LibreCAL SCPI Programming Guide

November 10, 2022

Contents

1 Introduction

The LibreCAL firmware contains its own SCPI parser. The LibreCAL-GUI is not necessary to use the SCPI commands.

There are two ways to send SCPI commands:

- **Virtual COM Port (VCP):** The LibreCAL implements a VCP over which SCPI commands can be received.
- **Custom USB interface:** The LibreCAL implements a custom USB interface over which SCPI commands can be received.

Only one of these options should be used at a time. The advantage of the VCP is ease of use (any terminal will work), while the custom interface provides easier device identification (no need to manually assign COM ports) and is used by the LibreCAL-GUI.

2 General Syntax

The syntax follows most SCPI rules:

- Commands are case **sensitive**. While non-standard, this allows for more flexibility when naming coefficients. Only upper case characters must be used for the events and queries, lower case characters are only allowed for the arguments.
- The command tree is organized in branches, separated by a colon:

```
:TEMPerature:STABLE?
```

- If a command starts with a colon it is evaluated from the root branch, otherwise the last used branch is assumed:

```
:TEMPerature:STABLE?
STABLE? #No colon, TEMPerature branch was used before
```

- Branches and commands can be abbreviated by using only the uppercase part of their name, the following commands are identical:

```
:HEATer:POWer?
:HEAT:POW?
```

- Every command generates a (possibly empty) response, terminated with a newline character.
- Some commands require additional arguments that have to be passed after the command (separated by spaces):

```
:TEMPerature 35
```

- Two types of commands are available:
  - **Events** change a setting or trigger an action. They usually have an empty response (unless there was an error).
  - **Queries** request information. They end with a question mark.

  Some commands are both events and queries, depending on whether the question mark is present:

```
:PORT? 1 # Returns the currently used standard at port 1
:PORT 1 OPEN # Configures port 1 to use the OPEN standard
```

3 Commands

3.1 General Commands

3.1.1 *IDN

Query:

Effect:	Returns the identifications string
Syntax:	*IDN?
Parameters:	None
Return value:	LibreCAL_<serial>

3.1.2 *LST

Query:

Effect:	Lists all available commands
Syntax:	*LST?
Parameters:	None
Return value:	List of commands, separated by newline

3.1.3 :FIRMWARE

Query:

Effect:	Returns the firmware version
Syntax:	:FIRMWARE?
Parameters:	None
Return value:	<major>.<minor>.<patch>

3.1.4 :BOOTloader

Event:

Effect:	Reboots and enters the bootloader mode
Syntax:	:BOOTloader
Parameters:	None

This is equivalent to pressing the "BOOTSEL" button when applying power.

3.2 Port Control Commands

This section contains commands to control and check the configuration of the ports.

3.2.1 :PORTS

Query:

Effect:	Returns the number of available ports
Syntax:	:PORTS?
Parameters:	None
Return value:	Number of ports, usually 4

3.2.2 :PORT

Event:

Effect:	Set a port standard
Syntax:	:PORT <port> <standard>
Parameters:	<port> Port number, 1-4 <standard> Selected standard, one of: OPEN SHORT LOAD THROUGH NONE

Query:

Effect:	Returns the currently used standard at a port
Syntax:	:PORT? <port>
Parameters:	<port> Port number
Return value:	Standard, one of: OPEN SHORT LOAD THROUGH NONE

3.2.3 :PORT:VALID

Query:

Effect:	Checks whether the current port configuration is valid (either none or exactly two ports set to THROUGH)
Syntax:	:PORT:VALID?
Parameters:	None
Return value:	TRUE or FALSE

3.3 Temperature Control Commands

This section contains commands related to the temperature regulation of the LibreCAL. While it is possible to change the target temperature, this is not recommended as calibration coefficients will no longer be accurate due to temperature drift.

3.3.1 :TEMPerature

Event:

Effect:	Sets the target temperature
Syntax:	:TEMPerature <temp>
Parameters:	<temp> Target temperature in celsius

Query:

Effect:	Returns the measured temperature

Syntax:	:TEMPerature?
Parameters:	NONE
Return value:	float value, current temperature in celsius

3.3.2 :TEMPerature:STABLE

Query:

Effect:	Checks whether the target temperature has been reached
Syntax:	:TEMPerature:STABLE?
Parameters:	NONE
Return value:	TRUE or FALSE

3.3.3 :HEATer:POWer

Query:

Effect:	Returns the currently used power by the heater
Syntax:	:HEATer:POWer?
Parameters:	NONE
Return value:	float value, heater power in watt

3.4 Coefficient Commands

This section containg commands related to reading, creating and deleting calibration coefficients.

Some definitions:

- **Coefficient data:** S parameters for a specific standard at a given port and selected frequency. It consists of one complex value for reflection standards (open, short, load) and four complex values for transmission standards (through).
- **Coefficient:** A list of coefficient data, containing multiple points over a frequency range. Each coefficient within a coefficient set is identified by its coefficient name. This name can be any of the following:
  - P1_OPEN
  - P1_SHORT
  - P1_LOAD
  - P2_OPEN
  - P2_SHORT
  - P2_LOAD
  - P3_OPEN
  - P3_SHORT
  - P3_LOAD
  - P4_OPEN
  - P4_SHORT
  - P4_LOAD
  - P12_THROUGH
  - P13_THROUGH
  - P14_THROUGH
  - P23_THROUGH
  - P24_THROUGH
  - P34_THROUGH
- **Coefficient set:** A named set of coefficients, contains all the different coefficients that are required

to cover every port setting. The coefficient set is identified by its set name.

3.4.1 :COEFFicient:LIST

Query:

Effect:	Returns the names of all available coefficient sets
Syntax:	:COEFFicient:LIST?
Parameters:	NONE
Return value:	comma-separated list of coefficient set names

3.4.2 :COEFFicient:DELete

Event:

Effect:	Deletes a calibration coefficient
Syntax:	:COEFFicient:DELete <set name> <coefficient name>
Parameters:	<set name> Name of the coefficient set <coefficient name> Name of the coefficient

3.4.3 :COEFFicient:NUMber

Query:

Effect:	Returns the number of data points within a coefficient
Syntax:	:COEFFicient:NUMber? <set name> <coefficient name>
Parameters:	<set name> Name of the coefficient set <coefficient name> Name of the coefficient>
Return value:	Integer, number of data points in coefficient

3.4.4 :COEFFicient:GET

Query:

Effect:	Returns coefficient data from a coefficient
Syntax:	:COEFFicient:GET? <set name> <coefficient name> <index>
Parameters:	<set name> Name of the coefficient set <coefficient name> Name of the coefficient> <index> Data point number
Return value:	comma-separated list of float values

The first value returned is the frequency (in GHz), followed by the S parameters. Each S parameter is split into two float values, the first value is the real part, the second value the imaginary part. For reflection standards, only one S parameter is returned (S11). For transmission standards, four S parameters are returned in S11, S21, S12, S22 order.

3.4.5 :COEFFicient:CREATE

Event:

Effect:	Creates a new calibration coefficient
Syntax:	:COEFFicient:CREATE <set name> <coefficient name>

Parameters:	<set name> Name of the coefficient set <coefficient name> Name of the coefficient

If the coefficient already exists, it will be deleted first (along with all its coefficient data). Afterwards, a new and empty coefficient will be created.

This command should be used in conjunction with :COEFFicient:ADD and :COEFFicient:FINish.

3.4.6 :COEFFicient:ADD

Event:

Effect:	Add a datapoint to an existing coefficient
Syntax:	:COEFFicient:ADD <frequency> <S parameters>
Parameters:	<frequency> Frequency of the datapoint, in GHz <S parameters> S parameters, split into real and imaginary parts

This command should be used in conjunction with :COEFFicient:CREATE and :COEFFicient:FINish. It must only be used after a coefficient has been created using :COEFFicient:CREATE and before the coefficient has been completed by :COEFFicient:FINish.

The order of values in the S parameter argument is the same as for the :COEFFicient:GET command, except that values are separated by spaces instead of a comma. Reflection standard require one S parameter (two arguments), transmission standards require four S parameters (eight arguments).

3.4.7 :COEFFicient:FINish

Event:

Effect:	Completes the creation of a coefficient
Syntax:	:COEFFicient:FINish
Parameters:	None

This command should be used in conjunction with :COEFFicient:CREATE and :COEFFicient:Add. It must be used after all data has been added to the coefficient.

3.4.8 :FACTory:ENABLEWRITE

The default coefficient set ("FACTORY") is read-only to prevent accidentally overwriting or deleting these important coefficients. Unless you are building your own LibreCAL, you should never change these coefficients!

This command makes all coefficient sets writable. DO NOT USE UNLESS YOU KNOW WHAT YOU ARE DOING!

Once issued, the default coefficient set stays writable until the next reboot.

Event:

Effect:	Make all coefficient sets writable
Syntax:	:FACTory:ENABLEWRITE <key>
Parameters:	<key> Prevents accidental usage, set to "I_AM_SURE"