

Offset Phase Lock and Current Controller Command Set

List of commands for the ICE Offst PHase Lock and Current Controller. Please see [Common Laser Controller Command Set](#) for commands relating to the laser controller and [Common Commands to all ICE Slave Boards](#) for generic slave board commands.

The theory, usage, and performance of the ICE-CP1 can be [found here](#). The ICE-CP1 web page can be [found here](#).

N

Description

Arguments:

No Arguments Taken

Example:

N
8

I₂C Command Number: 32

Returns the value of the N divider.

N

Description

Arguments:

[Int] VALUE

Example:

N 16
16

I₂C Command Number: 33

Set the value of the N divider. Valid settings are 8,16,32 and 64. Returns output from **N?** command.

Invert

Description

Arguments:

No Arguments Taken

Example:

```
Invert  
On
```

I₂C Command Number: 34

Returns whether the error signal is being inverted or not.

Invert

Description

Arguments:

[ASCII] On/Off

Example:

```
Invert Off  
Off
```

I₂C Command Number: 35

Turns on or off inverting the error signal. Effectively changes the gain sign of the loop and whether the slave laser will be to the red or blue of the master laser. Returns output of **Invert?** Command.

IntRef

Description

Arguments:

No Arguments Taken

Example:

```
IntRef  
Off
```

I₂C Command Number: 36

Returns if the internal reference oscillator is active (as opposed to using an external reference).

IntRef

Arguments: **Description**

```
[ASCII] On/Off
```

Example:

```
IntRef On  
On
```

I₂C Command Number: 37

Enables or disables using the internal reference (instead of external reference). Returns output of **IntRef?** command.

IntFreq

Description

Arguments:

```
No Arguments Taken
```

Example:

```
IntFreq  
162.875345
```

I₂C Command Number: 38

Returns the internal reference's oscillator's frequency in MHz.

IntFreq

Description

Arguments:

```
[Float] FREQ
```

Example:

```
IntFreq 204.543765  
204.543765
```

I₂C Command Number: 39

Sets the internal reference oscillator's frequency to FREQ, where FREQ is a the frequency in MHz. Returns the output of the **IntFreq?** command.

Servo

Arguments: **Description**

No Arguments Taken

Example:

Servo
On

I₂C Command Number: 40

Returns the status of the laser servo (on or off).

Servo

Arguments: **Description**

[ASCII] On/Off

Example:

Servo On
Off

I₂C Command Number: 41

Turns on the laser servo (engages the integrator). Returns the output of the command **Servo?**.

Gain

Arguments: **Description**

No Arguments Taken

Example:

Gain
25

I₂C Command Number: 42

Returns Servo Gain. Range is from 0-64 in steps of 2 dB. 0 is a special gain setting where there is no gain (error signal does not go to integrator).

Gain

Arguments:

Description

```
[Int] GAIN
```

Example:

```
Gain 24
24
```

I₂C Command Number: 43

Sets the Servo Gain. Range is from 0-28 in steps of 2 dB. 0 is a special gain setting where there is no gain (error signal does not go to integrator). Returns the output of the command **GetGain?**

SvOffst

Arguments:

Description

```
No Arguments Taken
```

Example:

```
SvOffst
2.341
```

I₂C Command Number: 44

Returns Servo Offset voltage (in volts). When the servo is engaged, this voltage is the starting voltage that the servo integrates from.

SvOffst

Arguments:

Description

```
[Float] OFFSET
```

Example:

```
SvOffst -1.232
-1.23
```

I₂C Command Number: 45

Sets the Servo Offset voltage (in volts) to OUTPUT when the servo is turned off. When the servo is

engaged, this voltage is the starting voltage that the servo integrates from. Returns the output of the command **SvOffst?**.

ReadVolt

Description

Arguments:

```
[Int] CHANNEL
```

Example:

```
ReadVolt 4
4.234
```

I₂C Command Number: 46

Returns the voltage measured on channel CHANNEL (in volts). The channels refer to:

- 1. Servo Out
- 2. Error Signal
- 3. NA
- 4. NA
- 5. Laser Current (1V = 1A)
- 6. +2.5V Ref
- 7. NA
- 8. Ground

RampSwp

Description

Arguments:

```
No Arguments Taken
```

Example:

```
RampSwp
5.72
```

I₂C Command Number: 47

Reads the sweep range (in volts) for the ramp.

RampSwp

Description

Arguments:

[Float] VOLTAGE

Example:

RampSwp 2.64
2.65

I₂C Command Number: 48

Writes the sweep range (in volts) for the ramp. Range is from -0V to +10V. Returns the output from the command **RampSwp?**. The ramp will step from SvOffst - RampSwp/2 to SvOffst + RampSwp/2 by increments determined by RampSwp and RampNum and then return to SvOffst every time a RampRun command is initiated.

Note: RampSwp value is rounded and truncated to match range of the servo offset.

RampNum

Description

Arguments:

No Arguments Taken

Example:

RampNum
100

I₂C Command Number: 49

Returns the number of data points to be acquired during the ramp. Number of data points is also the number of steps for the ramp.

Note that the amount of bytes stored by the ramp is 2*RampNum. The amount of bytes stored set how much data needs to be read back via the **ReadBlk** command.

RampNum

Description

Arguments:

[Int] NUMBER

Example:

RampNum 100
100

I₂C Command Number: 50

Sets the number of data points to be acquired during the ramp. Number of data points is also the number of steps for the ramp.

RampRun

Description

Arguments:

No Arguments Taken

Example:

RampRun
Busy

I₂C Command Number: 51

Begins ramping the servo output and taking data according to the values set for **RampSwp** and **SvOffst** as described above. The Ramp begins at SvOffst - RampSwp/2 and ends of SvOffst + RampSwp/2. When the ramp is completed, the servo output is returned to its starting value of SvOffst. Data can be retrieved with the **ReadBlk** command. Board will not respond to any new commands while taking data. Returns status of execution of the ramp, either Busy or Finished. Returns fault if ramp is misconfigured. See **RampNum?** for details. Returns fail if laser is off.

Poles

Description

Arguments:

No Arguments Taken

Example:

Poles
3 2

I₂C Command Number: 52

Returns two columns of data. First column is a number 1-5 representing the integrator pole value. Values are 3 kHz, 10 kHz, 30 kHz, 100 kHz and 300 kHz. 1 is the slowest integrator (3 kHz), and 5 is the fastest (300 kHz). The second column is a number 0 - 4 representing differential pole value. Values are Off, 10 kHz, 30 kHz, 100 kHz and 300 kHz. 0 is is no differential pole (Off), 1 is the slowest pole setting (10 kHz) and 4 is the fastest pole setting (300 kHz).

Poles

Description

Arguments:

```
[Int] INTEGRATOR
[Int] DIFFERENTIAL
```

Example:

```
Poles 3 2
12
```

I₂C Command Number: 53

Sets the integrator and differential pole positions. INTEGRATOR is a value from 1-5 corresponding to 3 kHz (1) up to 300 kHz (5). DIFFERENTIAL is a value from 0-4 with 0 disabling differential, 1 is 10 kHz and 4 is 300 kHz. See **Poles?** for full list of all pole values.

EvtJump

Description

Arguments:

```
No Arguments Taken
```

Example:

```
EvtJump
6
```

I₂C Command Number: 54

Reads the event address for jumping the offset phase lock frequency. Address range is 0-7 where address 0 is no event.

EvtJump

Description

Arguments:

```
[Int] ADDRESS
```

Example:

```
EvtJump 4
4
```

I₂C Command Number: 55

Sets the event address for jumping the offset phase lock frequency. Address range is 0-7 where address 0 is no event.

EvtLOff

Description

Arguments:

No Arguments Taken

Example:

EvtLOff
4

I₂C Command Number: 56

Reads the event address for turning the laser off. Address range is 0-7 where address 0 is no event.

EvtLOff

Description

Arguments:

[Int] ADDRESS

Example:

EvtLOff 2
2

I₂C Command Number: 57

Sets the event address for turning the laser off. Address range is 0-7 where address 0 is no event.

EvtData

Description

Arguments:

[Int] ROW
[Int] COLUMN
[Float] VALUE

Example:

EvtData 4 3 203.324

Success

I₂C Command Number: 58



This function only works as described with firmware 2.3 and higher. If using an older firmware, please upgrade to the latest firmware before using this function.

Loads data into the event system for jumping the offset phase lock frequency when a jump frequency event is received. The **ROW** sets when this frequency jump will be engaged; **ROW**=1 means the first event will trigger this jump value. **ROW**=3 will run on the 3rd event. Events loop back based on the number of event points (set by the EVTNUM command), so if there are 5 event points, then the **ROW** that is jumped will be sequentially: 1,2,3,4,5,1,2,3,4,5,1....

The **COLUMN** sets the column of data to be written. Column 0 sets the **MODE** (see below for explanation). Column 1 sets the reference frequency. Column 2 sets the amount of feedforward to jump the servo output to help the servo quickly acquire lock at the new offset frequency.

The data table for storing the settings looks like the following:

	Columnn 0 (Mode)	Column 1 (Ref. Freq)	Column 2 (FeedForward)
Row 1	3	102.34	0.435
Row 2	5	88.43	-0.23
Row 3	9	105.28	-0.76

The **MODE** sets the whether to invert the error signal, whether to use the internal or external frequency reference and the N value. Below is a table showing the different modes:

Mode	N	Invert	Internal VCO
0	8	Off	Off
1	8	Off	On
2	8	On	Off
3	8	On	On
4	16	Off	Off
5	16	Off	On
6	16	On	Off
7	16	On	On
8	32	Off	Off
9	32	Off	On
10	32	On	Off
11	32	On	On
12	64	Off	Off
13	64	Off	On
14	64	On	Off
15	64	On	On

The command returns SUCCESS if the data is successfully enter, FAIL if not. Data loaded is saved to memory with the **SAVE** command.

EvtNum

Arguments:

Description

No Arguments Taken

Example:

EvtNum
7

I₂C Command Number: 59

Reads the number of offset phase lock values to loop through via the offset phase lock jump event system. Range from 2 - 8. If set to 4, then the sequence of offset phase lock values will be 1,2,3,4,1,2,3,4,1... where the numbers correspond to the **ELEMENT NUMBER** set when loading data in with the EvtData command.

EvtNum

Arguments:

Description

[Int] NUMBER

Example:

EvtNum 5
5

I₂C Command Number: 60

Sets the number of offset phase lock values to loop through via the offset phase lock jump event system. Range from 2 - 8. If set to 4, then the sequence of offset phase lock values will be 1,2,3,4,1,2,3,4,1... where the numbers correspond to the **ELEMENT NUMBER** set when loading data in with the EvtData command.

EvtJRow

Arguments:

Description

No Arguments Taken

Example:

EvtJRow
3

I₂C Command Number: 61

Return the next row that will be read from the Event Jump Table when the device gets an event matching the event set with the **EvtJUMP** command.

EvtData

Description

Arguments:

```
[Int] ROW  
[Int] COLUMN
```

Example:

```
EvtData 4 3  
3
```

I₂C Command Number: 62

Reads back the data in the Jump Table, using the ROW and COLUMN to set what datapoint to read back. See **EvtData** for more details.

From:

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

Permanent link:

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

Last update: **2023/11/30 21:32**

