

# LibreCAL SCPI Programming Guide

March 18, 2023

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>General Syntax</b>	<b>2</b>
<b>3</b>	<b>Commands</b>	<b>2</b>
3.1	General Commands . . . . .	2
3.1.1	*IDN . . . . .	2
3.1.2	*LST . . . . .	3
3.1.3	:FIRMWARE . . . . .	3
3.1.4	:BOOTloader . . . . .	3
3.2	Port Control Commands . . . . .	3
3.2.1	:PORTS . . . . .	3
3.2.2	:PORT . . . . .	3
3.3	Temperature Control Commands . . . . .	4
3.3.1	:TEMPerature . . . . .	4
3.3.2	:TEMPerature:STABLE . . . . .	5
3.3.3	:HEATer:POWer . . . . .	5
3.4	Coefficient Commands . . . . .	5
3.4.1	:COEFFicient:LIST . . . . .	6
3.4.2	:COEFFicient:DELeTe . . . . .	6
3.4.3	:COEFFicient:NUMber . . . . .	6
3.4.4	:COEFFicient:GET . . . . .	6
3.4.5	:COEFFicient:CREATE . . . . .	7
3.4.6	:COEFFicient:ADD_COMMENT . . . . .	7
3.4.7	:COEFFicient:ADD . . . . .	7
3.4.8	:COEFFicient:FINish . . . . .	7
3.4.9	:FACTory:ENABLEWRITE . . . . .	8

# 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:

- Arguments are case **sensitive**. While non-standard, this allows for more flexibility when naming coefficients. The command is still case **insensitive**.
- The command tree is organized in branches, separated by a colon:

```
:TEMPerature:STABLE?
```

- 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
```

- Multiple commands per line (separated by a semicolon) are **not** supported.

## 3 Commands

### 3.1 General Commands

#### 3.1.1 \*IDN

Query:

<b>Effect:</b>	Returns the identifications string
<b>Syntax:</b>	*IDN?

<b>Parameters:</b>	None
<b>Return value:</b>	LibreCAL, LibreCAL, <serial>, <firmware_version>

### 3.1.2 \*LST

Query:

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

### 3.1.3 :FIRMWARE

Query:

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

### 3.1.4 :BOOTloader

Event:

<b>Effect:</b>	Reboots and enters the bootloader mode
<b>Syntax:</b>	:BOOTloader
<b>Parameters:</b>	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:

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

### 3.2.2 :PORT

Event:

<b>Effect:</b>	Set a port standard
<b>Syntax:</b>	:PORT <port> <standard> [<dest>]

<b>Parameters:</b>	<port> Port number, 1-4 <standard> Selected standard, one of: OPEN SHORT LOAD THROUGH NONE <dest> Destination port of other the end, 1-4 (only required for THROUGH)
--------------------	---

When a port is set to through, <dest> is automatically also set to through (with <port> as its <dest>). Setting a port that is currently set to through to any other standard, sets the <dest> of that port to NONE.

Example:

```
# Sets port 1 to LOAD standard
:PORT 1 LOAD
# Creates a through between port 2 and 3. This configures both ports
# as THROUGH with each other as the destination ports
:PORT 2 THROUGH 3
# Returns "THROUGH 2" because port 3 was also configured by the
# previous command
:PORT? 3
# Sets port 2 to SHORT standard. Because it was previously
# configured as THROUGH to port 3, port 3 will be set to NONE
:PORT 2 SHORT
# Returns "NONE"
:PORT? 3
```

Query:

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

### 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:

<b>Effect:</b>	Sets the target temperature
<b>Syntax:</b>	:TEMPerature <temp>

<b>Parameters:</b>	<temp> Target temperature in celsius
--------------------	--------------------------------------

Query:

<b>Effect:</b>	Returns the measured temperature
<b>Syntax:</b>	:TEMPerature?
<b>Parameters:</b>	NONE
<b>Return value:</b>	float value, current temperature in celsius

### 3.3.2 :TEMPerature:STABLE

Query:

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

### 3.3.3 :HEATer:POWer

Query:

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

## 3.4 Coefficient Commands

This section contains 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

- P<sub>14</sub>\_THROUGH
- P<sub>23</sub>\_THROUGH
- P<sub>24</sub>\_THROUGH
- P<sub>34</sub>\_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:

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

#### 3.4.2 :COEFFicient:DELeTe

Event:

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

#### 3.4.3 :COEFFicient:NUMber

Query:

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

#### 3.4.4 :COEFFicient:GET

Query:

<b>Effect:</b>	Returns coefficient data from a coefficient
<b>Syntax:</b>	:COEFFicient:GET? <set name> <coefficient name> <index>
<b>Parameters:</b>	<set name> Name of the coefficient set <coefficient name> Name of the coefficient <index> Data point number
<b>Return value:</b>	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 (S<sub>11</sub>). For transmission standards, four S parameters are returned in S<sub>11</sub>, S<sub>21</sub>, S<sub>12</sub>, S<sub>22</sub> order.

### 3.4.5 :COEFFicient:CREATE

#### Event:

<b>Effect:</b>	Creates a new calibration coefficient
<b>Syntax:</b>	:COEFFicient:CREATE <set name> <coefficient name>
<b>Parameters:</b>	<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\_COMMENT, :COEFFicient:ADD and :COEFFicient:FINish.

### 3.4.6 :COEFFicient:ADD\_COMMENT

#### Event:

<b>Effect:</b>	Add ASCII text comment before to add coefficient datapoint
<b>Syntax:</b>	:COEFFicient:ADD_COMMENT <ASCII text comment>
<b>Parameters:</b>	<ASCII text comment> add an ASCII text comment with size up to 120chars if the comment exceed 120 characters it will be truncated to 120 characters

This command should be used only after :COEFFicient:CREATE as it is intended to add comment before to add any datapoint (:COEFFicient:ADD).

There is a limitation of maximum 100 comments per calibration coefficient (:COEFFicient:CREATE)(if there is more comment the command will return an error and comment will be not added).

### 3.4.7 :COEFFicient:ADD

#### Event:

<b>Effect:</b>	Add a datapoint to an existing coefficient
<b>Syntax:</b>	:COEFFicient:ADD <frequency> <S parameters>
<b>Parameters:</b>	<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).

### 3.4.8 :COEFFicient:FINish

#### Event:

<b>Effect:</b>	Completes the creation of a coefficient
<b>Syntax:</b>	:COEFFicient:FINish
<b>Parameters:</b>	None

This command should be used in conjunction with :COEFFicient:CREATE and :COEFFi-

cient:Add. It must be used after all data has been added to the coefficient.

### 3.4.9 :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:**

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