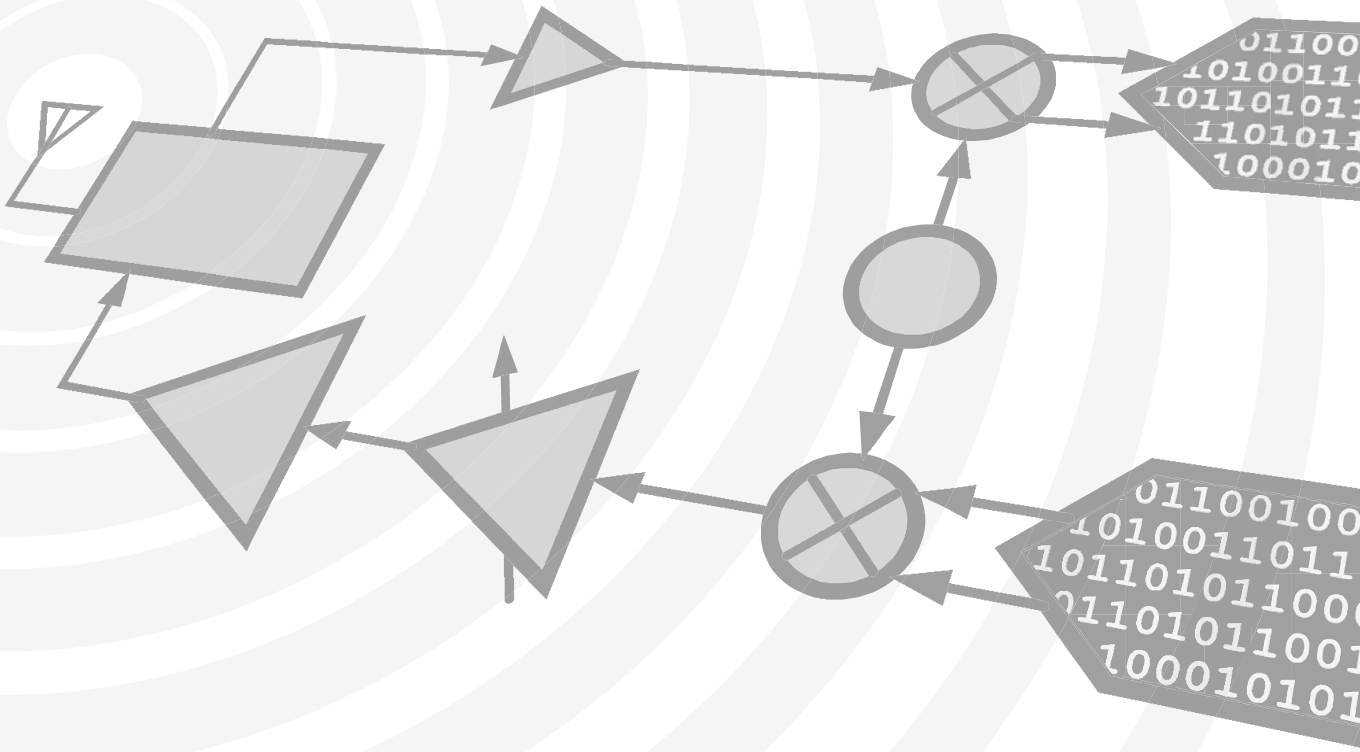
ANALOG DEVICES

Hittite MICROWAVE PRODUCTS FROM ANALOG DEVICES

Analog Devices Welcomes Hittite Microwave Corporation

NO CONTENT ON THE ATTACHED DOCUMENT HAS CHANGED



www.analog.com

www.hittite.com

THIS PAGE INTENTIONALLY LEFT BLANK

Synthesized Signal Generator, 10 MHz to 20 GHz

HMC-T2100

125791 Rev C - 01.0110

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

Programmer's Manual

Installation, Operation & Maintenance Guide for HMC-T2100 & HMC-T2100B

HMC-T2100

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

20 Alpha Road Chelmsford, MA 01824
Phone: 978-250-3343 • Fax: 978-250-3373 • sales@hittite.com

Table of Contents

Table of Contents *(Continued)*

Table of Contents *(Continued)*

1.0 Introduction

The HMC-T2100 and HMC-T2100B (Battery version) 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 identical for both the HMC-T2100 and HMC-T2100B. For simplicity only the HMC-T2100 will be referenced throughout the document.

2.0 Quick Start

2.1 USB Quick Start

The HMC-T2100 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-T2100.

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 part of the standard Windows distribution 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-T2100 doesn't need them, but they make the output more readable.)

2.1.2 HMCSynthDisplay

The HMC-T2100 install CD contains a graphical interface called HMCSynthDisplay which allows you to program Frequency, Power, RF Output, and sweep related parameters.

The display has a pulldown list of synthesizers which can be selected. Note that if another application has a synthesizer open it may not appear on the list (HMC-T2000) or you will not be able to open it (HMC-T2100).

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 pulldown list of signal generators when units have been added or removed.

- **Close** a HMC-T2100 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.

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

HMCSynthDisplay only works over USB.

2.1.3 HMCSynth Programming Interface

The HMC-T2100 install CD also contains HMCSynth.DLL, a "C" language and "COM" compatible DLL which provides backward compatibility for the HMC-T2000.

The .H, .LIB, and .DLL files are installed in the C:\Program Files\Hittite\HMCSynth directory tree by default. HMCSythnC.h is fro the "C" interface. This directory must be in the path for HMCSynth.DLL to be found by your application at runtime.

This interface only works over USB.

2.2 GPIB

The HMC-T2100 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-T2100 GPIB Address can be set[1] 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.

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-T2100 supports the following GPIB functions:

Function	Symbol
Source Handshake	SH1
Acceptor Handshake	AH1
Talker, no "talk only" mode	T6
Talker, Extended, no "talk only" mode	TE6
Listener, no "listen only" mode	L4
Listener, Extended, no "listen only" mode	LE4
Service Request	SR1

1 HW Ver 5.1 and later, SW Ver 1.8 and later. Prior versions do not support GPIB.

Function	Symbol
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-T2100's hardware input buffer holds 16 bytes. The maximum input string length is 256 bytes. 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.

The HMC-T2100 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-T2100 output lines are terminated with both EOI and newline.

Serial Polls and Parallel Polls can be used at "any" time and will not introduce jitter or otherwise interfere with the operation of the unit. Normal commands (*STB?, for instance,) can cause the response to a trigger or advancing to the next frequency or power in a sweep to be delayed.

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-T2100 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-T2100 after changing the GPIB Address. However, changing it during a GPIB operation may result in undesirable behavior.

The HMC-T2100 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-T2100 (SW Ver 1.8 and later) 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 either of these, the address of the HMC-T2100 must be configured.

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-T2100'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-T2100's Ethernet address is with DHCP. If your network is configured for DHCP (Dynamic Host Configuration Protocol), you should[2] be able to connect to the HMC-T2100 to the network and access it through its host name like any other computer.

The HMC-T2100's host name (machine name) is:

HMCT2100-<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:

- You are connecting a PC directly with a cable, and most PCs are not set up as DHCP servers. (The HMC-T2100 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.)

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

SYSTem:COMMunicate:ETHernet:DHCP ON

You must power cycle the HMC-T2100 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-T2100 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-T2100 before these settings will take effect. You may also want to set the socket port number (discussed below) before power cycling.

2.3.2 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 **hmct2100-000000**.

```
C:\>telnet hmct2100-000000
*=========================================================*
*          Hittite T2100 Embedded Telnet Server                                  *
*=========================================================*
*idn?
Hittite,HMC-T2100,000000,1.8 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-T2100 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-T2100 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-T2100 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::HMCT2100-%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-T2100 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-T2100 (Ver 1.8) does NOT have password protection or other mechanisms to prevent unauthorized access.

The HMC-T2100 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-T2100 (SW Rev 1.8[3]) 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 ex-

3. The software version may be found by pressing and holding the front panel knob to get the GPIB address menu and then pressed again to get the serial number and revision information. It is labeled with SW VER:.

pected 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. (HMC-T2100 specific)
- 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.
- The only "overlapped" operation supported by the HMC-T2100 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-T2100) 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-T2100.

3.3.1 ABORt - Stop Sweeping

ABORt stops a running sweep.

If **SWEep:CONTinuous ON**, the sweep will restart (**INITiate**) immediately; turn **SWEep: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.

3.3.15 *RST - Reset

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

Command	Reset Value
FORMat:SREGister	ASCii
INITiate:CONTinuous	OFF
OUTPut[:STATe]	OFF

Command	Reset Value
*OPC	Clears flag to set Operation Complete when sweep done
[SOURce:]FREQuency:CENTer	10.005 GHz (mid band)
[SOURce:]FREQuency[:FIXedl:CW]	10.005 GHz (mid band)
[SOURce:]FREQuency:MODe	CW (sweep off)
[SOURce:]FREQuency:STEP	10 kHz (min)
[SOURce:]FREQuency:SPAN	19.99 GHz (max)
[SOURce:]FREQuency:STARt	10 MHz (min)
[SOURce:]FREQuency:STOP	20 GHz (max)
[SOURce:]POWer:CENTer	-4.5 dBm (mid range)
[SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]	-36 dBm (min)
[SOURce:]POWer:MODe	FIXed (sweep off)
[SOURce:]POWer:STEP	0.1 dB (min)
[SOURce:]POWer:SPAN	63 dB (max)
[SOURce:]POWer:STARt	-36 dBm (min)
[SOURce:]POWer:STOP	+27 dBm (max)
SWEep:COUNt	1
SWEep:DIRection	UP
SWEep:DWELl	3ms
TRIGger:SOURce	IMMediate

3.3.1.1 Example: Shut Off a Continuous Sweep

SWE: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.

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 *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?**

3.3.4.1 Example: Errors Propagate to STB

Cause any error to set bit 5 in STB (summary Status Byte read by ***STB?**):

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

3.3.5 *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**

3.3.5.1 The bits in the ESR:

(0 is least significant)

0. Operation Complete (See *OPC)
1. Unused (The HMC-T2100 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]?**.

3.3.5.2 Example: Detect an error

a_bad_command
***esr?**
32

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

SYSTem:ERRor?
-113,"Undefined header"

This is the SCPI defined error for an unrecognized command.

3.3.6 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.

3.3.6.1 Example: FORMat:SREGister

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

3.3.7 FREQuency – See [SOURce:]FREQuency

SCPI specifies the optional "SOURce" keyword before FREQuency.

3.3.8 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 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-T2100 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-T2100 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**

3.3.8.1 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 slow down the HMC-T2100 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**

3.3.8.1.1 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 T2100
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);
}
```

3.3.9 *IDN? – Identify

*IDN? produces an identification string including the manufacturer's name (Hittite), the model (HMC-T2100), the serial number, and revision information.

3.3.8.1 Example: *IDN?

***IDN?**

Hittite,HMC-T2100,000006,1.8 5.3

3.3.9 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.

3.3.9.1 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

3.3.10 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.

3.3.10.1 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**

3.3.11 *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**

3.3.12 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.

3.3.12.1 Example: RF On/Off State

To Set and Query the RF output state:

OUTPut ON
OUTP?
1
OUTP OFF
OUTP?
0

3.3.12.2 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

3.3.13 *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**

3.3.13.1 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.

3.3.14 *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-T2100 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**.

3.3.14.1 Example: Monitoring Sweep Status

swe:dwel 0.1
freq:star 1e9;stop 2e9;step 100e6;mode swe
init
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?**.

3.3.15 POWer – See [SOURce:]POWer

POWer is specified under the SCPI optional keyword SOURce.

3.3.16 *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**

3.3.17 *RST – Reset

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

3.3.18 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**.

3.3.18.1 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

3.3.19 [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 written with the **e6** suffix. It accepts MINimum and Maximum as arguments.

3.3.19.1 Example: Center and Span

Center and Span:

FREQ:CENT 3e9;SPAN 2e9
FREQ:STARt?;STOP?
2000.00e6;4000.00e6

3.3.19.2 Example: Center and Start

FREQ:CENT 3e9;STAR 1e9
FREQ:STARt?;STOP?
1000.00e6;5000.00e6

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

***rst**
FREQ:STARt?;STOP?
10.00e6;20000.00e6
freq:cent 3e9
FREQ:STARt?;STOP?
10.00e6;20000.00e6
syst:err?
250,"FREQ-Sweep Calculation ERROR; Calc By[FREQ-Center 3000.00e6 FREQ-Span 19990.00e6] outside of range [10.00e6,20000.00e6]"

freq:start 1e9
FREQ:STARt?;STOP?
1000.00e6;20000.00e6
freq:cent?
10500.00e6

3.3.19.3 Example: Center Frequency Min/Max

freq:cent? min
10.00e6
freq:cent? max
20000.00e6

3.3.20 [SOURce:]FREQuency[:FIXed|:CW][?] – Output 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 written with the **e6** suffix. It accepts MINimum and Maximum as arguments.

See Also: FREQuency:MODE, FREQuency:STEP

3.3.20.1 Example: Frequency Commands

freq 3.14159e9
frequency 2GHz
source:frequency:cw 20e9
sour:freq:fix 10MHz
freq:cw 20000000000.000
freq?
20000.00e6
freq:step 100MHz
freq down
freq?
19900.00e6
freq up
freq?
20000.00e6
freq minimum
freq?
10.00e6
freq maximum
freq?
20000.00e6
freq? min
10.00e6
freq? max
20000.00e6

3.3.21 [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 written with the **e6** suffix. It accepts MINimum and Maximum as arguments.

3.3.21.1 Example: Frequency Step

freq:step?
100.00e6
freq:step? min
0.01e6
freq:step? max
19990.00e6
freq:step 1234.56MHz

3.3.22 [SOURce:]FREQuency:MODE – Sweep Enable

FREQuency:MODE {CW|FIXed|SWEep} controls whether the HMC-T2100 is configured for a stepped sweep or for single frequency (**CW** or **FIXed**) operation.

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

3.3.22.1 Example: FREQuency:MODE[?]

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

Copied the following in separately to introduce a delay:

freq?
22.37e6

Copied the following in separately to introduce a delay:

freq:mode cw
freq?
44.76e6

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

3.3.23 [SOURce:]FREQuency:RESolution?

FREQuency:RESolution? returns the smallest frequency change supported by the hardware.

There is no command form to set the frequency resolution.

3.3.23.1 Example: Frequency Resolution Readback

freq:res?
0.01e6

3.3.24 [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 written with the **e6** suffix. It accepts MINimum and Maximum as arguments.

See Also: **FREQuency:CENTer**

3.3.25 [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 written with the **e6** suffix. It accepts MINimum and Maximum as arguments.

See Also: **FREQuency:CENTer**

3.3.26 [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**

3.3.27 [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**

3.3.27.1 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

3.3.28 [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.

3.3.29 [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.

3.3.30 [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**

3.3.31 [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, [SOURce:]FREQuency:CENTer**

3.3.32 [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**

3.3.32.1 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
```

3.3.33 [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**

3.3.34 [SOURce:]SWEep:COUNt[?]

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

3.3.35 [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**.

3.3.35.1 Example: Sweep Direction

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

3.3.36 [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 is 250 usec and the minimum dwell time is 3ms (100 usec in a future firmware version) for a minimum frequency allowing frequencies to change every 3.25 ms (350 usec future). The time for the first point after trigger TBD.

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.

3.3.36.1 Example: Dwell Time

swe:dwel 0.1s
swe:dwel?
0.100000
swe:dwel? min
0.003000

3.3.37 *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**.

3.3.38 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.

3.3.39 STATus:OPERation Register Bit Definitions

The OPERation status register is 16 bits wide, but the HMC-T2100 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

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.

3.3.39.1 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
```

3.3.40 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**

3.3.41 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-T2100 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**

3.3.45 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**

3.3.46 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.

3.3.47 STATus:QUEStionable Register Bit Definitions

The QUEStionable status register is 16 bits wide, but the HMC-T2100 only sets the following bits at this time: (Embedded software rev 1.8)

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?

Other bits may be enabled in the future.

9 FAN - One or both fans have failed.

3.3.48 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**

3.3.49 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-T2100 does not support transition filter registers at this time. (Embedded software rev 1.8)

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

3.3.50 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**.

3.3.50.1 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?
-113,"Undefined header";0,"No error"

Turn off Service Request character

stat:srqc 256

3.3.51 *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**

3.3.52 Status Byte (STB) Bit Definitions

The HMC-T2100 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.

3.3.53 SWEep – See [SOURce:]SWEep

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

3.3.54 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-T2100 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**

3.3.54.1 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
```

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

SYST:COMM:ETH:DHCP ON|OFF enables or disables Dynamic Host Configuration Protocol. DHCP ON allows the HMC-T2100 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-T2100 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**

3.3.55.1 *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

3.3.56 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-T2100 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**

3.3.57 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-T2100 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**

3.3.58 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-T2100 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**

3.3.59 SYSTem:COMMunicate:GTLocal

GTLocal enables control of the HMC-T2100 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.

3.3.59.1 Example: Go To Local

SYST:COMM:GTL

3.3.60 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 HMCSynthDisplay 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.

3.3.60.1 Example: Go To Remote

SYST:COMM:GTR

3.3.61 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**

3.3.61.1 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?"

3.3.62 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 not affected by ***RST**.

3.3.62.1 Example: Issue Error Messages Immediately

SYST:ERR:BEH IMM
freq 3
200,"FREQuency out of range; 0.00e6 outside of range [10.00e6,20000.00e6]"
pow 20GHz
-131,"Invalid suffix"
syst:err:beh que
freq 3
pow 20GHz
syst:err?;err?;err?
200,"FREQuency out of range; 0.00e6 outside of range [10.00e6,20000.00e6]";-131, "Invalid suffix";0,"No error"

3.3.63 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?**

3.3.63.1 Example Read Error Message from Error Queue

typo
SYST:ERR?
-113,"Undefined header"
freq27THz
pow -173dBm
SYS:ERR?;ERR?;ERR?
200,"FREQuency out of range; 1230196.22e6 outside of range [10.00e6,20000.00e6]"
;300,"Power out of range; -173.0dBm outside of range [-36.0,27.0]dBm";0,"No error"

3.3.64 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**

3.3.64.1 Example: Local Lockout with KLOCk

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

3.3.65 SYSTem:PRESet

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

3.3.66 *TRG – Trigger

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

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

***TRG** is equivalent to **TRIGger**.

3.3.66.1 Example: *TRG – Trigger

```
*rst
SYST:ERR:BEH IMM
*trg
-211,"Trigger_ignored; Trigger Source BUS not selected"
trig:sour bus
*trg
-211,"Trigger_ignored; Trigger Source BUS not selected"
freq:mode swe
init
*trg
format:sreg hex
status:oper:cond?
#H08
abor
swe:count 5
init
status:oper:cond?
#H20
trigger
status:oper:cond?
#H08
```

3.3.67 TRIGger[:SEQuence][:IMMediate]

TRIGger is the same as ***TRG**.

3.3.68 TRIGger[:SEQuence][: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	Trigger or GBIP GET will start the sweep
EXTernal	A rising edge (TTL) on the trigger IO BNC starts the sweep.

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

3.3.68.1 Example: Trigger Source

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

3.3.69 *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**

3.3.69.1 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

SALES CONTACT INFORMATION:

Hittite Microwave Worldwide Sales Offices

Hittite Microwave Corporation
USA Corporate Headquarters
Phone: 978-250-3343
Fax: 978-250-3373
sales@hittite.com

Eastern North America
Phone: 978-337-1226
usa-east@hittite.com

Central (North) North America
Phone: 312-485-8730
usa-north@hittite.com

Central (South) North America
Phone: 978-808-0652
usa-south@hittite.com

Western North America
Phone: 817-727-7146
usa-west@hittite.com

Mexico & South America
Phone: 978-250-3343 x1250
southamerica@hittite.com

Western & Southern Europe
Hittite Microwave Europe Limited
Phone: +44-870-7664355
Fax: +978-250-3373
europe@hittite.com

Northern Europe
Hittite Microwave Europe Limited
Phone: +44-870-7664355
Fax: +978-250-3373
nordic@hittite.com

Central & Eastern Europe
Hittite Microwave Deutschland GmbH
Phone: +49-8031-97654
Fax: +49-8031-98883
germany@hittite.com

Republic of Korea
Hittite Microwave Asia Co., Limited
Phone: +82-2559-0638
Fax: +82-2559-0639
korea@hittite.com

Peoples Republic of China
Hittite Microwave Co., Limited
Shanghai Office
Phone: +86-21-6209-8809
Fax: +86-21-6209-6730
china@hittite.com

Shenzhen Office
Phone: +86-755-3322-2116
Fax: +86-755-3322-2117
shenzhen@hittite.com

Japan
Hittite KK
Phone: +81-3-6853-6654
japan@hittite.com

Instrumentation Sales Representatives

North America

CT, MA, ME, E. PA, NH,
So. NJ, RI & VT:
GTEK, LLC
(Instrumentation Only)
978-692-4954

TX:
Testech Inc.
972-644-5010

Europe, Mid-East & Africa

Finland & Sweden:
RF Partner AB
+46-31-475100

France:
MB Electronique
+33-1-39-67-67-67

Greece:
Vector Technologies LTD
+30-210-68-58-008

Turkey:
Spark Olcum Teknolojileri Ltd. S
+90 (312) 466 821

Asia & South Pacific

Japan:
Farad
+82-3-5261-3091