SLICE-QTC API

Links

* **Quick-Start Guides**

- SLICE-QTC Quick Start

* **Electronics**

- SLICE-QT Manual
- SLICE-QTC firmware upgrade instructions

* **Websites**

- SLICE-QT web page
- Github page for SLICE-QTC GUI
- Github page for SLICE-QTC firmware revisions

Implementation Instructions

Listed below is the command set for the Application Programming Interface (API). Most operations that can be performed via the touchscreen can be also be performed via the API.

Computer communication to the SLICE instrument occurs by first establishing a Serial COM port via the USB 2.0 physical interface. Computer communication should be platform independent, though appropriate USB drivers may have to be installed. The following table shows the serial port settings:

Baud Rate	9600
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

To perform an operation, an ASCII-based text message is first constructed from a command and its arguments, and then the message is transmitted to the SLICE instrument through the COM port. The following rules apply:

1. Syntax is *insensitive* to case.
2. A message is an ASCII string composed of a command followed by zero or more arguments.
3. A command and its arguments are delimited by spaces.
4. Messages are terminated by a carriage return (“\r”).
5. All valid messages return an ASCII string value.
6. [Int] refers to an integer argument that has no decimal point.
7. [Float] refers to a floating point argument that has a decimal point in its value.

SLICE-QT Specific Notes

1. Valid values for the CHANNEL argument are {1, 2, 3, 4}.
2. All enable commands use an integer argument to convey the state: 0 = Off, 1 = On.
3. All temperatures are set and returned in degrees Celsius [˚C].
4. All currents are set and returned in Amperes [A].

Temperature Settings

TempSet?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TempSet? 3
26.28
```

I_2C Command Number:

Returns the temperature set point for CHANNEL.

TempSet

Description

Arguments:

```
[Int] CHANNEL
[Float] TEMPERATURE
```

Example:

```
TempSet 3 26.283
26.282
```

I_2C Command Number:

Sets the temperature set point for CHANNEL to TEMPERATURE. Returns **TempSet?**. The temperature set point cannot be set outside the range set by the minimum and maximum allowed temperatures

(set by the user). If TEMPERATURE is outside this range, the set point will not be adjusted. Also, the temperature set point will be coerced to a value that has a internal digital representation closest to TEMPERATURE.

Temp?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Temp? 2
25.398
```

I_2C Command Number:

Returns the actual (measured) temperature for CHANNEL.

TError?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TError? 2
0.0024
```

I_2C Command Number:

Returns the temperature error for CHANNEL in degrees Celsius. The error is calculated as the set point temperature minus the actual temperature.

TempMin?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TempMin? 2
5.000
```

I_2C Command Number:

Returns the minimum allowed temperature for CHANNEL.

TempMin

Description

Arguments:

```
[Int] CHANNEL
[Float] TEMPERATURE
```

Example:

```
TempMin 3 -5.000
-5.000
```

I_2C Command Number:

Sets the minimum allowed temperature for CHANNEL to TEMPERATURE. Returns **TempMin?**.

TempMax?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TempMax? 2
55.000
```

I_2C Command Number:

Returns the maximum allowed temperature for CHANNEL.

TempMax

Description

Arguments:

```
[Int] CHANNEL
[Float] TEMPERATURE
```

Example:

```
TempMax 3 55.000
55.000
```

I_2C Command Number:

Sets the maximum allowed temperature for CHANNEL to TEMPERATURE. Returns **TempMax?**.

Transducer/Load Settings

Bipolar?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Bipolar? 2
1
```

I_2C Command Number:

Returns whether CHANNEL is configured for either Bipolar operation (for a thermo-electric cooler) or Unipolar operation (for a resistive heater).

- 1 = On = Bipolar operation, i.e., current is bi-directional through transducer.
- 0 = Off = Unipolar operation, i.e., current is uni-directional through transducer.

Bipolar

Description

Arguments:

```
[Int] CHANNEL
[Int] STATE
```

Example:

```
Bipolar 3 0
0
```

I_2C Command Number:

Sets the operational STATE for CHANNEL to be either Bipolar (for a thermo-electric cooler) or Unipolar (for a resistive heater). Returns **Bipolar?**. STATE can have the following values:

- 1 = On = Bipolar operation, i.e., current is bi-directional through transducer.
- 0 = Off = Unipolar operation, i.e., current is uni-directional through transducer.

MaxCurr?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
MaxCurr? 2
2.000
```

I_2C Command Number:

Returns the current limit for CHANNEL in Amps [A].

MaxCurr

Description

Arguments:

```
[Int] CHANNEL
[Float] CURRENT
```

Example:

```
MaxCurr 1 1.500
1.500
```

I_2C Command Number:

Sets the current limit for CHANNEL to CURRENT in Amps [A]. Returns **MaxCurr?**. Allowed values are {0.000 ... 6.000}.

Current?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Current? 2
0.654
```

I_2C Command Number:

Returns the measured current flowing through the temperature transducer (e.g. TEC or heater) in Amps [A].

MaxPwr?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
MaxPwr? 2
7.000
```

I_2C Command Number:

Returns the power limit for CHANNEL in Watts [W].

MaxPwr

Description

Arguments:

```
[Int] CHANNEL
[Float] POWER
```

Example:

```
MaxPwr 1 10.000
10.000
```

I_2C Command Number:

Sets the power limit for CHANNEL to POWER in Watts [W]. Returns **MaxPwr?**. Allowed values are {0.000 ... 20.000}.

Power?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Power? 2
1.582
```

I_2C Command Number:

Returns the measured power output for CHANNEL in Watts [W].

CVolt?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
CVolt? 4
2.573
```

I_2C Command Number:

Returns the voltage across the load for CHANNEL in Volts [V].

Thermistor Settings

Beta?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Beta? 1
3450.0000
```

I_2C Command Number:

For the B-parameter thermistor model: Returns the Beta coefficient in Kelvin [K] for CHANNEL.

Beta

Description

Arguments:

```
[Int] CHANNEL
[Float] VALUE
```

Example:

```
Beta 2 3450
3450.0
```

I_2C Command Number:

For the B-parameter thermistor model: Sets the Beta coefficient in Kelvin [K] to VALUE for CHANNEL. Returns **Beta?**.

RefTemp?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
RefTemp? 1
25.0
```

I_2C Command Number:

For the B-parameter thermistor model: Returns the Reference Temperature in degrees Celsius [˚C] for CHANNEL.

RefTemp

Description

Arguments:

```
[Int] CHANNEL
[Float] VALUE
```

Example:

```
RefTemp 2 25.0
25.0
```

I_2C Command Number:

For the B-parameter thermistor model: Sets the Reference Temperature in degrees Celsius [˚C] to VALUE for CHANNEL. Returns **RefTemp?**.

RefRes?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
RefRes? 1
10000.0
```

I_2C Command Number:

For the B-parameter thermistor model: Returns the Reference Resistance in Ohms [Ω] (at the Reference Temperature) for CHANNEL.

RefRes

Description

Arguments:

```
[Int] CHANNEL
[Float] VALUE
```

Example:

```
RefRes 2 10000.0
10000.0
```

I_2C Command Number:

For the B-parameter thermistor model: Sets the Reference Resistance in Ohms [Ω] to VALUE for CHANNEL. Returns **RefRes?**.

TCoefA?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TCoefA? 1
2.108508173
```

I_2C Command Number:

For the Steinhart-Hart thermistor model: Returns the A coefficient for CHANNEL.

TCoefA

Description

Arguments:

```
[Int] CHANNEL
[Float] VALUE
```

Example:

```
TCoefA 1 2.108508173
2.108508173
```

I_2C Command Number:

For the Steinhart-Hart thermistor model: Sets the A coefficient to VALUE for CHANNEL. Returns **TCoefA?**.

TCoefB?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TCoefB? 1
0.797204727
```

I_2C Command Number:

For the Steinhart-Hart thermistor model: Returns the B coefficient for CHANNEL.

TCoefB

Description

Arguments:

```
[Int] CHANNEL
[Float] VALUE
```

Example:

```
TCoefB 1 0.797204727
0.797204727
```

I_2C Command Number:

For the Steinhart-Hart thermistor model: Sets the B coefficient to VALUE for CHANNEL. Returns **TCoefB?**.

TCoefC?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TCoefC? 1
6.535076315
```

I_2C Command Number:

For the Steinhart-Hart thermistor model: Returns the C coefficient for CHANNEL.

TCoefC

Description

Arguments:

```
[Int] CHANNEL
[Float] VALUE
```

Example:

```
TCoefC 1 6.535076315
6.535076315
```

I_2C Command Number:

For the Steinhart-Hart thermistor model: Sets the C coefficient to VALUE for CHANNEL. Returns **TCoefC?**.

Loop Filter Settings

Control?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Control? 2
3
```

I_2C Command Number:

Returns the control mode for CHANNEL:

- 0 = Manual Mode Off
- 1 = Servo Mode Off
- 2 = Manual Mode On
- 3 = Servo Mode On

Control

Description

Arguments:

```
[Int] CHANNEL
[Int] MODE
```

Example:

```
Control 2 3
3
```

I_2C Command Number:

Sets the control MODE for CHANNEL:

- 0 = Manual Mode Off
- 1 = Servo Mode Off
- 2 = Manual Mode On
- 3 = Servo Mode On

PGain?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
PGain? 2
1.8
```

I_2C Command Number:

Returns the proportional gain for CHANNEL.

PGain

Description

Arguments:

```
[Int] CHANNEL
[Float] GAIN
```

Example:

```
PGain 3 1.8
1.8
```

I_2C Command Number:

Sets the proportional gain for CHANNEL to GAIN. Returns **PGain?**.

PGainEn?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
PGainEn? 2
1
```

I_2C Command Number:

Returns the On/Off STATE of the proportional gain for CHANNEL.

PGainEn

Description

Arguments:

```
[Int] CHANNEL
[Int] STATE
```

Example:

```
PGainEn 4 0
0
```

I_2C Command Number:

Sets the On/Off STATE of the proportional gain for CHANNEL. Returns **PGainEn?**.

Integ?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Integ? 1
2.000
```

I_2C Command Number:

Returns the integral time constant for CHANNEL in seconds [s].

Integ

Description

Arguments:

```
[Int] CHANNEL
[Float] TIME_CONSTANT
```

Example:

```
Integ 3 2.000
2.000
```

I_2C Command Number:

Sets the integral time constant for CHANNEL to TIME_CONSTANT in seconds [s]. Returns **Integ?**.

IntegEn?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
IntegEn? 2
1
```

I_2C Command Number:

Returns the On/Off STATE of the integral gain for CHANNEL.

IntegEn

Description

Arguments:

```
[Int] CHANNEL
[Int] STATE
```

Example:

```
IntegEn 3 0
0
```

I_2C Command Number:

Sets the On/Off STATE of the integral gain for CHANNEL. Returns **IntegEn?**.

Deriv?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Deriv? 1
0.500
```

I_2C Command Number:

Returns the derivative time constant for CHANNEL in seconds [s].

Deriv

Description

Arguments:

```
[Int] CHANNEL
[Float] TIME_CONSTANT
```

Example:

```
Deriv 1 0.500
0.500
```

I_2C Command Number:

Sets the derivative time constant for CHANNEL to TIME_CONSTANT in seconds [s]. Returns **Deriv?**.

DerivEn?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
DerivEn? 4
1
```

I_2C Command Number:

Returns the On/Off STATE of the derivative gain for CHANNEL.

DerivEn

Description

Arguments:

```
[Int] CHANNEL
[Int] STATE
```

Example:

```
DerivEn 3 1
1
```

I_2C Command Number:

Sets the On/Off STATE of the derivative gain for CHANNEL. Returns **DerivEn?**.

Slew?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Slew? 1
1.5
```

I_2C Command Number:

Returns the slew rate limit for CHANNEL in degrees Celsius per second [˚C/s].

Slew

Description

Arguments:

```
[Int] CHANNEL
[Float] RATE
```

Example:

```
Slew 1 1.5
1.5
```

I_2C Command Number:

Sets the slew rate limit for CHANNEL to RATE in degrees Celsius per second [˚C/s]. Returns **Slew?**.

SlewEn?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
SlewEn? 4
1
```

I_2C Command Number:

Returns the On/Off STATE of the slew rate limiter for CHANNEL.

SlewEn

Description

Arguments:

```
[Int] CHANNEL
[Int] STATE
```

Example:

```
SlewEn 3 1
1
```

I_2C Command Number:

Sets the On/Off STATE of the slew rate limiter for CHANNEL. Returns **SlewEn?**.

I/O Channel Settings (Not yet implemented 27 Feb, 2019)

Output Function	CHANNEL	FUNCTION	VALUE1	VALUE2
	[Int]	[Int]	[Float]	[Float]
Temperature Error	1 - 4	0	Gain [V/˚C]	Offset [˚C]
Temperature	1 - 4	1	Gain [V/˚C]	Offset [˚C]
Current Output	1 - 4	2	Gain [V/A]	Offset [A]

Output1?

Description

Arguments:

```
No Arguments Taken
```

Example:

```
Output1?
2, 3, 0.500, 0.0
```

I_2C Command Number:

For the front panel Output 1, returns a comma-delimited ASCII string with the following format: “CHANNEL, FUNCTION, VALUE1, VALUE2”.

Output1

Description

Arguments:

```
[Int] CHANNEL
[Int] FUNCTION
[Float] VALUE1
[Float] VALUE2
```

Example:

```
Output1 1 2 1.0 0.0
1, 2, 1.0, 0.0
```

I_2C Command Number:

Sets the FUNCTION and CHANNEL for the front panel Output 1 with the settings prescribed by VALUE1 and VALUE2. Returns **Output1?**.

Output2?

Description

Arguments:

```
No Arguments Taken
```

Example:

```
Output2?
```

```
2, 3, 0.500, 0.0
```

I_2C Command Number:

For the front panel Output 2, returns a comma-delimited ASCII string with the following format: “CHANNEL, FUNCTION, VALUE1, VALUE2”.

Output2

Description

Arguments:

```
[Int] CHANNEL
[Int] FUNCTION
[Float] VALUE1
[Float] VALUE2
```

Example:

```
Output2 1 2 1.0 0.0
1, 2, 1.0, 0.0
```

I_2C Command Number:

Sets the FUNCTION and CHANNEL for the front panel Output 2 with the settings prescribed by VALUE1 and VALUE2. Returns **Output2?**.

Input Function	CHANNEL	FUNCTION	VALUE1	VALUE2	VALUE3
	[Int]	[Int]	[Float]	[Float]	[Int]
Slow Servo	1 - 4	0	Integral Gain	Set point voltage [V]	Polarity [0/1 = +/-]
External Temperature	1 - 4	1	Gain [˚C/V]	Offset [V]	N/A
External Error	1 - 4	2	Gain [˚C/V]	Offset [V]	N/A
External Setpoint	1 - 4	3	Gain [˚C/V]	Offset [V]	N/A
Feedforward	1 - 4	4	Gain [A/V]	Offset [V]	N/A

InputA?

Description

Arguments:

```
No Arguments Taken
```

Example:

```
InputA?
```

```
2, 3, 0.500, 0.0
```

I_2C Command Number:

For the front panel Input A, returns a comma-delimited ASCII string with the following format: “CHANNEL, FUNCTION, VALUE1, VALUE2, VALUE3 (if applicable)”.

InputA

Description

Arguments:

```
[Int] CHANNEL
[Int] FUNCTION
[Float] VALUE1
[Float] VALUE2
[Int] VALUE3
```

Example:

```
InputA 1 0 1.0 0.0 0
1, 0, 1.0, 0.0, 0
```

I_2C Command Number:

Sets the FUNCTION and CHANNEL for the front panel Input B with the settings prescribed by VALUE1-3. Returns **InputA?**.

InputB?

Description

Arguments:

```
No Arguments Taken
```

Example:

```
InputB?
2, 0, 0.500, 0.0, 1
```

I_2C Command Number:

For the front panel Input B, returns a comma-delimited ASCII string with the following format: “CHANNEL, FUNCTION, VALUE1, VALUE2, VALUE3 (if applicable)”.

InputB

Description

Arguments:

```
[Int] CHANNEL
[Int] FUNCTION
[Float] VALUE1
[Float] VALUE2
[Int] VALUE3
```

Example:

```
InputB 1 0 1.0 0.0 1
1, 0, 1.0, 0.0, 1
```

I_2C Command Number:

Sets the FUNCTION and CHANNEL for the front panel Input B with the settings prescribed by VALUE1-3. Returns **InputB?**.

Error Handling
