

Quad Temperature Controller Command Set

List of commands for the ICE Quad Temperature Controller.

TempSet?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TempSet? 3  
26.283
```

I₂C Command Number: 16

Returns the temperature setpoint for channel CHANNEL.

TempSet

Description

Arguments:

```
[Int] CHANNEL  
[Float] TEMP
```

Example:

```
TempSet 3 24.55  
24.548
```

I₂C Command Number: 17

Sets the temperature setpoint for channel Channel to TEMP. Returns output from **TempSet?**.

Note: temperature setpoint cannot be set outside of the bounds of the minimum and maximum temperature. If TEMP is outside that range, temperature setpoint will not be adjusted.

BiPolar?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
BiPolar? 3  
On
```

I₂C Command Number: 18

Returns whether the temperature loop for channel CHANNEL is bipolar.

BiPolar

Description

Arguments:

```
[Int] CHANNEL  
[ASCII] On/Off
```

Example:

```
BiPolar 3 Off  
Off
```

I₂C Command Number: 19

Sets the temperature loop for channel CHANNEL bipolar (heating only) on or off. Turn Bipolar off when driving resistive heaters. Returns output from **BiPolar?**.

Servo?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Servo? 3  
Off
```

I₂C Command Number: 20

Returns the status (ON or OFF or FAULT) of the temperature servo loop for channel CHANNEL. A return of fault means that the temperature is outside the min/max temperature range and that the temperature loop has been temporarily disabled.

Servo

Description

Arguments:

```
[Int] CHANNEL  
[ASCII] On/Off
```

Example:

```
Servo 3 On  
On
```

I₂C Command Number: 21

Turns the temperature servo loop for channel CHANNEL on or off. Returns the output of **Servo?**

Temp?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Temp? 3  
24.21
```

I₂C Command Number: 22

Returns the current temperature for channel CHANNEL.

TError?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TError? 3  
0.021
```

I₂C Command Number: 23

Returns the temperature error ($T_{\text{setpoint}} - T_{\text{actual}}$) in degrees Celsius. `</div>`

Current?

Arguments:

Description

```
[Int] CHANNEL
```

Example:

```
Current? 3  
0.651
```

I₂C Command Number: 24

Returns the current flowing through TEC (or resistive heater) in Amps. [</div>](#)

TempMin?

Arguments:

Description

```
[Int] CHANNEL
```

Example:

```
TempMin? 3  
15.3
```

I₂C Command Number: 25

Returns the minimum temperature for channel CHANNEL. If the temperature is ever less than MINTEMP, the board will temporarily disable the temperature loop until the temperature rises above MINTEMP. [</div>](#)

TempMin

Arguments:

Description

```
[Int] CHANNEL  
[Float] MINTEMP
```

Example:

```
TempMin 3 14.32  
14.3
```

I₂C Command Number: 26

Sets the minimum temperature channel CHANNEL to MINTEMP. Returns **TempMin?**. Note: You cannot set MINTEMP greater than the temperature setpoint. Attempting to do so will not change the minimum temperature. </div>

TempMax?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
TempMax? 3  
55.3
```

I₂C Command Number: 27

Returns the maximum temperature for channel CHANNEL. If the temperature is ever greater than MAXTEMP, the board will temporarily disable the temperature loop until the temperature drops below MAXTEMP. </div>

TempMax

Description

Arguments:

```
[Int] CHANNEL  
[Float] MAXTEMP
```

Example:

```
TempMax 3 54.32  
54.3
```

I₂C Command Number: 28

Sets the maximum temperature channel CHANNEL to MAXTEMP. Returns **TempMax?**. Note: You cannot set MAXTEMP less than the temperature setpoint. Attempting to do so will not change the maximum temperature. </div>

Gain?

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
Gain? 3
34
```

I₂C Command Number: 29

Returns the gain setting temperature loop servo gain for channel CHANNEL. Gain can be from 1 to 255. </div>

Gain

Description

Arguments:

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

Example:

```
Gain 3 34
34
```

I₂C Command Number: 30

Sets the gain for the temperature loop servo gain for channel CHANNEL. Gain range can be 1 to 255. Returns output of **Gain?**. </div>

RecData

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
RecData 2
BUSY
```

I₂C Command Number: 31

Record data on the temperature error of channel CHANNEL with amplitude, interval and number of points set by the **RecAmp**, **RecInt**, **RecNum** functions. Returns **RecStat?** </div>

RecStat? Description

Arguments:

```
No Arguments Taken
```

Example:

```
RecStat?  
BUSY
```

I₂C Command Number: 32

Returns the status of recording data from the RecData command. Returns status of either NODATA, BUSY, FINISHED. </div>

RecInt? Description

Arguments:

```
No Arguments Taken
```

Example:

```
RecInt?  
34
```

I₂C Command Number: 33

Returns the setting for the time interval between data points when taking data with the RecData command. Time between points is 10ms * INTERVAL. Value can be from 1 to 255 (10ms - 2.55s) </div>

RecInt Description

Arguments:

```
[Int] INTERVAL
```

Example:

```
RecInt 200  
200
```

I₂C Command Number: 34

Sets the interval setting for time between data points when taking data with the RecData command. Time between points is 10mS * INTERVAL. Value can be from 1 to 255 (10ms – 2.55s). Returns output of **RecInt?** Command. </div>

RecAmp?

Description

Arguments:

```
No Arguments Taken
```

Example:

```
RecAmp?  
3
```

I₂C Command Number: 35

Returns the amplitude for data stored by the RecData command. Valid range is 0-5. When AMPLITUDE is 0, the range for the temperature error is from -18.75mK to 18.75mK in steps of 0.15mK. When AMPLITUDE is 1, the range for the temperature error is from -37.5mK to 37.5mK in steps of 0.3mK. When AMPLITUDE is 5, the range for the temperature error is from -600mK to 600mK in steps of 4.7mK.

The formula for the temperature range is $\pm 18.75\text{mK} \cdot 2^{\text{AMPLITUDE}}$ in steps of $0.15\text{mK} \cdot 2^{\text{AMPLITUDE}}$. </div>

RecAmp

Description

Arguments:

```
[Int] AMPLITUDE
```

Example:

```
RecAmp 3  
3
```

I₂C Command Number: 36

Sets the amplitude for the data stored by the **RecData** command. See **RecAmp?** for more details. </div>

RecNum?

Description

Arguments:

No Arguments Taken

Example:

```
RecNum?  
199
```

I₂C Command Number: 37

Returns the number of data points to be collected when **RecData** is run. Range is from 1 to 256. </div>

RecNum

	Description
--	-------------

Arguments:

```
[Int] NUMBER
```

Example:

```
RecNum 50  
50
```

I₂C Command Number: 38

Sets the command **RecData** to take NUMBER + 1 data points. NUMBER can be from 0 to 255. Returns **RecNum?** </div>

MaxCurr?

	Description
--	-------------

Arguments:

```
[Int] CHANNEL
```

Example:

```
MaxCurr? 2  
0.32
```

I₂C Command Number: 39

Returns the maximum TEC current allowed for the temperature loop on channel CHANNEL. </div>

MaxCurr

Description

Arguments:

```
[Int] CHANNEL  
[Float] MAXCURRENT
```

Example:

```
MaxCurr 2 0.25  
0.25
```

I₂C Command Number: 40

Sets the maximum TEC current allowed for the temperature loop on channel CHANNEL. Returns the output of **MaxCurr?**. `</div>`

From:

<https://www.vescent.com/manuals/> - **Product Manuals**

Permanent link:

<https://www.vescent.com/manuals/doku.php?id=ice:commands:quadtemp>

Last update: **2021/08/26 14:26**

