at higher frequencies for the HMC-T2240 or HMC-T2270; it does not operate below 20 GHz.

Temperature Compensation should be ON when the unit is shipped to you. It can be controlled from the front panel or with the syntax **POWer:TCOMpensation {ON|OFF}**. It is not affected by either *RST or power cycling. We recommend that you leave it ON as cold temperatures can result in significantly more power than programmed.

POWer:TCOMpensation? returns 0 for OFF and 1 for ON.

Temperature Compensation requires SW ver 2.4 or later.

3.3.64.1 Example: Temperature Compensation On/Off

power:tcompensation off
power:tcompensation?

0
pow:tcom on
pow:tcom?

1

3.3.65 [SOURce:]SWEep:COUNt[?]

SWEep:COUNt <number> determines the number of sweeps that happen for each **TRIGger/*TRG**, or **INIT** if **TRIGger:SOURce IMMediate**.

3.3.66 [SOURce:]SWEep:DIRection[?]

SWEep:DIRection {UP|DOWN} determines whether sweeps have increasing or decreasing frequency. **FREQ:STARt** must always be less than **FREQ:STOP**; **POW:STARt** must be less than **POW:STOP**. The query form returns the string **UP** or **DOWN**.

3.3.66.1 Example: Sweep Direction

sweep:dir?
UP
sweep:dir down
sweep:dir?
DOWN

3.3.67 [SOURce:]SWEep:DWELl[?]

SWEep:DWELl <number> controls the time each frequency or power is held stable during a sweep.

<number> is a time in seconds with 1usec resolution or the string **MINimum** or **MAXimum**.

While the unit is sweeping, a timer determines when each frequency or power change starts. It takes a certain amount of time to change from the current frequency to the new frequency. This time is called the Frequency Change time. When the Frequency Change time is over, the Dwell time begins. At the end of the Dwell time, the timer signals the next frequency change or the completion of the sweep. The timer period is therefore the sum of the Frequency Change time and Dwell time. The frequency change time, in addition to any TRIGger:DELay, is 250 usec and the minimum dwell time varies by model. The time for the first point after trigger is:

Model	Min. Dwell	Min. Period	Min. Delay from Trigger
HMC-T2220	25 usec	275 usec	250 usec
HMC-T2240	300usec	550usec	500usec
HMC-T2270	25 usec	275 usec	250 usec
T2100 Compatibility Mode	100 usec	350 usec	250 usec

Sending commands to the unit during a sweep can delay when the frequency changes actually happen, cutting into the Dwell time. Delays are not cumulative.

3.3.67.1 Example: Dwell Time

swe:dwel 0.1s
swe:dwel?
0.100000
swe:dwel? min
0.000500

3.3.68 [SOURce:]SWEep:MODE[?]

SWEep:MODE AUTO|MANual determines whether sweeps, whether normal or list, advance based on th**e DWELl** time **(AUTO)** or wait for another trigger event **(MANual.)**

The ***RST** state is **AUTO.**

The query form **[SOURce:]SWEep:MODE?** returns **AUTO** or **MAN.**

3.3.69 *SRE[?] – Service Request Enable

***SRE <mask>** enables bits in the Status Byte (STB) to generate a Service Request.

<mask> is a number in the range [0,255].

This is useful with GPIB which has a dedicated service request line. For other interfaces, such as USB, it is possible to poll using ***STB?** or use the **STATus:SRQChar** command to indicate you want a specific character to be sent to indicate a service request.

***SRE** is usually used in conjunction with ***ESE** to report errors and **STATus:QUEStionable:ENABle** to report when the **POWer** setting goes outside the calibrated range.

The query form is affected by **FORMat:SREGister**.

3.3.70 STATus:OPERation:CONDition?

STATus:OPERation:CONDition? provides a mechanism to look at the current instrument status. Unlike the EVENt register, the CONDition register is not sticky, so it can be polled without the need to do an initial read or call ***CLS** to clear any stale values.

The query form is affected by **FORMat:SREGister**.

There is no command to set the CONDition register.

3.3.71 STATus:OPERation Register Bit Definitions

The OPERation status register is 16 bits wide, but the HMC-T2240 only sets the following bits at this time: (Embedded software rev 1.4)

3 SWEeping – Sweep is initiated and triggered

5 Waiting for TRIGger – Sweep is initiated but not triggered

8 Battery Present (HMC-T2220B)

9 Battery Charging (HMC-T2220B)

These bits will not be set in the CONDition register at the same time as they are mutually exclusive. If **TRIGger:SOURce IMMediate**, the Waiting for TRIGger bit will not be set before the SWEeping bit as the instrument does not actually wait.

Other bits may be enabled in the future.

3.3.71.1 Example: Operation Condition Status

```
freq:mode swe
trig:sour bus
init
stat:oper:cond?
32
*trg
stat:oper:cond?
8
abor
stat:oper:cond?
0
```

3.3.72 STATus:OPERation:ENABle[?]

STATus:OPERation:ENABle <mask> allows the state of the OPERation:EVENt status register to be summarized in bit 7 of the standard status byte (STB.)

<mask> is in the range [0, 65535].

The ENABle registers can be cleared by writing to them or with **STATus:PRESet**.

The query form is affected by **FORMat:SREGister**.

See Also: ***ESE, *STB?, STATus:OPERation Register Bit Definitions, STATus:OPERation?, STATus:PRESet**

3.3.73 STATus:OPERation[:EVENt]?

STATus:OPERation? reads the sticky EVENt register which holds the value of the STATus:OPERation:CONDition register if it goes high even momentarily. This allows fast events to be captured reliably.

The EVENt register is cleared by reading it and by ***CLS**.

The HMC-T2240 does not support transition filter registers at this time. (Embedded software rev 1.8)

The query form is affected by **FORMat:SREGister**.

See Also: ***ESR?, *STB?, *CLS, STATus:OPERation Register Bit Definitions**

3.3.74 STATus:PRESet

STATus:PRESet clears the following status registers:

- **STATus:OPERation:ENABle**
- **STATus:QUEStionable:ENABle**

It does not affect *ESE or *SRE, or the EVENt registers.

See Also: ***CLS**

3.3.75 STATus:QUEStionable:CONDition?

STATus:QUEStionable:CONDition? provides a mechanism to look at the current instrument status. Unlike the EVENt register, the CONDition register is not sticky, so it can be polled without the need to do an initial read or call ***CLS** to clear any stale values.

There is no command to set the CONDition register.

3.3.76 STATus:QUEStionable Register Bit Definitions

The QUEStionable status register is 16 bits wide, but the HMC-T2240 only sets the following bits at this time:

3	POWer – The current power is outside the calibrated range.
4	TEMPerature - unit is too hot/cold.
5	FREQuency - Frequency not locked. **SOUR:ROSC:SOUR EXT** and external 10 MHz reference not present?
9	FAN - One or both fans have failed.
10	Battery Low (HMC-T2220B)

Other bits may be enabled in the future.

3.3.77 STATus:QUEStionable:ENABle[?]

STATus:QUEStionable:ENABle <mask> allows the state of the QUEStionable:EVENt status register to be summarized in bit 7 of the standard status byte (STB.)

<mask> is in the range [0, 65535].

The ENABle registers can be cleared by writing to them or with **STATus:PRESet**.

The query form is affected by **FORMat:SREGister**.

See Also: ***ESE, *STB?, STATus:QUEStionable Register Bit Definitions, STATus:QUEStionable?, STATus:PRESet**

3.3.78 STATus:QUEStionable[:EVENt]?

STATus:QUEStionable? reads the sticky EVENt register which holds the value of the STATus:QUEStionable:CONDition register if it goes high even momentarily. This allows fast events to be captured reliably. It also means you do not have to poll constantly to detect if the output power was programmed outside the calibrated range.

The EVENt register is cleared by reading it and by ***CLS**.

The query form is affected by **FORMat:SREGister**.

The HMC-T2240 does not support transition filter registers at this time. (Embedded software rev 1.8)

See Also: ***ESR?, *STB?, *CLS, STATus:QUEStionable Register Bit Definitions**

3.3.79 STATus:SRQChar[?] – Service Request over USB

STATus:SRQChar <code> is a Hittite extension to allow service requests to be generated for interfaces other than GPIB. This command is inherently not portable to other instrumentation and should not be used if compatibility is a requirement.

<code> is a value in the range [0 to 256] and should be chosen to be an obvious error signal. 256 turns off the Service Request character.

The query form is affected by **FORMat:SREGister**.

3.3.79.1 Example: STATus:SRQChar – Service Request Character

```
stat:srqc 1
stat:srqc?
1
*SRE #hff
*ESE #hff
sfs
□
```

The above character appeared as a smiley face in HyperTerminal. "sfs" is not a legal command.

```
*stb?
100
form:sreg hex
*stb?
#H64
```

Bit 6 is the Service Request, Bit 5 is the ESR summary, and Bit 2 is the Error Queue summary.

```
*ESR?
#H20
```

Command Error

syst:err?;err?
-13,"Undefined header";0,"No error"

Turn off Service Request character

stat:srqc 256

3.3.80 *STB? – Status Byte

***STB?** reads the summary Status Byte which reflects the state of a number of other status registers in the system to allow a quick look at whether the instrument needs attention. It is affected by **FORMat:SREGister**.

There is no command to write to the STB. It may be cleared by clearing the registers that feed into it.

See Also: ***ESR?**, **STATus:OPERation?**, **STATus:QUEStionable?**, **SYSTem:ERRor?**, ***CLS**, **STATus:SRQChar**

3.3.81 Status Byte (STB) Bit Definitions

The HMC-T2240 uses the following bits in the Status Byte (STB.)

0, 1 Reserved
2 Error Queue is not empty
3 SCPI QUEStionable Status summary
4 Message Available
5 IEEE 488 (GPIB) Standard Event Status Register (ESR) summary
6 Service Request
7 SCPI OPERation Status summary

The Message Available bit is not useful on interfaces other than GPIB because other interfaces send their responses immediately instead of waiting for them to be read.

Reserved bits may be used in the future.

3.3.82 SWEep – See [SOURce:]SWEep

The SWEep subsystem is placed under the SOURce subsystem by SCPI.

3.3.83 SYSTem:COMMunicate:ETHernet:ADDRess[?] – IP Address

SYST:COMM:ETH:ADDRess sets the IP address.

Each IP address on a subnet must be unique. See your network administrator for policies on IP address assignment.

You do not need to set the ADDRess if you are using DHCP.

Only IPv4 addresses (four numbers separated by dots like 192.168.1.27) are supported.

You must power cycle the HMC-T2240 after changing this or any or any other network parameter.

If you are setting ADDRess, you will also need to set the NETMask and GATeway.

SYST:COMM:ETH:ADDRess? reports the current IP adress regardless of whether DHCP is on or off.

IP addresses are treated as strings. You do not have to send quotes, and the results of queries will not be quoted.

See Also: **SYSTem:COMMunicate:ETHernet:DHCP, SYSTem:COMMunicate:ETHernet:NETMask, SYSTem:COMMunicate:ETHernet:GATeway, SYSTem:COMMunicate:ETHernet:PORT**

3.3.83.1 Example: Network configuration with DHCP OFF

Show error messages immediately in case of typo (not for GPIB)
syst:err:beh imm

syst:comm:eth:addr 192.168.1.27
syst:comm:eth:netm 255.255.255.0
syst:comm:eth:gat 192.168.1.1
syst:comm:eth:port 65432
syst:comm:eth:dhcp OFF

Reading values back before unit power cycled shows some old values
syst:comm:eth:dhcp?;addr?;netm?;gat?;port?
0;196.168.0.5;255.255.0.0;192.168.0.3;65432

Power cycle here

Values programmed above are now in effect
syst:comm:eth:dhcp?;addr?;netm?;gat?;port?
0;192.168.1.27;255.255.255.0;192.168.1.1;65432

3.3.84 SYSTem:COMMunicate:ETHernet:DHCP[?] – Automatic Configuration

SYST:COMM:ETH:DHCP ON|OFF enables or disables Dynamic Host Configuration Protocol. DHCP ON allows the HMC-T2240 to get its IP Address, subnet mask, and default gateway IP address automatically when it connects to a network.

See your network administrator to find out if you can use DHCP or must use static IP addresses.

You must still set the socket port number (**SYSTem:COMMunicate:ETHernet:PORT**) even if you are using DHCP, if you are using sockets.

Most routers support DHCP, although they allow you to turn it off. Most PCs are not configured to be DHCP servers, so you may not be able to use DHCP if you are connecting directly from a PC to the HMC-T2240 with a crossover cable.

The query form returns 1 for ON and 0 for OFF.

See Also: **SYSTem:COMMunicate:ETHernet:PORT, SYSTem:COMMunicate:ETHernet:ADDress, SYSTem:COMMunicate:ETHernet:NETMask, SYSTem:COMMunicate:ETHernet:GATeway**

3.3.84.1 Example: Network Configuration with DHCP ON

syst:comm:eth:dhcp on
syst:comm:eth:port 54321
syst:comm:eth:dhcp?;port?
1;54321

Settings do not take effect until unit power cycled

After power cycling
syst:comm:eth:dhcp?;port?;addr?;netm?;gat?
1;54321;10.0.0.2;255.255.255.0;10.0.0.1

3.3.85 SYSTem:COMMunicate:ETHernet:GATeway[?]

SYST:COMM:ETH:GATeway specifies the default gateway IP Address.

You do not have to set the GATeway address if you are using DHCP.

The HMC-T2240 does not have any features requiring outgoing network access. (No automatic firmware updates or product registration, for instance.) However, the gateway must be set to a reasonable value for the unit to work correctly. Setting the gateway to the unit's own address will cause it to hang. You must disconnect the network cable, power cycle the unit, and fix the gateway address through USB or GPIB to recover.

The query form returns the IP address as an unquoted string.

See Also: **SYSTem:COMMunicate:ETHernet:DHCP, SYSTem:COMMunicate:ETHernet:ADDRess**

3.3.86 SYSTem:COMMunicate:ETHernet:NETMask[?]

SYST:COMM:ETH:NETMask specifies the subnet mask.

You do not have to set the NETMask if you are using DHCP.

The HMC-T2240 NETMask setting should match the net mask of the router or PC it is connected to.

See your Network Administrator if you do not know what setting to use.

The query form returns an unquoted string, typically something like:

255.255.255.0

See Also: **SYSTem:COMMunicate:ETHernet:DHCP, SYSTem:COMMunicate:ETHernet:ADDRess**

3.3.87 SYSTem:COMMunicate:ETHernet:PORT[?] – Socket Number

SYST:COMM:ETH:PORT <number> specifies the port number to use for socket connections.

The port number is not configured by DHCP, so you must set it explicitly if you wish to use sockets.

<number> is in the range 0 to 65535.

If you are already using a particular socket number for your applications, set the HMC-T2240 to use that number too.

If you do not already have a standard port number, pick one from the range 49152-65535. Lower number ports may already be in use (23 = Telnet) or be used in the future (80 = HTTP, 111 = VXI-11, 5044 = LXI, for example) or may be used by common network protocols resulting in interference or future incompatibility.

The query form is *not* affected by **FORMat:SREG**.

See Also: **SYSTem:COMMunicate:ETHernet:DHCP, SYSTem:COMMunicate:ETHernet:ADDRess**

3.3.88 SYSTem:COMMunicate:GTLocal

GTLocal enables control of the HMC-T2240 from the front panel knob and button even if **SYSTem-KLOCk** is set. The user does not have to press and hold the front panel knob to take the unit out of remote.

This is a Hittite extension and is not portable. Do not use if compatibility with other instruments is required.

GTLocal is an event; it does not have a query form.

3.3.88.1 Example: Go To Local

SYST:COMM:GTL

3.3.89 SYSTem:COMMunicate:GTRemote

GTRemote puts the unit under remote control. If **KLOCk** is enabled, the user cannot get control back by pressing and holding the front panel knob. If the user does press and hold the front panel knob, the ESR "User Request" bit is set, and that bit can be polled or connected to the service request mechanism so the controlling computer can send **SYST:COMM:GTLocal** to re-enable front panel control at a point in the program where it is safe to do so.

Since commands which affect the RF Output will put the unit into remote anyway, this command is not especially useful. (It was created to support the GUI Remote check box.)

This is a Hittite extension and is not portable. Do not use if compatibility with other instruments is required.

GTRemote is an event; it does not have a query form.

3.3.89.1 Example: Go To Remote

SYST:COMM:GTR

3.3.90 SYSTem:ERRor:ALL? – Read All Error Messages

SYSTem:ERRor:ALL? reads back all the error messages from the error queue.

If there are no error messages in the queue, the result is **0, "No error"**.

NOTE: This query can respond with > 256 characters because each error message may be up to 256 characters and the error queue can hold up to 10 error messages plus the **-350, "Queue overflow"** message for a total of up to 2583 characters.

See Also: **SYSTem:ERRor[:NEXT]?, *CLS**

3.3.90.1 Example: Read All Error Messages from Error Queue

typo
oops
freq 0dBm
pow 20GHz
****idn?**
syst:err:all?
-113,"Undefined header; typo",-113,"Undefined header; oops",-131,"Invalid suffix; freq 0dBm",-131,"Invalid suffix; pow 20GHz",-113,"Undefined header; **idn?"

3.3.91 SYSTem:ERRor:BEHavior

SYSTem:ERRor:BEHavior {QUEue|IMMediate} determines whether error messages are queued per the SCPI standard or issued immediately. The default is **QUEued**.

This is useful when typing commands in by hand as you don't have to guess whether you got an error or not.

This is a Hittite extension and should not be used where portability is required.

The query form returns **QUE** or **IMM**.

This function is not affected by ***RST**.

3.3.91.1 Example: Issue Error Messages Immediately

SYST:ERR:BEH IMM
freq 3
200,"FREQuency out of range; 3 outside of range [10000000,40000000000]"
pow 20GHz
-131,"Invalid suffix; pow 20GHz"
syst:err:beh que
freq 3
pow 20GHz
syst:err?;err?;err?
200,"FREQuency out of range; 3 outside of range [10000000,40000000000]";
-131,"Invalid suffix; pow 20GHz";0,"No error"

3.3.92 SYSTem:ERRor[:NEXT]?

SYSTem:ERRor? reads back error messages from the error queue.

If the error queue is empty, the result is **0,"No error"**.

There is no command to write errors into the queue.

See Also: **SYSTem:ERRor:BEHavior, *CLS, SYSTem:ERRor:ALL?**

3.3.92.1 Example Read Error Message from Error Queue

typo
SYST:ERR?
-113,"Undefined header"
freq 27THz

pow -173dBm
SYS:ERR?;ERR?;ERR?
200,"FREQuency out of range; 27000000000000 outside of range [10000000,40000000000]"
;300,"Power out of range; -173.0dBm outside of range [-60.0,35.0]dBm";0,"No error"

3.3.93 SYSTem:KLOCk[?] – Front Panel Control Lock

SYSTem:KLOCk {ON|OFF} locks out the front panel controls so pressing and holding the front panel knob does not return to load control.

The query **SYSTem:KLOCk?** returns 1 for **ON** and 0 for **OFF**.

See Also: **SYSTem:COMMunicate:GTLocal, SYSTem:COMMunicate:GTRemote, LLO**

3.3.93.1 Example: Local Lockout with KLOCk

syst:kloc?
0
syst:klock on
syst:kloc?
1

3.3.94 SYSTem:PRESet

SYSTem:PRESet is the same as ***RST**.

3.3.95 *TRG – Trigger

***TRG** triggers a sweep. It is an error to send ***TRG** when the hardware is not in the **INITiated** state, waiting for a bus trigger. This includes sending a trigger during a sweep.

***TRG** is an event; there is no query form.

***TRG** is similar to **TRIGger,** except **TRIGger** can be used with **TRIGger:SOURCE EXTernal** also.

3.3.95.1 Example: *TRG – Trigger

***rst**
syst:err:beh imm
***trg**
-211,"Trigger ignored;TRIG:SOUR BUS not selected"
trig:sour bus
***trg**
-211,"Trigger ignored; not INITiated"
init
-213,"Init ignored; Wrong MODE-of-operation"
freq:mode swe
init
stat:oper:cond?
32
***trg**
stat:oper:cond?
8

abor
stat:oper:cond?
0
init
stat:oper:cond?
32
trig
stat:oper:cond?
8

3.3.96 TRIGger:DELay[?]

TRIGger:DELay <number> causes the start of the sweep or the advance to the next state **(SWEep:MODE MANual)** to be delayed by <number> seconds. This can be used to deskew the HMC-T2200 relative to other instruments.

<number> can be 0 or >= 100us.

The ***RST** value is 0.

The query form returns a number in seconds.

Requires SW Ver >= 2.5.

3.3.96.1 Example:TRIGger:DELay

0.000000
trig:del? max
900.000000
trig:del? min
0.000000
trig:del 250us
syst:err:all?
0,"No error"
trig:del 50us
syst:err:all?
900,"Trigger delays > 0us and < 100us not supported"

3.3.97 TRIGger[:IMMediate]

TRIGger is the same as ***TRG**, except it can cause an EXTernal trigger as welll as a BUS trigger.

3.3.98 TRIGger[:IMMediate]:SOURce[?]

TRIGger:SOURce {IMMediate|BUS|EXTernal} selects whether a sweep that is INITiated begins immediately or waits for a trigger event to start sweeping.

The query form returns **IMM** or **BUS** or **EXT**.

IMMediate	**INITiate** starts the sweep without waiting for anything else.
BUS	***TRG**, **TRIGger** or GPIB **GET** will start the sweep
EXTernal	A rising edge (TTL) on the trigger IN BNC starts the sweep.

See Also: ***TRG**, **TRIGger, TRIGger:SLOPe**

3.3.98.1 Example: Trigger Source

trig:sour bus
trigger:source?
BUS
trig:sour immediate
trig:sour?
IMM
trig:sour external
trig:sour?
EXT

3.3.99 TRIGger:SLOPe POSitive|NEGative|EITHer[?]

TRIGger:SLOPe <slope> determines whether the **EXTernal** (BNC) trigger input responds to rising edges (**POSitive,**) falling edges (**NEGative**,) or whichever comes first (**EITHer**,).

<slope> can be:

- **POSitive** - rising edge (Reset value)
- **NEGative** - falling edge
- **EITHer** - whichever edge comes first

The query form, **TRIGger:SLOPe?**, returns **POS, NEG, or EITH**.

See Also: **TRIGger:SOURce**

3.3.99.1 Example: Trigger a sweep on the falling edge

The following sets up a frequency sweep triggered by the falling edge on the EXTernal trigger input.

FREQ:STARt 3e9;STOP 5e9;STEP 5e6;MODE SWEep
TRIG:SOURce EXTernal;SLOPe NEGative
INITiate

Unit will wait for a falling edge on Trigger In.

Use **TRIGger** to force the sweep to start, or **ABORt** to exit the **INITiate**d state, if no external trigger occurs.

3.3.100 *WAI – Wait for Operation (Sweep) to Complete

***WAI** waits for an operation to complete. Since most operations do not return control to the user until they have completed, ***WAI** usually does nothing. If a sweep is running, however, ***WAI** prevents any further commands from executing until the sweep is completed. This is very similar to the behavior of ***OPC?** except that ***OPC?** returns a "1" when the sweep completes and ***WAI** does not. ***WAI** can be used in conjunction with ***OPC?**.

If the sweep is never going to terminate, whether because it isn't going to be triggered or because **SWEep:CONTinuous ON**, **SDC** can cancel the effect of ***WAI** to allow new commands to be received.

***WAI** is an event; there is no query form.

See Also: ***OPC**, ***OPC?**, **SDC**

3.3.100.1 Example: *WAI to Wait for Sweep to Complete

Repeat a frequency sweep at different power levels, starting from +10 dBm.

```
pow 10dBm;pow:step 5
freq:star 3e9;stop 5e9;step 100e6;mode swe
init;*wai;pow down
```

*wai prevents the power from changing until each frequency sweep is completed.

```
init;*wai;pow down
init;*wai;pow down
```


2 Elizabeth Drive Chelmsford, MA 01824
978-250-3343 tel • 978-250-3373 fax • TE@hittite.com