TEXAS INSTRUMENTS

TI-Innovator™
Hub Commands

This guidebook applies to TI-Innovator™ Hub software version 1.1. To obtain the latest version of the documentation, go to *education.ti.com/guides*.

Important Information

Except as otherwise expressly stated in the License that accompanies a program, Texas Instruments makes no warranty, either express or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding any programs or book materials and makes such materials available solely on an "as-is" basis. In no event shall Texas Instruments be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of these materials, and the sole and exclusive liability of Texas Instruments, regardless of the form of action, shall not exceed the amount set forth in the license for the program. Moreover, Texas Instruments shall not be liable for any claim of any kind whatsoever against the use of these materials by any other party.

Learning More with the TI-Innovator™ System eGuide

Parts of this document refer you to the TI-Innovator™ System eGuide for more details. The eGuide is a Web-based source of TI-Innovator™ information, including:

- Programming with the TI CE Family of Graphing Calculators and TI-Nspire™ Technology, including sample programs.
- Available I/O Modules and their commands.
- Available breadboard components and their commands.
- Link to update the TI-Innovator™ Sketch software.
- Free classroom activities for TI-Innovator™ Hub.

To access the eGuide, visit the Web address shown below, or use your mobile device to scan the corresponding QR Code®.

https://education.ti.com/go/eguide/hub/EN

Apple®, Chrome®, Excel®, Google®, Firefox®, Internet Explorer®, Mac®, Microsoft®, Mozilla®, Safari®, and Windows® are registered trademarks of their respective owners.

QR Code® is a registered trademark of DENSO WAVE INCORPORATED.

Select images were created with Fritzing.

Contents

TI-Innovator™ Hub Commands version 1.1

Use the Hub menus to create or edit a program. They can save you time building commands and help you with correct command spelling and syntax.

Note: To build a command from the Hub menu, you need to know:

- The unique name of the component that you are addressing, such as "SOUND" for the on-board speaker.
- The command parameters that apply to the component, such as sound frequency and duration. Some parameters are optional, and you might need to know the value range of a parameter.

Understanding Syntax

- Capitalized words are keywords
- Lower case words are placeholders for numbers
- Commands within brackets are optional parameters

For example in: SET LIGHT ON [[BLINK|TOGGLE] frequency] [[TIME] seconds], "frequency" is entered as "**1**" and "seconds" is entered as "**10**".

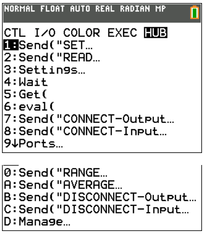
Send(“SET LIGHT **1** BLINK 2 TIME **10**”)

NOTE: The commands listed below are for the TI-84 Plus CE Hub Menu. If you are using TI-Nspire™ CX technology the parentheses are omitted. In addition, you will notice some other minor differences in the commands such as “**Endfor**” instead of “**End**” with the TI-Nspire™ CX technology. Screenshots are provided for reference.

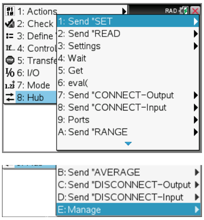
HUB Menus

- Send("SET...
- Send("READ...
- Settings
- Wait
- Get(
- eval(
- Send("CONNECT-Output...
- Send("CONNECT-Input...
- Ports...
- Send("RANGE...
- Send("AVERAGE...
- Send("DISCONNECT-Output...
- Send("DISCONNECT-Input...
- Manage...

TI-84 Plus CE



TI-Nspire™ CX

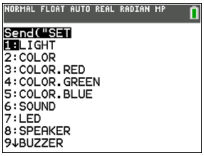


Send("SET…

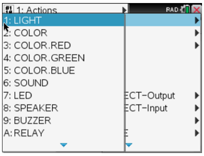
- SET
  - LIGHT
  - COLOR
  - COLOR.RED
  - COLOR.GREEN
  - COLOR.BLUE
  - SOUND
  - LED
  - SPEAKER
  - BUZZER
  - RELAY
  - SERVO
  - DCMOTOR
  - VIB.MOTOR
  - SQUAREWARE
  - RGB
  - ANALOG.OUT
  - DIGITAL.OUT
  - AVERAGING

Additional **Set** Commands

TI-84 Plus CE



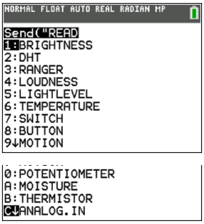
TI-Nspire™ CX



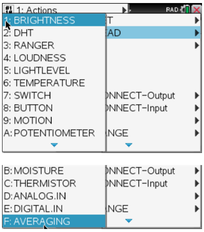
Send("READ…

- READ
  - BRIGHTNESS
  - DHT
  - RANGER
  - LOUDNESS
  - LIGHTLEVEL
  - TEMPERATURE
  - SWITCH
  - BUTTON
  - MOTION
  - POTENTIOMETER
  - MOISTURE
  - THERMISTOR

TI-84 Plus CE



TI-Nspire™ CX



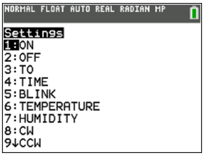
- ANALOG.IN
- DIGITAL.IN
- AVERAGING

Additional **READ** Commands

Settings…

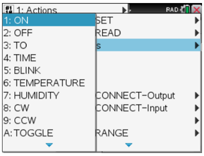
- Settings
  - ON
  - OFF
  - TO
  - TIME
  - BLINK
  - TEMPERATURE
  - HUMIDITY
  - CW
  - CCW
  - TOGGLE

TI-84 Plus CE



```
NORMAL FLOAT AUTO REAL RADIAN MP
Settings
1:ON
2:OFF
3:TO
4:TIME
5:BLINK
6:TEMPERATURE
7:HUMIDITY
8:CW
9↓CCW
```

TI-Nspire™ CX

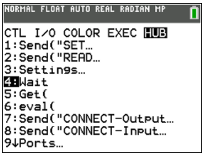


```
1: Actions                      RAD
1: ON          SET
2: OFF         READ
3: TO          s
4: TIME
5: BLINK
6: TEMPERATURE
7: HUMIDITY    CONNECT–Output
8: CW          CONNECT–Input
9: CCW
A: TOGGLE      RANGE
```

Wait

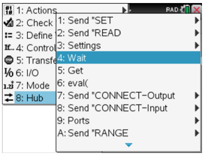
- Wait

TI-84 Plus CE



```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
1:Send("SET…
2:Send("READ…
3:Settings…
4:Wait
5:Get(
6:eval(
7:Send("CONNECT-Output…
8:Send("CONNECT-Input…
9↓Ports…
```

TI-Nspire™ CX

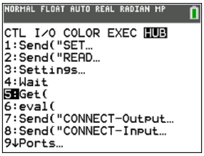


```
1: Actions                                RAD
2: Check    1: Send "SET
3: Define   2: Send "READ
4: Control  3: Settings
5: Transfe  4: Wait
6: I/O      5: Get
7: Mode     6: eval(
8: Hub      7: Send "CONNECT–Output
            8: Send "CONNECT–Input
            9: Ports
            A: Send "RANGE
```

Get(

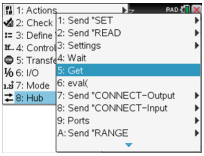
- Get(

TI-84 Plus CE



```
NORMAL FLOAT AUTO REAL RADIAN MP
CTL I/O COLOR EXEC HUB
1:Send("SET…
2:Send("READ…
3:Settings…
4:Wait
5:Get(
6:eval(
7:Send("CONNECT-Output…
8:Send("CONNECT-Input…
9↓Ports…
```

TI-Nspire™ CX



```
1: Actions                                RAD
2: Check    1: Send "SET
3: Define   2: Send "READ
4: Control  3: Settings
5: Transfe  4: Wait
6: I/O      5: Get
7: Mode     6: eval(
8: Hub      7: Send "CONNECT–Output
            8: Send "CONNECT–Input
            9: Ports
            A: Send "RANGE
```

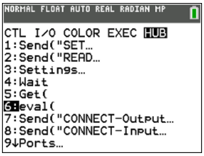
---

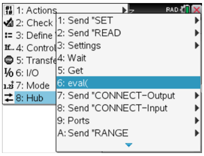
eval(

- eval(

TI-84 Plus CE

TI-Nspire™ CX

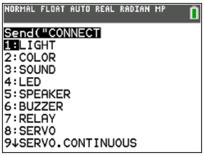


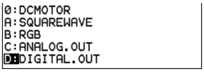


Send("CONNECT-Output...

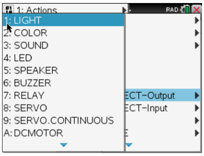
- CONNECT-Output
  - LIGHT
  - COLOR
  - SOUND
  - LED
  - SPEAKER
  - BUZZER
  - RELAY
  - SERVO
  - SERVO.CONTINUOUS
  - DCMOTOR
  - SQUAREWAVE
  - RGB
  - ANALOG.OUT
  - DIGITAL.OUT

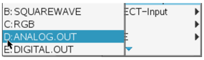
TI-84 Plus CE





TI-Nspire™ CX

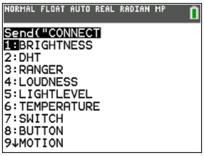


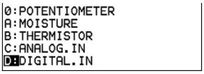


Send("CONNECT-Input...

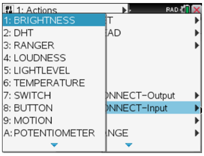
- CONNECT-Input
  - BRIGHTNESS
  - DHT
  - RANGER
  - LOUDNESS
  - LIGHTLEVEL
  - TEMPERATURE
  - SWITCH
  - BUTTON
  - MOTION
  - POTENTIOMETER
  - MOISTURE
  - THERMISTOR
  - ANALOG.IN

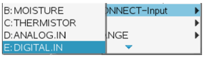
TI-84 Plus CE





TI-Nspire™ CX



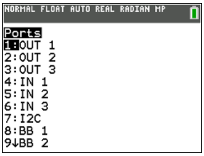


– DIGITAL.IN

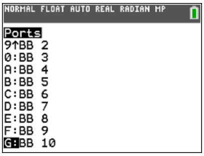
Ports…

- Ports
  - OUT 1
  - OUT 2
  - OUT 3
  - IN 1
  - IN 2
  - IN: 3
  - I2C
  - BB 1
  - BB 2
  - BB 3
  - BB 4
  - BB 5
  - BB 6
  - BB 7
  - BB 8
  - BB 9
  - BB 10

TI-84 Plus CE

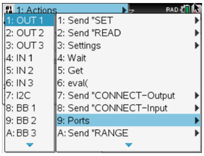


```
NORMAL FLOAT AUTO REAL RADIAN MP
Ports
1:OUT 1
2:OUT 2
3:OUT 3
4:IN 1
5:IN 2
6:IN 3
7:I2C
8:BB 1
9↓BB 2
```

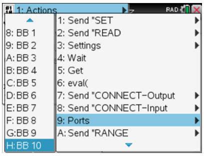


```
NORMAL FLOAT AUTO REAL RADIAN MP
Ports
9↑BB 2
0:BB 3
A:BB 4
B:BB 5
C:BB 6
D:BB 7
E:BB 8
F:BB 9
G:BB 10
```

TI-Nspire™ CX



```
1: Actions                          RAD
1: OUT 1   1: Send "SET
2: OUT 2   2: Send "READ
3: OUT 3   3: Settings
4: IN 1    4: Wait
5: IN 2    5: Get
6: IN 3    6: eval(
7: I2C     7: Send "CONNECT−Output
8: BB 1    8: Send "CONNECT−Input
9: BB 2    9: Ports
A: BB 3    A: Send "RANGE
```

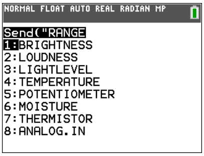


```
1: Actions                          RAD
           1: Send "SET
8: BB 1    2: Send "READ
9: BB 2    3: Settings
A: BB 3    4: Wait
B: BB 4    5: Get
C: BB 5    6: eval(
D: BB 6    7: Send "CONNECT−Output
E: BB 7    8: Send "CONNECT−Input
F: BB 8    9: Ports
G:BB 9     A: Send "RANGE
H: BB 10
```

Send("RANGE…

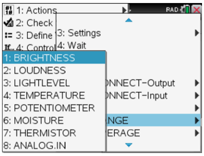
- RANGE
  - BRIGHTNESS
  - LOUDNESS
  - LIGHTLEVEL
  - TEMPERATURE
  - POTENTIOMETER
  - MOISTURE
  - THERMISTOR
  - ANALOG.IN

TI-84 Plus CE



```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("RANGE
1:BRIGHTNESS
2:LOUDNESS
3:LIGHTLEVEL
4:TEMPERATURE
5:POTENTIOMETER
6:MOISTURE
7:THERMISTOR
8:ANALOG.IN
```

TI-Nspire™ CX



```
1: Actions                          RAD
2: Check
3: Define  3: Settings
4: Control 4: Wait
1: BRIGHTNESS
2: LOUDNESS
3: LIGHTLEVEL         NNECT−Output
4: TEMPERATURE        NNECT−Input
5: POTENTIOMETER
6: MOISTURE           NGE
7: THERMISTOR         ERAGE
8: ANALOG.IN
```

Send("AVERAGE…

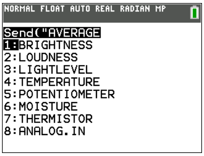
- AVERAGE
  - BRIGHTNESS
  - LOUDNESS

TI-84 Plus CE

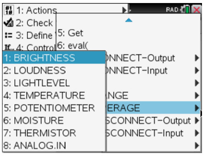
TI-Nspire™ CX

- LIGHTLEVEL
  - TEMPERATURE
  - POTENTIOMETER
  - MOISTURE
  - THERMISTOR
  - ANALOG.IN

Additional **AVERAGE** Commands



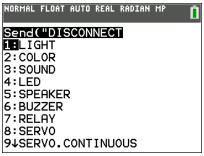
```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("AVERAGE
1:BRIGHTNESS
2:LOUDNESS
3:LIGHTLEVEL
4:TEMPERATURE
5:POTENTIOMETER
6:MOISTURE
7:THERMISTOR
8:ANALOG.IN
```



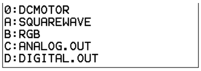
Send("DISCONNECT-Output...

- DISCONNECT-Output...
  - LIGHT
  - COLOR
  - SOUND
  - LED
  - SPEAKER
  - BUZZER
  - RELAY
  - SERVO
  - SERVO.CONTINUOUS
  - DCMOTOR
  - SQUAREWAVE
  - RGB
  - ANALOG.OUT
  - DIGITAL.OUT

TI-84 Plus CE

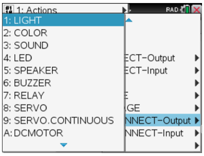


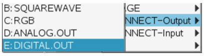
```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("DISCONNECT
1:LIGHT
2:COLOR
3:SOUND
4:LED
5:SPEAKER
6:BUZZER
7:RELAY
8:SERVO
9↓SERVO.CONTINUOUS
```



```
0:DCMOTOR
A:SQUAREWAVE
B:RGB
C:ANALOG.OUT
D:DIGITAL.OUT
```

TI-Nspire™ CX

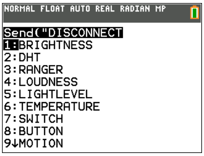




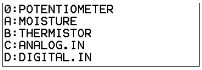
Send("DISCONNECT-INPUT...

- DISCONNECT-Input...
  - BRIGHTNESS
  - DHT
  - RANGER
  - LOUDNESS
  - LIGHTLEVEL
  - TEMPERATURE
  - SWITCH
  - BUTTON
  - MOTION
  - POTENTIOMETER

TI-84 Plus CE

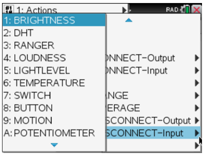


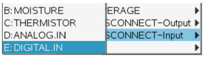
```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("DISCONNECT
1:BRIGHTNESS
2:DHT
3:RANGER
4:LOUDNESS
5:LIGHTLEVEL
6:TEMPERATURE
7:SWITCH
8:BUTTON
9↓MOTION
```



```
0:POTENTIOMETER
A:MOISTURE
B:THERMISTOR
C:ANALOG.IN
D:DIGITAL.IN
```

TI-Nspire™ CX



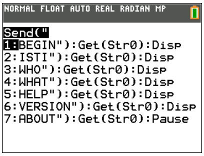


- MOISTURE
- THERMISTOR
- ANALOG.IN
- DIGITAL.IN

MANAGE

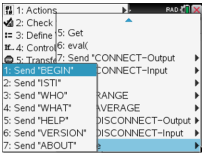
- MANAGE
  - BEGIN
  - ISTI
  - WHO
  - WHAT
  - HELP
  - VERSION
  - ABOUT

TI-84 Plus CE



```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("
1:BEGIN"):Get(Str0):Disp
2:ISTI"):Get(Str0):Disp
3:WHO"):Get(Str0):Disp
4:WHAT"):Get(Str0):Disp
5:HELP"):Get(Str0):Disp
6:VERSION"):Get(Str0):Disp
7:ABOUT"):Get(Str0):Pause
```

TI-Nspire™ CX



Additional Supported Commands Not Found in the Hub Menu

- Additional **SET** Commands
  - FORMAT ERROR STRING/NUMBER
  - FORMAT ERROR NOTE/QUIET
  - FLOW [TO] ON/OFF
  - OUT1/2/3 [TO]

- Additional **READ** Commands
  - ANALOG.OUT
  - BUZZER
  - COLOR
    - RED
    - GREEN
    - BLUE
  - DCMOTOR i
  - DIGITAL.OUT i
  - FORMAT
  - FLOW
  - IN1/IN2/IN3
  - LAST ERROR
  - LED i
  - LIGHT

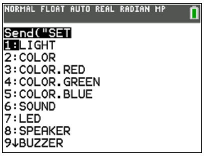
SET

The **SET** command is used to generate outputs on pins or ports, or control output devices such as **LED**s, Servo motors, speaker tones, or other output operations. It is also used to control a variety of system settings. These include formatting of error information, and communications flow control. **SET** does NOT generate any response that requires reading. The success or failure of a **SET** command may be determined by sending a **READ LAST ERROR** command and obtaining the response to that command. The sensors, controls, and settings that **SET** can operate against are in the following table.

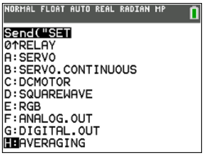
SET something'

Command:	**SET**
Command Syntax:	**SET**
Range:	
Describe:	Used to set options, or output states, or provide information used to control an external actuator or output device, such as turning on a **RELAY**.
Result:	
Type or Addressable Component:	

TI-84 Plus CE

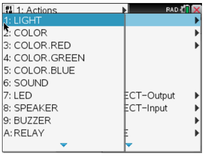


```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("SET
1:LIGHT
2:COLOR
3:COLOR.RED
4:COLOR.GREEN
5:COLOR.BLUE
6:SOUND
7:LED
8:SPEAKER
9↓BUZZER
```

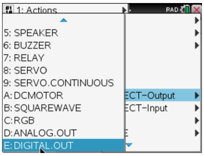


```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("SET
0↑RELAY
A:SERVO
B:SERVO.CONTINUOUS
C:DCMOTOR
D:SQUAREWAVE
E:RGB
F:ANALOG.OUT
G:DIGITAL.OUT
H:AVERAGING
```

TI-Nspire™ CX



```
1: Actions                RAD
1: LIGHT
2: COLOR
3: COLOR.RED
4: COLOR.GREEN
5: COLOR.BLUE
6: SOUND
7: LED                    ECT−Output
8: SPEAKER                ECT−Input
9: BUZZER
A: RELAY                  E
```



```
1: Actions                RAD
5: SPEAKER
6: BUZZER
7: RELAY
8: SERVO
9: SERVO.CONTINUOUS
A: DCMOTOR                ECT−Output
B: SQUAREWAVE             ECT−Input
C:RGB
D:ANALOG.OUT              E
E: DIGITAL.OUT
```

LIGHT [TO] ON/OFF

Command:	LIGHT [TO] ON/OFF

Command Syntax:	**SET LIGHT ON [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
	SET LIGHT OFF
	- same as **LED**, but for on-board red **LED**.

Range:	

Describe:	Provides control over the on-board digital **RED LED**. Set optional blink frequency and duration.
	SET LIGHT ON [[BLINK|TOGGLE] frequency] [[TIME] seconds]
	SET LIGHT OFF

Result:	Turns on LIGHT. Turns off LIGHT
Type or Addressable Component:	Control

COLOR [TO] r g b [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	COLOR [TO] r g b [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command Syntax:	**SET COLOR r g b [[BLINK|TOGGLE] frequency] [[TIME]seconds]**
	SET COLOR.component x [[BLINK|TOGGLE] frequency] [[TIME]seconds]

Range:	
Describe:	On-board **COLOR RGB LED** with sub-components **.RED**, **.GREEN**, **.BLUE**. Can have a blink frequency, and blink time for entire item, or for each component individually, as well as PWM levels given individually, or at one time.
Result:	Where r g b is r-value g-value b-value respectively, or operators from ON/OFF/UP/DOWN/STOP.
Type or Addressable Component:	Control

See Also:

COLOR.RED [TO] r [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	**COLOR.RED [TO] r [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Command Syntax:	**Send("SET COLOR.RED...")** **ON/OFF/UP/DOWN/STOP/0-255 (red element)** **[BLINK frequency] (in Hz)** **[TIME duration] (in secs)**
Range:	
Describe:	RED component of On-board **COLOR RGB LED**. Can have a blink frequency, and blink time for entire item, or for each component individually, as well as PWM levels given individually, or at one time.
Result:	Where r is red level, or operators from ON/OFF/UP/DOWN/STOP.
Type or Addressable Component:	Control

COLOR.GREEN [TO] g [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	**COLOR.GREEN [TO] g [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Command Syntax:	**SET COLOR.GREEN [TO] g [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Range:	
Describe:	GREEN component of On-board **COLOR RGB LED**. Can have a blink frequency, and blink time for entire item, or for each component individually, as well as PWM levels given individually, or at one time.
Result:	Where g is green level, or operators from ON/OFF/UP/DOWN/STOP.
Type or Addressable Component:	Control

COLOR.BLUE [TO] b [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	**COLOR.BLUE [TO] b [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Command Syntax:	**SET COLOR.BLUE [TO] b [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Range:	
Describe:	BLUE component of On-board **COLOR RGB LED**. Can have a blink frequency, and blink time for entire item, or for each component individually, as well as PWM levels given individually, or at one time.
Result:	Where b is blue level, or operators from ON/OFF/UP/DOWN/STOP.
Type or Addressable Component:	Control

SOUND [TO] frequency [[TIME] seconds]

Command:	**SOUND [TO] frequency [[TIME] seconds]**
Command Syntax:	**SET SOUND frequency [[TIME] seconds]**
Range:	
Describe:	**SOUND** is the on-board speaker and can generate a sound with a specified frequency. If not specified, sound will play for 1 second default. **SET SOUND frequency [[TIME] seconds]**
Result:	Play tone through on-board speaker.
Type or Addressable Component:	Control

SOUND OFF/0

Command:	**SOUND OFF/0**
Command Syntax:	**SET SOUND 0**

Command:	**SOUND OFF/0**
Range:	
Describe:	**SOUND** is the on-board speaker and can generate a sound with a specified frequency. If not specified, sound will play for 1 second default. **SET SOUND 0** – turns off sound on internal speaker immediately.
Result:	Stop playing sound.
Type or Addressable Component:	Control

LED i [TO] ON/OFF

Command:	**LED i [TO] ON/OFF**
Command Syntax:	**SET LED i ON/ OFF [[BLINK|TOGGLE] frequency] [[TIME] seconds]** – digital LED (on or off only)
Range:	
Describe:	Provides control over an external **LED** to set optional blink frequency and duration, as well as **PWM** capability if the associated pin connected to the **LED** supports it. **SET LED i ON [[BLINK|TOGGLE] frequency] [[TIME] seconds]** – digital LED (on or off only) **SET LED i OFF** – turns off LED (same as SET LED i 0).
Result:	Turns on LED. Turns off LED When connected to an Analog-PWM pin.
Type or Addressable Component:	Control

LED i [TO] 0-255

Command:	**LED i [TO] 0-255**
Command Syntax:	**SET LED i 0-255 [[BLINK|TOGGLE] frequency] [[TIME] seconds]** – analog LED (pwm duty cycyle)
Range:	
Describe:	Provides control over an external **LED** to set optional blink frequency and duration, as well as **PWM** capability if the associated pin connected to the **LED** supports it. **SET LED i 0-255 [[BLINK|TOGGLE] frequency] [[TIME] seconds]** – analog LED (pwm duty cycyle)
Result:	When connected to an Analog-PWM pin.
Type or Addressable Component:	Control

SPEAKER i [TO] frequency [[TIME] seconds]

Command:	**SPEAKER i [TO] frequency [[TIME] seconds]**
Command Syntax:	**SET SPEAKER i [TO] frequency [[TIME] seconds]**
Range:	
Describe:	Same as **SOUND** above, except sound is played on an external speaker attached to a digital output pin, available on any **IN/OUT** port, or the breadboard connector port. **Note**: On-board SOUND and external SPEAKER cannot be used concurrently.
Result:	Play tone with frequency given, optional duration in milliseconds, default = 1 second.
Type or Addressable Component:	Control

BUZZER i [TO] ON [TIME seconds]

Command:	**BUZZER i [TO] ON [TIME seconds]**
Command Syntax:	**SET BUZZER i ON [[TIME] seconds]**
Range:	
Describe:	Used to turn ON or OFF a tone on an active BUZZER for either 1 second default, or given length of time. **SET BUZZER i ON [[TIME] seconds]**
Result:	Sound tone on **ACTIVE** buzzer for 1 second, or specified duration in seconds.
Type or Addressable Component:	Control

BUZZER i [TO] OFF

Command:	**BUZZER i [TO] OFF**
Command Syntax:	**SET BUZZER i OFF**
Range:	
Describe:	Used to turn ON or OFF a tone on an active BUZZER for either 1 second default, or given length of time. **SET BUZZER i OFF**
Result:	Turn off tone on active buzzer.
Type or Addressable Component:	Control

RELAY i [TO] ON/OFF

Command:	**RELAY i [TO] On/Off**
Command Syntax:	**SET RELAY i ON/OFF /0/1 [[TIME] seconds]**.

Command:	**RELAY i [TO] On/Off**
Range:	Turns the specified **RELAY ON** or **OFF** for the given specified **TIME** in seconds.
Describe:	Control interface to an external RELAY control. **SET RELAY i ON/OFF/1/0 [[TIME] seconds]**
Result:	Turns RELAY on or off
Type or Addressable Component:	Control RELAY

SERVO i [TO] position

Command:	**SERVO i [TO] position**
Command Syntax:	**SET SERVO i [TO] position**.
Range:	
Describe:	Servo motor control interface. Servos can be either continuous or sweep style servos. Position = value from -90 to 90, ranged to -90 to 90) - used with **SWEEP SERVOS**
Result:	Sweep servos: position is a value from -90 to 90. Value 0 is same as specifying **ZERO**.
Type or Addressable Component:	Control

SERVO i [TO] STOP

Command:	**SERVO i [TO] STOP**
Command Syntax:	**SET SERVO i STOP**

Command:	**SERVO i [TO] STOP**
Range:	
Describe:	Servo motor control interface. Servos can be either continuous or sweep style servos. **Note**: Sweep style servos will stop automatically at the end of the sweep. **SET SERVO i STOP** – stops motion on servo
Result:	Halt any continuous servo operation in progress.
Type or Addressable Component:	Control

SERVO i [TO] ZERO

Command:	**SERVO i [TO] ZERO**
Command Syntax:	**SET SERVO i ZERO/position**
Range:	
Describe:	Set servo to zero position on sweep servo, or no motion on continuous servo.
Result:	Sweep servos: position is a value from -90 to 90. Value 0 is same as specifying **ZERO**.
Type or Addressable Component:	Control

SERVO i [TO] [CW/CCW] speed [[TIME] seconds]

Command:	**SERVO i [TO] [CW/CCW] speed [[TIME] seconds]**
Command Syntax:	**SET SERVO i CW/CCW speed [[TIME] seconds]**
Range:	
Describe:	Speed from -100 to 100, **CW/CCW** optional, if speed <0, **CCW**, else **CW**

Command:	**SERVO i [TO] [CW/CCW] speed [[TIME] seconds]**
	unless **CW/CCW** keyword is specified, TIME optional, in seconds, default=1 second (for continuous servo operation) (**CW/CCW** required if TIME/seconds NOT specified.)
Result:	Continuous servo where direction of rotation is specified, along with speed, from 0 (no motion) to 100 (fastest). Optional time parameter used to specify how long the servo should rotate in seconds.
Type or Addressable Component:	Control

DCMOTOR i [TO] frequency [duty [[TIME] seconds]]

Command:	**DCMOTOR i [TO] frequency [duty [[TIME] seconds]]**
Command Syntax:	**SET DCMOTOR i frequency [duty]**
Range:	
Describe:	Generates a specific frequency and duty cycle digital pulse to a motor. **SET DCMOTOR i frequency [duty]**
Result:	Generate a digital pulse at given frequency from 1 to 500 hz at 1-99 % duty cycle; shares number-space with SQUAREWAVE. duty=50% default, seconds=1.0 default.
Type or Addressable Component:	Control

DCMOTOR i OFF

Command:	**DCMOTOR i OFF**
Command Syntax:	**SET DCMOTOR i OFF**
Range:	
Describe:	Generates a specific frequency and duty cycle digital pulse to a motor.

Command:	**DCMOTOR i OFF**
	SET DCMOTOR i OFF
Result:	Stop motor.
Type or Addressable Component:	Control

VIB.MOTOR i [TO] PWM

Command:	**VIB.MOTOR i [TO] PWM**
Command Syntax:	**SET VIB.MOTOR i [TO] PWM**
Range:	PWM from 0 (none) and 255 (full on)
Describe:	Vibration motor control interface.
Result:	Vibrations : intensity is a value from 0 to 255.
Type or Addressable Component:	Control

VIB.MOTOR i [TO] OFF|STOP

Command:	**VIB.MOTOR i [TO] OFF|STOP**
Command Syntax:	**SET VIB.MOTOR i OFF|STOP**
Range:	
Describe:	Vibration motor control interface. **SET VIB.MOTOR i OFF|STOP** – stops motion on vibrations
Result:	Shut down vibration motor.
Type or Addressable Component:	Control

VIB.MOTOR i [TO] 0-255/UP/DOWN/ON/OFF [[BLINK|TOGGLE] freq] [[TIME] seconds]

Command:	**VIB.MOTOR i [TO] 0-255/UP/DOWN/ON/OFF [[BLINK|TOGGLE] freq] [[TIME] seconds]**
Command Syntax:	**SET VIB.MOTOR i 0-255/UP/DOWN/ON/OFF [[BLINK|TOGGLE] freq] [[TIME] seconds]**
Range:	PWM from 0 (none) and 255 (full on)
Describe:	Run vibration motor with numerous options
Result:	Run vibration motor with numerous options Optional time parameter used to specify how long the vibration should rotate in seconds.
Type or Addressable Component:	Control

SQUAREWAVE i [TO] frequency [duty [[TIME] seconds]]

Command:	**SQUAREWAVE i [TO] frequency [duty [[TIME] seconds]]**
Command Syntax:	**SET SQUAREWAVE i frequency [duty]**
Range:	
Describe:	**SQUAREWAVE** is used to generate a square wave form with a default duty cycle of 50% with frequencies from 0.1 Hz to 500 Hz. frequencies slower than 0.1 Hz are set to 0.1 Hz. frequencies above 500 Hz are set to 500 Hz. The optional duty cycle is a value from 1 to 99. **SET SQUAREWAVE i frequency [duty]**
Result:	Generate a digital squarewave from 1 to 500 hz at 1-99 duty cycle on up to 6 pins (i=1-4) duty=50% default, seconds=1.0 default.
Type or Addressable Component:	Control

SQUAREWAVE i OFF

Command:	**SQUAREWAVE i OFF**
Command Syntax:	**SET SQUAREWAVE i OFF** **frequency [duty]**
Range:	
Describe:	**SQUAREWAVE** is used to generate a square wave form with a default duty cycle of 50% with frequencies from 0.1 Hz to 500 Hz. frequencies slower than 0.1 Hz are set to 0.1 Hz. frequencies above 500 Hz are set to 500 Hz. The optional duty cycle is a value from 1 to 99. **SET SQUAREWAVE i OFF** – turn off squarewave generation
Result:	Stop generating squarewave output.
Type or Addressable Component:	Control

RGB i [TO] r g b [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	**RGB i [TO] r g b [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Command Syntax:	**SET RGB i r g b [[BLINK|TOGGLE] frequency] [[TIME]seconds]**
Range:	
Describe:	External **RGB LED** controls, with same options as available for the on-board COLOR object. Individual color components can be addressed with the same index value i by name, **RED i**, **GREEN i**, **BLUE i**.
Result:	Where r g b is r-value g-value b-value respectively, or operators from ON/OFF/STOP.
Type or Addressable Component:	Control

RED i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	**RED i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Command Syntax:	**SET.RED i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Range:	
Describe:	RED component of External RGB LED controls, with same options as available for the on-board COLOR object. Individual color components can be addressed with the same index value i by name, RED i, GREEN i, BLUE i.
Result:	
Type or Addressable Component:	Control

GREEN i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	**GREEN i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Command Syntax:	**SET.GREEN i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Range:	
Describe:	GREEN component of External RGB LED controls, with same options as available for the on-board COLOR object. Individual color components can be addressed with the same index value i by name, RED i, GREEN i, BLUE i.
Result:	
Type or Addressable Component:	Control

BLUE i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	**BLUE i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Command Syntax:	**SET.BLUE i [TO] ON/OFF/UP/DOWN/value [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Range:	
Describe:	BLUE component of External RGB LED controls, with same options as available for the on-board COLOR object. Individual color components can be addressed with the same index value i by name, RED i, GREEN i, BLUE i.
Result:	
Type or Addressable Component:	Control

ANALOG.OUT i [TO]

Command:	**ANALOG.OUT i [TO]**
Command Syntax:	**SET ANALOG.OUT i 0-255 [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Range:	
Describe:	Software (or hardware, if available) generated pulse-width modulation output at 490 Hz with the specified duty cycle between 0 (off) and 255 (on). The PWM output can be toggled at a frequency from 0.1 to 20.0 Hz for a given duration. If no duration is given, the PWM continues until stopped or turned off. **SET ANALOG.OUT i 0-255 [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Result:	Generate pwm value (hw or sw) on analog output object.
Type or Addressable Component:	Control

ANALOG.OUT i OFF|STOP

Command:	**ANALOG.OUT** i OFF|STOP

Command Syntax:	**SET ANALOG.OUT i OFF**
	SET ANALOG.OUT i STOP

Range:	

Describe:	Software (or hardware, if available) generated pulse-width modulation output at 490 Hz with the specified duty cycle between 0 (off) and 255 (on). The PWM output can be toggled at a frequency from 0.1 to 20.0 Hz for a given duration. If no duration is given, the PWM continues until stopped or turned off.
	SET ANALOG.OUT i OFF
	SET ANALOG.OUT i STOP

Result:	Turn off pwm on associated pin, including blinking, etc.
Type or Addressable Component:	Control

DIGITAL.OUT i [TO] ON/OFF/HIGH/LOW/[[BLINK|TOGGLE] frequency] [[TIME] seconds]

Command:	**DIGITAL.OUT i [TO] ON/OFF/HIGH/LOW/[[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Command Syntax:	**SET DIGITAL.OUT i [TO] ON/OFF/HIGH/LOW [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Range:	
Describe:	Used to generate output digital signal(s). **SET DIGITAL.OUT i ON/OFF/HIGH/LOW [[BLINK|TOGGLE] frequency] [[TIME] seconds]**
Result:	Digital.out operations.
Type or Addressable Component:	Control

DIGITAL.OUT i [TO] OUTPUT/CLOCK

Command:	**DIGITAL.OUT i [TO] OUTPUT/CLOCK**
Command Syntax:	**SET DIGITAL.OUT i [TO] OUTPUT/CLOCK**
Range:	
Describe:	Output or drive a clock pulse - digital.out other operations.
Result:	Output or drive a clock pulse - digital.out other operations.
Type or Addressable Component:	Control

DIGITAL.IN i [TO] INPUT/PULLUP/PULLDOWN

Command:	**DIGITAL.IN i [TO] INPUT/PULLUP/PULLDOWN**
Command Syntax:	**SET DIGITAL.IN i [TO] INPUT/PULLUP/PULLDOWN**
Range:	
Describe:	Used for Pulldown and/or pullup control for digital.in operations.
Result:	Pulldown and pullup control for digital.in operations.
Type or Addressable Component:	Control

AVERAGING [TO] n

Command:	**AVERAGING [TO] n** **Advanced user**
Command Syntax:	**AVERAGING.[TO] n**
Range:	
Describe:	Global setting for how many times we sample analog inputs when obtaining a reading from a sensor using analog input

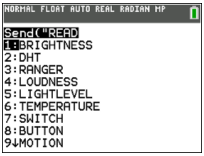
Command:	**AVERAGING [TO] n** **Advanced user**
	n - (global default)
Result:	Sample analog inputs '**n**' times, averaging results (default is 3 unless changed; sets "global" averaging value.)
Type or Addressable Component:	Setting Default if not set with this command is 3
Note:	Global averaging value can be individually overridden by sensor by using the **AVERAGING** command on an item.

READ

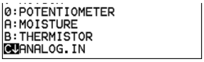
The **READ** command generates responses based on what is being requested.

Tells the Innovator to obtain data from the specified sensor, control, port, pin, or status information including the setup of the hub, such as flow control, error settings, etc. Must be followed by a Get() operation to receive the requested data.

TI-84 Plus CE

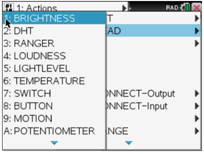


```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("READ
1:BRIGHTNESS
2:DHT
3:RANGER
4:LOUDNESS
5:LIGHTLEVEL
6:TEMPERATURE
7:SWITCH
8:BUTTON
9↓MOTION
```

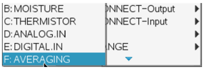


```
0:POTENTIOMETER
A:MOISTURE
B:THERMISTOR
C↓ANALOG.IN
```

TI-Nspire™ CX



```
1: Actions
1: BRIGHTNESS        T
2: DHT               AD
3: RANGER
4: LOUDNESS
5: LIGHTLEVEL
6: TEMPERATURE
7: SWITCH            NNECT−Output
8: BUTTON            NNECT−Input
9: MOTION
A: POTENTIOMETER     NGE
```



```
B: MOISTURE          NNECT−Output
C: THERMISTOR        NNECT−Input
D: ANALOG.IN
E: DIGITAL.IN        NGE
F: AVERAGING
```

BRIGHTNESS

Command:	**BRIGHTNESS**
Command Syntax:	**READ BRIGHTNESS**
Range:	
Describe:	Returns the current internal reading from the on-board ambient light sensor. Note the optional keywords of **RANGE** and **AVERAGE** can be appended to the command to return the current **RANGE** setting for the **BRIGHTNESS** sensor if set or the current **AVERAGE** value applied when reading the ADC to obtain the reading. **READ BRIGHTNESS**
Result:	Read on-board light sensor level.
Type or Addressable Component:	Control

BRIGHTNESS AVERAGE

Command:	**BRIGHTNESS AVERAGE** **Advanced user**
Command Syntax:	**READ BRIGHTNESS.AVERAGE**
Range:	
Describe:	Returns the current internal reading from the on-board ambient light sensor. Note the optional keywords of **RANGE** and **AVERAGE** can be appended to the command to return the current **RANGE** setting for the **BRIGHTNESS** sensor if set or the current **AVERAGE** value applied when reading the ADC to obtain the reading. **READ BRIGHTNESS AVERAGE**
Result:	Read on-board light sensor level.
Type or Addressable Component:	Control

BRIGHTNESS RANGE

Command:	**BRIGHTNESS RANGE** **Advanced user**
Command Syntax:	**READ BRIGHTNESS.RANGE**
Range:	
Describe:	Returns the current internal reading from the on-board ambient light sensor. Note the optional keywords of **RANGE** and **AVERAGE** can be appended to the command to return the current **RANGE** setting for the **BRIGHTNESS** sensor if set or the current **AVERAGE** value applied when reading the ADC to obtain the reading. **READ BRIGHTNESS RANGE**
Result:	Read on-board light sensor level.
Type or Addressable Component:	Control

DHT i

Command:	**DHT i**
Command Syntax:	**READ DHT i**
Range:	Temperature reading default is in Celsius Humidity reading from 0 to 100 %
Describe:	Returns a list consisting of the current temperature, humidity, type of sensor, and last cached read status. The temperature and humidity can be obtained by themselves by appending the TEMPERATURE or HUMIDITY keywords to the end of the command. The type of sensor is indicated by a 1 for a DHT11, and a 2 for DHT22 style sensors. The status values are: 1=OK, 2=Timeout, 3=Checksum/bad reading. **READ DHT i** – returns full cached information from last reading the DHT task obtained. **READ DHT i TEMPERATURE** – returns latest temperature reading. **READ DHT i HUMIDITY** – returns latest humidity reading.
Result:	Return list with current temperature in C, humidity in %, type(1=DHT11, 2=DHT22), and status (type/status only available in full list). Where the status = 1:OK, =2:Timeout, =3:Checksum.
Type or Addressable Component:	Sensor

DHT i TEMPERATURE

Command:	**DHT i TEMPERATURE**
Command Syntax:	**READ DHT i TEMPERATURE**
Range:	Temperature reading default is in Celsius Humidity reading from 0 to 100 %
Describe:	Returns a list consisting of the current temperature, humidity, type of sensor, and last cached read status. The temperature and humidity can be obtained by themselves by appending the TEMPERATURE or HUMIDITY keywords to the end of the command. The type of sensor is indicated by a 1 for a DHT11, and a 2 for DHT22 style sensors. The status values are:

Command:	**DHT i TEMPERATURE**
	1=OK, 2=Timeout, 3=Checksum/bad reading. **READ DHT i** – returns full cached information from last reading the DHT task obtained. **READ DHT i TEMPERATURE** – returns latest temperature reading. **READ DHT i HUMIDITY** – returns latest humidity reading.
Result:	Returns temperature component.
Type or Addressable Component:	Sensor

DHT i HUMIDITY

Command:	**DHT i HUMIDITY**
Command Syntax:	**READ DHT i HUMIDITY**
Range:	Temperature reading default is in Celsius Humidity reading from 0 to 100 %
Describe:	Returns a list consisting of the current temperature, humidity, type of sensor, and last cached read status. The temperature and humidity can be obtained by themselves by appending the TEMPERATURE or HUMIDITY keywords to the end of the command. The type of sensor is indicated by a 1 for a DHT11, and a 2 for DHT22 style sensors. The status values are: 1=OK, 2=Timeout, 3=Checksum/bad reading. **READ DHT i** – returns full cached information from last reading the DHT task obtained. **READ DHT i TEMPERATURE** – returns latest temperature reading. **READ DHT i HUMIDITY** – returns latest humidity reading.
Result:	Returns humidity component.
Type or Addressable Component:	Sensor

RANGER i

Command:	**RANGER i**
Command Syntax:	**READ RANGER i**
Range:	
Describe:	Return the current distance measurement from the specified ultrasonic ranging device; distance in meters. If no measurement is made due to the distance being too far; a value of 0 will be returned. Valid measurements are in +meters.
Result:	Read distance in centimeters meters from distance sensor.
Type or Addressable Component:	Sensor

LOUDNESS i

Command:	**LOUDNESS i**
Command Syntax:	**READ LOUDNESS i**
Range:	
Describe:	Return the current analog level reported by the sound loudness level sensor specified. Supports the **AVERAGE** and **RANGE** options. **READ LOUDNESS i** **READ LOUDNESS i AVERAGE** **READ LOUDNESS i RANGE**
Result:	Return level of sound detected by sound sensor.
Type or Addressable Component:	Sensor

LOUDNESS i AVERAGE

Command:	**LOUDNESS i** **Advanced user**
Command Syntax:	**READ LOUDNESS i AVERAGE**
Range:	
Describe:	Return the current analog level reported by the sound loudness level sensor specified. Supports the **AVERAGE** and **RANGE** options. **READ LOUDNESS i AVERAGE**
Result:	Return level of sound detected by sound sensor.
Type or Addressable Component:	Sensor

LOUDNESS i RANGE

Command:	**LOUDNESS i RANGE** **Advanced user**
Command Syntax:	**READ LOUDNESS i.RANGE**
Range:	
Describe:	Return the current analog level reported by the sound loudness level sensor specified. Supports the **AVERAGE** and **RANGE** options. **READ LOUDNESS i** **READ LOUDNESS i AVERAGE** **READ LOUDNESS i RANGE**
Result:	Return level of sound detected by sound sensor.
Type or Addressable Component:	Sensor

LIGHTLEVEL i

Command:	**LIGHTLEVEL i**
Command Syntax:	**READ LIGHTLEVEL i**
Range:	An integer value between 0 and 16383 (14 bit resolution)
Describe:	Returns the current **ADC** value from the specified external light sensor. External light sensors may be analog, or I2C (BH1750FVI I2C Light sensor). When an analog sensor is present, it is generally assumed to be a photodiode. Additionally, the light level sensor may have **AVERAGE** and/or **RANGE** values specified. These can be obtained by appending the **AVERAGE** or **RANGE** keywords to the **READ** command. **READ LIGHTLEVEL i** **READ LIGHTLEVEL i AVERAGE** **READ LIGHTLEVEL i RANGE**
Result:	Read analog value of light sensor (uses averaging), or I2C (value in **LUX** returned).
Type or Addressable Component:	Sensor

LIGHTLEVEL i AVERAGE

Command:	**LIGHTLEVEL i AVERAGE** **Advanced user**
Command Syntax:	**READ LIGHTLEVEL i AVERAGE**
Range:	An integer value between 0 and 16383 (14 bit resolution)
Describe:	Returns the current **ADC** value from the specified external light sensor. External light sensors may be analog, or I2C (BH1750FVI I2C Light sensor). When an analog sensor is present, it is generally assumed to be a photodiode. Additionally, the light level sensor may have **AVERAGE** and/or **RANGE** values specified. These can be obtained by appending the **AVERAGE** or **RANGE** keywords to the **READ** command. **READ LIGHTLEVEL i AVERAGE**
Result:	Read analog value of light sensor (uses averaging), or I2C (value in **LUX**

Command:	**LIGHTLEVEL i AVERAGE** **Advanced user**
	returned).
Type or Addressable Component:	Sensor

LIGHTLEVEL i RANGE

Command:	**LIGHTLEVEL i RANGE** **Advanced user**
Command Syntax:	**READ LIGHTLEVEL i RANGE**
Range:	An integer value between 0 and 16383 (14 bit resolution)
Describe:	Returns the current **ADC** value from the specified external light sensor. External light sensors may be analog, or I2C (BH1750FVI I2C Light sensor). When an analog sensor is present, it is generally assumed to be a photodiode. Additionally, the light level sensor may have **AVERAGE** and/or **RANGE** values specified. These can be obtained by appending the **AVERAGE** or **RANGE** keywords to the **READ** command. **READ LIGHTLEVEL i RANGE**
Result:	Read analog value of light sensor (uses averaging), or I2C (value in **LUX** returned).
Type or Addressable Component:	Sensor

TEMPERATURE i

Command:	**TEMPERATURE i**
Command Syntax:	**READ TEMPERATURE i**

Command:	**TEMPERATURE i**
Range:	Temperature reading default is in Celsius. The range depends on the specific temperature sensor being used. Humidity reading from 0 to 100 %
Describe:	Returns the current temperature reading from the associated temperature sensor. The temperature is given, by default, in Celsius. **READ TEMPERATURE i**
Result:	Return current temperature reading in Celsius.
Type or Addressable Component:	Sensor

TEMPERATURE i AVERAGE

Command:	**TEMPERATURE i AVERAGE** **Advanced user**
Command Syntax:	**READ TEMPERATURE i AVERAGE**
Range:	Temperature reading default is in Celsius. The range depends on the specific temperature sensor being used. Humidity reading from 0 to 100 %
Describe:	Returns the current temperature reading from the associated temperature sensor. The temperature is given, by default, in Celsius. **READ TEMPERATURE i AVERAGE**
Result:	Return current temperature reading in Celsius.
Type or Addressable Component:	Sensor

TEMPERATURE i CALIBRATION

Command:	TEMPERATURE i CALIBRATION Advanced user
Command Syntax:	**READ TEMPERATURE i CALIBRATION**
Range:	Temperature reading default is in Celsius. The range depends on the specific temperature sensor being used. Humidity reading from 0 to 100 %
Describe:	Returns the current temperature reading from the associated temperature sensor. The temperature is given, by default, in Celsius.
Result:	Returns list with current {c1,c2,c3,r} values used for connected analog temperature sensor.
Type or Addressable Component:	Sensor

SWITCH i

Command:	SWITCH i
Command Syntax:	**READ SWITCH i**
Range:	
Describe:	Returns the current state of the associated switch. If the switch is connected, a value of 1 is returned. Not connected returns a value of 0. If the switch was connected since the last reading, but is no longer connected, a value of 2 is returned. **READ SWITCH i**
Result:	Returns state of switch (same status as **BUTTON** object, 0=not pressed, 1=pressed, 2=was pressed).
Type or Addressable Component:	Sensor

BUTTON i

Command:	**BUTTON i**
Command Syntax:	**READ BUTTON i**
Range:	
Describe:	Reads the current cached state of the button. A return value of 0 = *not pressed*, 1 = *currently pressed*, 2 = *was pressed* and released since the last reading. **READ BUTTON i**
Result:	Read state of button/switch n - 0=not pressed, 1=pressed, 2=was pressed.
Type or Addressable Component:	Sensor

MOTION i

Command:	**MOTION i**
Command Syntax:	**READ MOTION i**
Range:	
Describe:	Return the current **PIR Motion sensor** information. **PIR Motion sensors** are digital in nature, so are treated similar to a button in that the value returned indicates motion presence or not. **0**=*no motion detected.* **1**=*motion detected.* **2**=*motion was detected.*
Result:	Read state of **PIR Motion detector** - 0=no motion, 1=motion, 2=motion was detected but none now.
Type or Addressable Component:	Sensor

POTENTIOMETER i

Command:	**POTENTIOMETER i**
Command Syntax:	**READ POTENTIOMETER i**
Range:	
Describe:	Read analog value of the potentiometer (linear or rotary). The optional **AVERAGE** and **RANGE** keywords can be appended to the command to obtain the current average count, or mapped range being used, if present, for the given potentiometer. **READ POTENTIOMETER i** **READ POTENTIOMETER i RANGE** **READ POTENTIOMETER i AVERAGE**
Result:	Read analog value of rotary encoder / potentiometer (uses averaging).
Type or Addressable Component:	Sensor

POTENTIOMETER i AVERAGE

Command:	**POTENTIOMETER i AVERAGE** **Advanced user**
Command Syntax:	**READ POTENTIOMETER i AVERAGE**
Range:	
Describe:	Read analog value of the potentiometer (linear or rotary). The optional **AVERAGE** and **RANGE** keywords can be appended to the command to obtain the current average count, or mapped range being used, if present, for the given potentiometer. **READ POTENTIOMETER i AVERAGE**
Result:	Read analog value of rotary encoder / potentiometer (uses averaging).
Type or Addressable Component:	Sensor

POTENTIOMETER i RANGE

Command:	**POTENTIOMETER i RANGE** **Advanced user**
Command Syntax:	**READ POTENTIOMETER i RANGE**
Range:	
Describe:	Read analog value of the potentiometer (linear or rotary). The optional **AVERAGE** and **RANGE** keywords can be appended to the command to obtain the current average count, or mapped range being used, if present, for the given potentiometer. **READ POTENTIOMETER i RANGE**
Result:	Read analog value of rotary encoder / potentiometer (uses averaging).
Type or Addressable Component:	Sensor

MOISTURE i

Command:	**MOISTURE i**
Command Syntax:	**READ MOISTURE i**
Range:	An integer value between 0 and 16383 (14 bit resolution)
Describe:	Return the current analog level reported by the moisture sensor specified. Supports the **AVERAGE** and **RANGE** options. **READ MOISTURE i** **READ MOISTURE i AVERAGE** **READ MOISTURE i RANGE**
Result:	Read analog value of moisture sensor (uses averaging).
Type or Addressable Component:	Sensor

MOISTURE i AVERAGE

Command:	**MOISTURE i AVERAGE** **Advanced user**
Command Syntax:	**READ MOISTURE i AVERAGE**
Range:	
Describe:	Return the current analog level reported by the moisture sensor specified. Supports the **AVERAGE** and **RANGE** options. **READ MOISTURE i AVERAGE**
Result:	Read analog value of moisture sensor (uses averaging).
Type or Addressable Component:	Sensor

MOISTURE i RANGE

Command:	**MOISTURE i RANGE**
Command Syntax:	**READ MOISTURE i RANGE**
Range:	
Describe:	Return the current analog level reported by the moisture sensor specified. Supports the **AVERAGE** and **RANGE** options. **READ MOISTURE i RANGE**
Result:	Read analog value of moisture sensor (uses averaging).
Type or Addressable Component:	Sensor

THERMISTOR i

Command:	**THERMISTOR i**
Command Syntax:	**READ THERMISTOR i**
Range:	
Describe:	Returns the current temperature reading from the associated thermistor sensor. Temperature is returned in Celsius.
Result:	Return current thermistor temperature in Celsius.
Type or Addressable Component:	Sensor

THERMISTOR i AVERAGE

Command:	**THERMISTOR i AVERAGE** **Advanced user**
Command Syntax:	**READ THERMISTOR i AVERAGE**
Range:	
Describe:	Returns the current temperature reading from the associated thermistor sensor. Temperature is returned in Celsius.
Result:	Return current thermistor temperature in Celsius.
Type or Addressable Component:	Sensor

THERMISTOR i CALIBRATION

Command:	**THERMISTOR i CALIBRATION** **Advanced user**
Command	**READ THERMISTOR i CALIBRATION**

Command:	**THERMISTOR i CALIBRATION** **Advanced user**
Syntax:	
Range:	
Describe:	Returns the current temperature reading from the associated thermistor sensor. Temperature is returned in Celsius.
Result:	Returns list with current {c1,c2,c3,r} values used for connected thermistor.
Type or Addressable Component:	Sensor

ANALOG.IN i

Command:	**ANALOG.IN i**
Command Syntax:	**READ.ANALOG.IN i**
Range:	
Describe:	Generic analog input sensor. **READ ANALOG.IN i** – will return the ADC reading on the analog input associated with the object.
Result:	Reads generic **ANALOG.IN** input object
Type or Addressable Component:	Sensor

ANALOG.IN i AVERAGE

Command:	**ANALOG.IN i AVERAGE** **Advanced user**
Command Syntax:	**READ.ANALOG.IN i AVERAGE**

Command:	**ANALOG.IN i AVERAGE** **Advanced user**
Range:	
Describe:	**READ ANALOG IN i AVERAGE** – gets the current averaging value for the object.
Result:	Reads generic **ANALOG.IN** input object
Type or Addressable Component:	Sensor

ANALOG.IN i RANGE

Command:	**ANALOG.IN i RANGE** **Advanced user**
Command Syntax:	**READ.ANALOG.IN i RANGE**
Range:	
Describe:	**READ ANALOG IN i RANGE** – returns the upper and lower range values associated with the object if specified, or error otherwise
Result:	Reads generic **ANALOG.IN** input object
Type or Addressable Component:	Sensor

ANALOG.OUT i

Command:	**ANALOG.OUT i**
Command Syntax:	**READ ANALOG.OUT i**
Range:	
Describe:	Returns current PWM duty cycle if the output is on, or 0 if not on.
Result:	Reads current PWM duty cycle on pin, 0 if none.

Command:	**ANALOG.OUT i**
Type or Addressable Component:	Control

DIGITAL.IN i

Command:	**DIGITAL.IN i**
Command Syntax:	**READ DIGITAL.IN i**
Range:	
Describe:	Returns the current state of the digital pin connected to the DIGITAL object, or the cached state of the digital output value last SET to the object.
Result:	Return 0 (low), 1 (high).
Type or Addressable Component:	Control/Sensor

AVERAGING

Command:	**AVERAGING** **Advanced user**
Command Syntax:	**READ AVERAGING**
Range:	
Describe:	Returns the current global setting for the analog averaging default value.
Result:	Return current oversampling/averaging count for sampling analog inputs (this is the GLOBAL default value currently in use).
Type or Addressable Component:	Setting

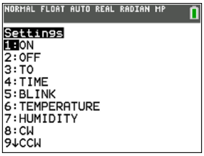
Settings

Settings menu contains operations to set the state of digital and analog pin operations such as the **LED** in the TI-Innovator™ Hub or a connected servo motor movement to states such as ON, OFF, CW (clockwise), and CCW (counterclockwise).

- 1: ON
- 2: OFF
- 3: TO
- 4: TIME
- 5: BLINK
- 6: TEMPERATURE
- 7: HUMIDITY
- 8: CW
- 9: CCW
- 0: TOGGLE

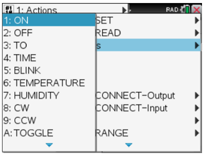
TI-84 Plus CE

See : TI-84 Plus CE Programming



TI-Nspire™ CX

See: TI-Nspire™ CX Programming



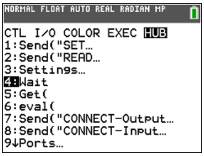
Wait

Wait suspends execution of a program for a given time. Maximum time is 100 seconds. During the wait time, the busy indicator is on in the top-right corner of the screen.

Wait may be used in TI-Innovator™ Hub programs to allow time for sensor or control communications prior to the program executing the next command line.

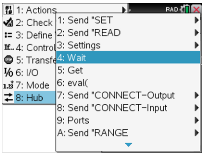
TI-84 Plus CE

See: TI-84 Plus CE Programming



TI-Nspire™ CX

See: TI-Nspire™ CX Programming



Wait

Command:	**Wait**
Command Syntax:	Wait *timeInSeconds* Suspends execution for a period of *timeInSeconds* seconds.

Command:	Wait
Range	0 through 100
Describe:	**Wait** may be used in TI-Innovator™ Hub programs to allow time for sensor or control communications prior to the program executing the next command line. **Wait** is particularly useful in a program that needs a brief delay to allow requested data to become available. The argument *timeInSeconds* must be an expression that simplifies to a decimal value in the range 0 through 100. The command rounds this value up to the nearest 0.1 seconds. **Note**: You can use the **Wait** command within a user-defined program but not within a function.
Result:	**Wait** suspends execution of a program for a given time. Maximum time is 100 seconds. During the wait time, the busy indicator is on in the top-right corner of the screen.
Type or Addressable Component:	Not Applicable

Get(

Get(Retrieves a value from a connected TI-Innovator™ Hub and stores the data to a variable on the receiving CE calculator.

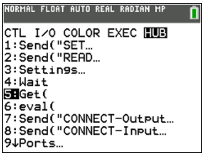
TI-84 Plus CE

Get(command definition is specific to the TI-8x calculator and the cable connection via DBus or USB. The CE calculator is USB connectivity only and here, **Get(** is designed for communication with the TI-Innovator™ Hub.

TI-Nspire™ CX

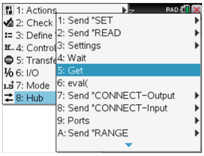
TI-84 Plus CE

See: TI-84 Plus CE Programming



TI-Nspire™ CX

See: TI-Nspire™ CX Programming



Get(

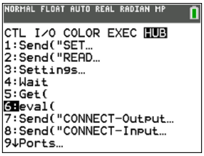
Command:	**Get(**
Command Syntax:	**TI-84 Plus CE**: **Get(***variable***)** **TI-Nspire CX** platform: **Get** [*promptString*,] *var*[, *statusVar*] **Get** [*promptString*,] *func*(*arg1, ...argn*) [, *statusVar*]
Range	
Describe:	
Result:	Programming command: Retrieves a value from a connected TI-Innovator™ Hub and assigns the value to variable *var*. The value must be requested: • In advance, through a **Send "READ ..."** command. — or — • By embedding a **"READ ..."** request as the optional *promptString* argument. This method lets you use a single command to request the value and retrieve it. (**TI-Nspire™ CX** platform only). Implicit simplification takes place. For example, a received string of "123" is interpreted as a numeric value. **The information below applies only on the TI-Nspire CX platform**: To preserve the string, use **GetStr** instead of **Get**. If you include the optional argument *statusVar*, it is assigned a value based on the success of the operation. A value of zero means that no data was received. In the second syntax, the *func()* argument allows a program to store the received string as a function definition. This syntax operates as if the program executed the command: Define *func(arg1, ...argn) = received string* The program can then use the defined function *func()*. **Note**: You can use the **Get** command within a user-defined program but not within a function.
Type or Addressable Component:	All input devices.

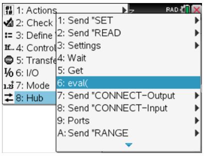
eval(

The software evaluates expression *Expr* and replaces the **eval()** statement with the result as a character string.

The argument *Expr* must simplify to a real number.

	TI-84 Plus CE	**TI-Nspire™ CX**
	See: TI-84 Plus CE Programming	See: TI-Nspire™ CX Programming





eval(

Command:	**eval(**
Command Syntax:	**eval(***Expr***) ⇒ string**
Range	
Describe:	The software evaluates expression *Expr* and replaces the **eval()** statement with the result as a character string. The argument *Expr* must simplify to a real number. **TI-84 Plus CE**: **eval()** can be used as a standalone command outside a TI-Innovator™ Hub command. **TI-Nspire™ CX** platform: **eval()** is valid only in the TI-Innovator™ Hub Command argument of programming commands **Get**, **GetStr**, and **Send**.
Result:	**TI-84 Plus CE**: For debugging purposes, using the command line Disp Ans immediately after a command line using Send(displays the complete string being sent. **TI-Nspire™ CX** platform: Although **eval()** does not display its result, you can view the resulting Hub command string after executing the command by inspecting any of the following special variables. *iostr.SendAns* *iostr.GetAns* *iostr.GetStrAns*
Type or	Not Applicable

Command:	**eval(**
Addressable Component:	

CONNECT - Output

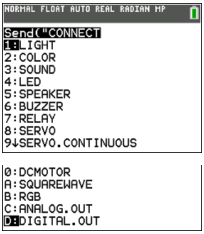
CONNECT associates a given control or sensor with a pin or port on the TI-Innovator. If the specified control or sensor is currently in use, an error will be generated. If the pin or port specified in the **CONNECT** command is currently in use, an error will be generated.

The **CONNECT** command does not generate an active response, but a variety of errors may occur during a connection attempt, such as pin-in-use, unsupported, invalid options, bad options, etc.

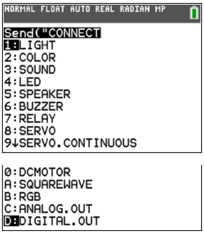
CONNECT 'something i' [TO] IN1/IN2/IN3/OUT1/OUT2/OUT3/BB1

Command:	**CONNECT**
Command Syntax:	**CONNECT**
Range:	
Describe:	Associates a sensor or control with a given port or pin(s). Places the respective pin(s) in use
Result:	
Type or Addressable Component:	

TI-84 Plus CE



TI-Nspire™ CX



LIGHT

Command:	**LIGHT**
Command Syntax:	**CONNECT LIGHT**
Range:	

Command:	**LIGHT**
Describe:	This command is not needed for typical use since the on-board LIGHT (i.e. RED LED) is automatically connected. Re-connect a previously disconnected on-board RED LED. The LIGHT is always connected when the system is reset, or powered-on, or the BEGIN command is used to restore system state. No pin number is required. **CONNECT LIGHT**
Result:	Connects on-board digital LED (red) to known fixed pin. Digital only.
Type or Addressable Component:	Control

COLOR

Command:	**COLOR**
Command Syntax:	**CONNECT COLOR**
Range:	
Describe:	This command is not needed for typical use since the on-board COLOR LED is automatically connected. (Re-)connect the internal **RGB LED**. No pins are required for this command to operate as the internal pins are known. This sensor is automatically connected when the TI-Innovator is initially powered, and when the **BEGIN** command is used. When disconnected, two **PWM** signals are freed for external use by other pins. **CONNECT COLOR**
Result:	Connects on-board **RGB LED** to fixed pins on board. Uses 3 **PWM**s.
Type or Addressable Component:	Control

SOUND

Command:	**SOUND**
Command Syntax:	**CONNECT SOUND**

Command:	SOUND
Range:	
Describe:	This command is not needed for typical use since the on-board object SOUND is automatically connected. Re-connect the on-board speaker for sound generation. No pin needed as it uses known, fixed pin for signal. **CONNECT SOUND**
Result:	Connects on-board speaker to fixed output digital pin.
Type or Addressable Component:	Control

LED i [TO] OUT n/BB n

Command:	LED i [TO] OUT n/BB n
Command Syntax:	**CONNECT LED i [TO] OUT n/BB n**
Range:	
Describe:	This object provides the ability to connect external **LED** objects. The **LED** object is either connected to a **PWM** function (if available, and the pin connecting to supports it), or a digital output pin which will be driven at 50% duty cycle; or the specified blink rate if one is specified in the **SET** operation. **CONNECT LED 1i [TO] BB3** **CONNECT LED 2i [TO] OUT1**
Result:	LED connected to specific port.
Type or Addressable Component:	Control

SPEAKER i [TO] OUT n/BB n

Command:	SPEAKER i [TO] OUT n/BB n
Command Syntax:	**CONNECT SPEAKER i [TO] OUT n/BB n**

Command:	SPEAKER i [TO] OUT n/BB n
Range:	
Describe:	Connect an external speaker for sound generation. Requires a digital output pin. **CONNECT SPEAKER 1 [TO] OUT 1** **CONNECT SPEAKER i [TO] BB 3**
Result:	Connect a speaker to a digital output port or pin.
Type or Addressable Component:	Control

BUZZER i [TO] OUT n/BB n

Command:	BUZZER i [TO] OUT n/BB n
Command Syntax:	**CONNECT BUZZER i [TO] OUT n/BB n**
Range:	
Describe:	Connect an external active buzzer to an output digital pin. Active buzzers play a tone when their signal is set high/on, and stop the tone when the signal is dropped to ground. For piezo or passive buzzers, use the **SPEAKER** object type to allow generation of multiple tones. **CONNECT BUZZER i [TO] OUT1**
Result:	**ACTIVE** buzzers connect to a digital pin.
Type or Addressable Component:	Control

RELAY i [TO] OUT n/BB n

Command:	RELAY i [TO] OUT n/BB n
Command Syntax:	**CONNECT RELAY i [TO] OUT n/BB n**
Range:	
Describe:	With external power required, connect a relay module to a given control

Command:	**RELAY i [TO] OUT n/BB n**
	signal pin. Since the control is digital, as long as external power is present, any pin may be used. **CONNECT RELAY 1 [TO] BB 3** **CONNECT RELAY 1 [TO] OUT 2**
Result:	Relays.
Type or Addressable Component:	Control

SERVO i [TO] OUT n

Command:	**SERVO i [TO] OUT n**
Command Syntax:	**CONNECT SERVO i [TO] OUT n**
Range:	
Describe:	Used to connect either a normal sweep servo motor, or a continuous servo motor. External power must be provided before attempting to connect the servo. **CONNECT SERVO 1 [TO] OUT 1**
Result:	Servo motor is connected to port.
Type or Addressable Component:	Control

SERVO.CONTINUOUS i [TO] BB 6

Command:	**SERVO.CONTINUOUS i [TO] BB 6**
Command Syntax:	**CONNECT SERVO.CONTINUOUS i [TO] BB 6**
Range:	
Describe:	Used to connect either a normal sweep servo motor, or a continuous servo motor. External power must be provided before attempting to connect the servo. **CONNECT SERVO.CONTINUOUS i [TO] BB 6**

Command:	**SERVO.CONTINUOUS i [TO] BB 6**
Result:	Servo motor with -90 to 90 degree movement.
Type or Addressable Component:	Control

DCMOTOR i [TO] OUT n/BB n

Command:	**DCMOTOR i [TO] OUT n/BB n**
Command Syntax:	**CONNECT DCMOTOR i [TO] OUT n/BB n**
Range:	
Describe:	Connect an external **DC Motor** object. This object requires the presence of power on the external power connector to allow operation. These objects share the number-space with the **SQUAREWAVE** output objects and **ANALOG.OUT** objects. The associated pin is configured as a digital output signal. **CONNECT DCMOTOR i [TO] OUT1**
Result:	Connect **DCMOTOR** to a digital output pin.
Type or Addressable Component:	Control

RGB i / COLOR [TO] BB r BB g BB b

Command:	**RGB i / COLOR [TO] BB r BB g BB b**
Command Syntax:	**CONNECT RGB i / COLOR [TO] BB r BB g BB b**
Range:	
Describe:	Connects an external **RGB LED** to three **PWM**-capable pins. If insufficient PWM pins are available for mapping to PWM function, an error will be given. To connect an external RGB, the on-board **RGB LED** should be **DISCONNECT**ed before the attempt to connect the external RGB is performed. **CONNECT RGB 1 [TO] BB8 BB9 BB10**

Command:	**RGB i / COLOR [TO] BB r BB g BB b**
Result:	Digital pins supporting PWM.
Type or Addressable Component:	Control

SQUAREWAVE i [TO] OUT n/BB n

Command:	**SQUAREWAVE i [TO] OUT n/BB n**
Command Syntax:	**CONNECT SQUAREWAVE i [TO] OUT n/BB n**
Range:	
Describe:	Connect a software generated digital waveform generator object. These objects share the number-space with the **DCMOTOR** and **ANALOG.OUT** output objects. The associated pin is configured as a digital output signal. **CONNECT SQUAREWAVE n [TO] BB 2**
Result:	Digital output squarewave from 1 to 500 hz.
Type or Addressable Component:	Control

ANALOG.OUT i [TO] OUT i/BB i

Command:	**ANALOG.OUT i [TO] OUT n/BB n**
Command Syntax:	**CONNECT ANALOG.OUT i [TO] OUT n/BB n**
Range:	
Describe:	Connect a generic "analog" output control to a pin/port that supports analog input. **ANALOG.OUT** shares number space with **DCMOTOR** and **SQUAREWAVE** objects. **CONNECT ANALOG.OUT i [TO] OUT 1** **CONNECT ANALOG.OUT i [TO] BB 4** **CONNECT ANALOG.OUT i [TO] BB 1**
Result:	Connect analog output to pin. If pin supports hardware pulse with modulation (**PWM**), the object uses.

Command:	**ANALOG.OUT i [TO] OUT n/BB n**
	If the pin does not support hardware-generated **PWM**, the sketch will generate **PWM** in software at 490 Hz with the duty cycle specfic between 0 (none) and 255 (full on).
Type or Addressable Component:	Control

DIGITAL.OUT i [TO] OUT n/BB n [[AS] OUTPUT]

Command:	**DIGITAL.OUT i [TO] OUT n/BB n [[AS] OUTPUT]**
Command Syntax:	**CONNECT DIGITAL.OUT i [TO] OUT n/BB n**
Range:	
Describe:	Connects a generic digital object to a specified pin or port. The connected pin is configured either as a digital output signal, default LOW, or a digital input signal, default INPUT with no pullup or pulldown enabled. The index number can refer to either an input or output. The index is shared by both items since a **DIGITAL** signal can be either an input or output. **CONNECT DIGITAL.OUT 1 [TO] OUT n/BB n**
Result:	Connect pin to digital object default output state, default **OUTPUT**, low.
Type or Addressable Component:	Control/Sensor

CONNECT-Input

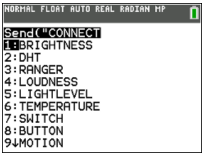
CONNECT associates a given control or sensor with a pin or port on the TI-Innovator. If the specified control or sensor is currently in use, an error will be generated. If the pin or port specified in the **CONNECT** command is currently in use, an error will be generated.

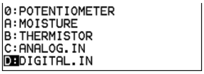
The **CONNECT** command does not generate an active response, but a variety of errors may occur during a connection attempt, such as pin-in-use, unsupported, invalid options, bad options, etc.

CONNECT 'something i' [TO] IN1/IN2/IN3/OUT1/OUT2/OUT3/BB1

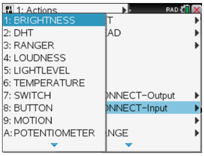
Command:	**CONNECT**
Command Syntax:	**CONNECT**
Range:	
Describe:	Associates a sensor or control with a given port or pin(s). Places the respective pin(s) in use
Result:	
Type or Addressable Component:	

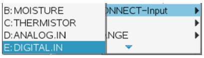
TI-84 PlusCE





TI-Nspire™ CX





BRIGHTNESS

Command:	**BRIGHTNESS**
Command Syntax:	**CONNECT BRIGHTNESS**
Range:	

Command:	BRIGHTNESS
Describe:	This command is not needed for typical use since the on-board BRIGHTNESS sensor is automatically connected. (Re-)connect the internal analog ambient light sensor. No pin or port name is used with this internal object.
Result:	Connects on-board light sensor to known analog input pin.
Type or Addressable Component:	Sensor

DHT i [TO] IN n

Command:	DHT i [TO] IN n
Command Syntax:	**CONNECT DHT i [TO] IN n**
Range:	Temperature reading default is in Celsius Humidity reading from 0 to 100 %
Describe:	The **DHT** digital temperature humidity sensor can be connected via this object. The **DHT** can be either a **DHT11** or **DHT22** and is identified automatically when connected to the system via a digital signal line. **CONNECT DHT i [TO] IN1**
Result:	Digital humidity/temperature sensors (DHT11/DHT22, type is auto-detected).
Type or Addressable Component:	Sensor

RANGER i [TO] IN n

Command:	RANGER i [TO] IN n
Command Syntax:	**CONNECT RANGER i [TO] IN n**
Range:	
Describe:	Connect an external ultrasonic distance ranging module to an input port. **CONNECT RANGER 1i [TO] IN 1**

Command:	**RANGER i [TO] IN n**
Result:	Ultrasonic ranging sensors with either individual trigger/echo pins, or same pin used for trigger/echo.
Type or Addressable Component:	Sensor

LOUDNESS i [TO] IN n

Command:	**LOUDNESS i [TO] IN n**
Command Syntax:	**CONNECT LOUDNESS i [TO] IN n**
Range:	
Describe:	The **LOUDNESS** object measure sound intensity (loudness). **CONNECT LOUDNESS i1 [TO] IN2**
Result:	Analog sound level sensors.
Type or Addressable Component:	Sensor

LIGHTLEVEL i [TO] IN n/BB n

Command:	**LIGHTLEVEL i [TO] IN n/BB n**
Command Syntax:	**CONNECT LIGHTLEVEL i [TO] IN n/BB n**
Range:	An integer value between 0 and 16383 (14 bit resolution)
Describe:	Connects an external light sensor. External light sensors can be analog sensors. **CONNECT LIGHTLEVEL 1i [TO] IN1**
Result:	Analog light level sensors is connected to the specific port.
Type or Addressable Component:	Sensor

TEMPERATURE i [TO] IN n/BB n

<table>
<tr><th>Command:</th><th>TEMPERATURE i [TO] IN n/BB n</th></tr>
<tr><td>Command Syntax:</td><td><b>CONNECT TEMPERATURE i [TO] IN n/BB n</b></td></tr>
<tr><td>Range:</td><td>Temperature reading default is in Celsius. The range depends on the specific temperature sensor being used.
Humidity reading from 0 to 100 %</td></tr>
<tr><td>Describe:</td><td>Connects a temperature sensor to the system using either of several connection methods.
<b>Note:</b> The default temperature sensor is included in the Breadboard pack

If the sensor is based on a thermistor and provides an analog output, it uses a single analog input pin. If the sensor is a DS18B20 digital temperature sensor, it uses a single bi-directional digital GPIO pin.
The analog thermistor temperature sensors is by default, assumed to be a PTC thermistor. If the thermistor is an NTC style, an optional keyword can be added to the connect command sequence to change the style of the thermistor.
The analog thermistor temperature sensor uses a specific set of thermistor constants, different than those used by the THERMISTOR object, to convert the reading into a temperature reading. The constants are used in the Steinhart-Hart model to convert the analog reading to temperature.
<table>
<tr><th>Description</th><th>Value</th></tr>
<tr><td>C1</td><td>8.76741e-8</td></tr>
<tr><td>C2</td><td>2.34125e-4</td></tr>
<tr><td>C3</td><td>1.129148e-3</td></tr>
<tr><td>R1 – reference resistance</td><td>10000.0 ohms</td></tr>
</table>
<b>CONNECT TEMPERATURE i [TO] IN 1</b> – thermistor sensor attached to analog input.
<b>CONNECT TEMPERATURE i [TO] BB 1</b> – DS18B20 digital attached to digital pin.
<b>CONNECT TEMPERATURE i [TO] I2 C</b> – LM75A attached to I2C port.
<b>CONNECT TEMPERATURE i [TO] BB 5 NTC</b> – connect an analog temperature sensor to analog input and specifies an NTC style thermistor.
<b>CONNECT TEMPERATURE i [TO] BB 6 PTC</b> – connect an analog temperature sensor to analog input and specifies a PTC style thermistor.</td></tr>
<tr><td>Result:</td><td>Analog temperature sensor.</td></tr>
<tr><td>Type or</td><td>Sensor</td></tr>
</table>

Command:	**TEMPERATURE i [TO] IN n/BB n**
Addressable Component:	

SWITCH i [TO] IN n/BB n

Command:	**SWITCH i [TO] IN n/BB n**
Command Syntax:	**CONNECT SWITCH i [TO] IN n/BB n**
Range:	
Describe:	Connect an external switch to a digital input pin. The button task will monitor the state of the switch allowing reporting for the switch on, not on, and was on since last checked. The connected pin is set to a digital input state with its internal pulldown enabled. The other side of the switch is connected to a power supply (3.3v) pin (or 5v supply if using IN3 port). Switches share number space with Buttons. **CONNECT SWITCH 1 [TO] IN 1** **CONNECT SWITCH 2 [TO] BB 5**
Result:	Connect a switch object (similar to button, but connected to **Vcc** instead of **Gnd** when enabled.)
Type or Addressable Component:	Sensor

BUTTON i [TO] IN n/BB n

Command:	**BUTTON i [TO] IN n/BB n**
Command Syntax:	**CONNECT BUTTON i [TO] IN n/BB n**
Range:	
Describe:	Connect an external button to a digital input pin. The button task will monitor the state of the button allowing reporting for the button pressed, not pressed, and was pressed since last checked. The connected pin is set to a digital input state with its internal pullup enabled. The other side of the button is connected to a ground pin. Buttons share number space with Switches.

Command:	BUTTON i [TO] IN n/BB n
	CONNECT BUTTON i [TO] IN1
Result:	Digital button/switch/etc.
Type or Addressable Component:	Sensor

MOTION i [TO] IN n/BB n

Command:	MOTION i [TO] IN n/BB n
Command Syntax:	**CONNECT MOTION i [TO] IN n/BB n**
Range:	
Describe:	Connects a digital PIR (passive infrared) motion detection sensor to a digital input pin. This sensor is monitored the same as button objects for a three-state result. **CONNECT MOTION 1i [TO] IN 1**
Result:	Passive **I/R** motion detectors.
Type or Addressable Component:	Sensor

POTENTIOMETER i [TO] IN n/BB n

Command:	POTENTIOMETER i [TO] IN n/BB n
Command Syntax:	**CONNECT POTENTIOMETER i [TO] IN n/BB n**
Range:	
Describe:	Connect an external slide or rotary potentiometer to an analog input pin. **CONNECT POTENTIOMETER 1i [TO] IN 2** **CONNECT POTENTIOMETER 1 [TO] BB 2**
Result:	Rotary- potentiometer sensors.
Type or	Sensor

Command:	**POTENTIOMETER i [TO] IN n/BB n**
Addressable Component:	

MOISTURE i [TO] IN n/BB n

Command:	**MOISTURE i [TO] IN n/BB n**
Command Syntax:	**CONNECT MOISTURE i [TO] IN n/BB n**
Range:	An integer value between 0 and 16383 (14 bit resolution)
Describe:	Connect an analog moisture sensor to return relative moisture readings. **CONNECT MOISTURE 1i [TO] IN 1**
Result:	Analog moisture sensors.
Type or Addressable Component:	Sensor

THERMISTOR i [TO] IN n/BB n

Command:	**THERMISTOR i [TO] IN n/BB n**
Command Syntax:	**CONNECT THERMISTOR i [TO] IN n/BB n**
Range:	
Describe:	Connects a PTC thermistor to the system using a single analog input pin. The thermistor sensor uses the following values in the Steinhart-Hart model to convert the reading into a temperature. These values match those in the TI Temperature probe that came with the CBL™ System and TI Lab Cradle.

Description	**Value**
C1	1.33342e-7
C2	2.22468e-4

Command:	THERMISTOR i [TO] IN n/BB n
	<table><tr><th>Description</th><th>Value</th></tr><tr><td>C3</td><td>1.02119e-3</td></tr><tr><td>R1 – reference resistance</td><td>15000.0 ohms</td></tr></table> **CONNECT THERMISTOR i [TO] IN 1** **CONNECT THERMISTOR i [TO] BB 5**
Result:	Analog thermistor.
Type or Addressable Component:	Sensor

ANALOG.IN i [TO] IN n/BB n

Command:	ANALOG.IN i [TO] IN n/BB n
Command Syntax:	**CONNECT ANALOG.IN i [TO] IN n/BB n**
Range:	
Describe:	Connect a generic "analog" input sensor to a pin/port that supports analog input. **CONNECT ANALOG.IN i [TO] IN 1** **CONNECT ANALOG.IN i [TO] BB 5**
Result:	Connect analog input to pin that supports that function (error if pin is not analog-input capable).
Type or Addressable Component:	Sensor

DIGITAL.IN i [TO] IN n/BB n [[AS] INPUT|PULLUP|PULLDOWN]

Command:	DIGITAL.IN i [TO] IN n/BB n [[AS] INPUT|PULLUP|PULLDOWN]
Command Syntax:	**CONNECT DIGITAL.IN i [TO] IN n/OUT n/BB n**

Command:	DIGITAL.IN i [TO] IN n/BB n [[AS] INPUT|PULLUP|PULLDOWN]
Range:	
Describe:	Connects a generic digital object to a specified pin or port. The connected pin is configured either as a digital output signal, default LOW, or a digital input signal, default INPUT with no pullup or pulldown enabled. The index number can refer to either an input or output. The index is shared by both items since a **DIGITAL** signal can be either an input or output. **CONNECT DIGITAL.IN 1 [TO] IN 1**
Result:	Connect pin to digital object default input state, default **INPUT**.
Type or Addressable Component:	Control/Sensor

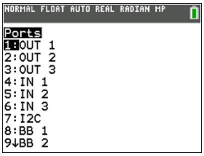
Ports

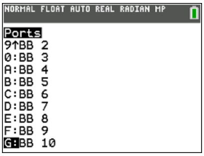
Settings menu contains operations to set the state of digital and analog pin operations such as the **LED** in the TI-Innovator™ Hub or a connected servo motor movement to states such as ON, OFF, CW (clockwise), and CCW (counterclockwise).

- 1: OUT 1
- 2: OUT 2
- 3: OUT 3
- 4: IN 1
- 5: IN 2
- 6: IN: 3
- 7: I2C
- 8: BB 1
- 9: BB 2
- 0: BB 3
- A: BB 4
- B: BB 5
- C: BB 6
- D: BB 7
- E: BB 8
- F: BB 9
- G: BB 10

TI-84 Plus CE

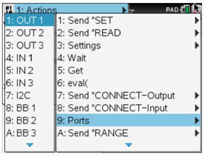
See TI-84 Plus CE Programming

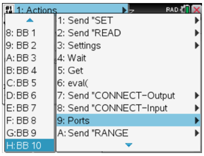




TI-Nspire™ CX

See: TI-Nspire™ CX Programming





See also: Breadboard Components and Usable Pins

RANGE

The **RANGE** command is used with several analog input sensors to re-map the internal ADC (Analog to Digital Converter) range of 0 to 16383 (14-bit ADC values) to a floating point range specified as the parameters to this command, along with the sensor to which the range is applied. The format for setting the range of a sensor is **RANGE sensor [i] minimum maximum**. To remove/reset to default the range from a given sensor, set the minimum and maximum value to zero. The minimum value must be less than the maximum value when setting a valid range.

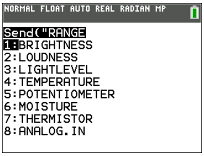
A sensors current range, if present, can be obtained by **READ sensor [i] RANGE**. A two-element list of numbers in the form *{ minimum, maximum }* will be returned.

Note: If no range has been applied to the sensor, an error will be returned if an attempt to read the sensor range is performed .

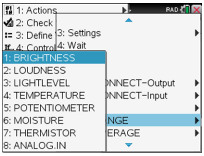
An individual sensors averaging value may be obtained by **READ sensor [i] RANGE**.

RANGE 'something' (for analog devices, maps ADC range from 0 to 16383 to the range specified, min < max, min, max any values.)

TI-84 Plus CE



TI-Nspire™ CX



BRIGHTNESS minimum maximum

Command:	BRIGHTNESS minimum maximum Advanced user
Command Syntax:	**RANGE BRIGHTNESS minimum maximum**
Range:	
Describe:	Changes/Sets the mapping of ADC input values from the ADC 0-16383 range to a user-selected range. The resulting sensor reading is mapped to this and a floating point result is returned. By default, the on-board BRIGHTNESS sensor is ranged to a 0-100 range. **RANGE BRIGHTNESS minimum maximum**
Result:	Set mapping for on-board brightness/light sensor.
Type or	Sensor

Command:	BRIGHTNESS minimum maximum Advanced user
Addressable Component:	

LOUDNESS i minimum maximum

Command:	LOUDNESS i minimum maximum Advanced user
Command Syntax:	**RANGE LOUDNESS i minimum maximum**
Range:	
Describe:	Changes/Sets the mapping of ADC input values from the ADC 0-16383 range to a user-selected range. The resulting sensor reading is mapped to this and a floating point result is returned. **RANGE LOUDNESS i minimum maximum**
Result:	Set mapping for sound-level analog sensor.
Type or Addressable Component:	Sensor

LIGHTLEVEL i minimum maximum

Command:	LIGHTLEVEL i minimum maximum Advanced user
Command Syntax:	**RANGE LIGHTLEVEL i minimum maximum**
Range:	An integer value between 0 and 16383 (14 bit resolution)
Describe:	Changes/Sets the mapping of ADC input values from the ADC 0-16383 range to a user-selected range. The resulting sensor reading is mapped to this and a floating point result is returned. **RANGE LIGHTLEVEL i minimum maximum**

Command:	**LIGHTLEVEL i minimum maximum** **Advanced user**
Result:	Set mapping for off-board light sensor (analog).
Type or Addressable Component:	Sensor

TEMPERATURE i minimum maximum

Command:	**TEMPERATURE i minimum maximum** **Advanced user**
Command Syntax:	**RANGE TEMPERATURE i minimum maximum**
Range:	
Describe:	. **RANGE TEMPERATURE i minimum maximum**
Result:	Set mapping for soil moisture analog sensor.
Type or Addressable Component:	Sensor

POTENTIOMETER i minimum maximum

Command:	**POTENTIOMETER i minimum maximum** **Advanced user**
Command Syntax:	**RANGE POTENTIOMETER i minimum maximum**
Range:	
Describe:	Changes/Sets the mapping of ADC input values from the ADC 0-16383 range to a user-selected range. The resulting sensor reading is mapped to this and a floating point result is returned. **RANGE POTENTIOMETER i minimum maximum**

Command:	**POTENTIOMETER i minimum maximum** **Advanced user**
Result:	Set mapping for rotary/linear potentiometers.
Type or Addressable Component:	Sensor

MOISTURE i minimum maximum

Command:	**MOISTURE i minimum maximum** **Advanced user**
Command Syntax:	**RANGE MOISTURE i minimum maximum**
Range:	An integer value between 0 and 16383 (14 bit resolution)
Describe:	Changes/Sets the mapping of ADC input values from the ADC 0-16383 range to a user-selected range. The resulting sensor reading is mapped to this and a floating point result is returned. **RANGE MOISTURE i minimum maximum**
Result:	Set mapping for soil moisture analog sensor.
Type or Addressable Component:	Sensor

THERMISTOR i minimum maximum

Command:	**THERMISTOR i minimum maximum** **Advanced user**
Command Syntax:	**RANGE THERMISTOR i minimum maximum**
Range:	
Describe:	. **RANGE THERMISTOR i minimum maximum**

Command:	**THERMISTOR i minimum maximum** **Advanced user**
Result:	Set mapping for xxxxxxxxxx.
Type or Addressable Component:	Sensor

ANALOG.IN i minimum maximum

Command:	**ANALOG.IN i minimum maximum** **Advanced user**
Command Syntax:	**RANGE ANALOG.IN i minimum maximum**
Range:	
Describe:	Changes/Sets the mapping of ADC input values from the ADC 0-16383 range to a user-selected range. The resulting sensor reading is mapped to this and a floating point result is returned. **RANGE ANALOG.IN i minimum maximum**
Result:	Set mapping for generic analog input objects.
Type or Addressable Component:	Sensor

AVERAGE

The **AVERAGE** command is used to set the number of ADC (Analog to Digital converter) samples taken to represent a single analog sensor reading. By default, the TI-Innovator™ Hub sets a global value of three (3) readings to be taken for a sensor measurement. This is done to reduce variation due to noise etc. This default is adjustable between 1 and 25 by the **SET AVERAGING n** command. The current default can be obtained by the **READ AVERAGING** command.

For individual sensors, the default can be changed after the **CONNECT** operation by using the **AVERAGE** command. The format is **AVERAGE sensor [i] value** where sensor is a sensor from the table below, **[i]** is the index, if needed to identify the specific sensor, and value is a number from 1 to 25.

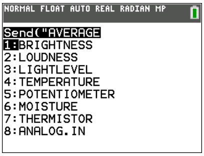
The sensor, when a sample is requested, will take value number of readings, 10 microseconds apart, summing the readings together and averaging them over the number of readings taken.

An individual sensors averaging value may be obtained by **READ sensor [i] AVERAGE**.

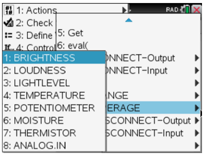
AVERAGE 'something' (for analog devices, sets the individual oversampling value for reading, from 1 to 25)

Command:	**AVERAGE**
Command Syntax:	**AVERAGE**
Describe:	Specifies the number of analog readings to take on a specific sensor to obtain a single reading of that sensor. Valid values are from 1 to 25 readings, taken 10 microseconds apart and averaged together. Sensors use the system default of 3 readings if not altered by changing the system global setting via a **SET AVERAGING** command.
Result:	
Type or Addressable Component:	

TI-84 Plus CE



TI-Nspire™ CX



BRIGHTNESS n

Command:	BRIGHTNESS n
Command Syntax:	**AVERAGE BRIGHTNESS n**
Range:	Where **n** ranges from 1 to 25
Describe:	Set the number of readings from the ADC to be used for the on-board light sensor.
Result:	Set oversampling for on-board brightness/light sensor.
Type or Addressable Component:	Sensor

LOUDNESS i n

Command:	LOUDNESS i n
Command Syntax:	**AVERAGE LOUDNESS i n**
Range:	– where **n** ranges from 1 to 25
Describe:	Set the number of readings from the ADC to be used with an external sound loudness sensor.
Result:	Set oversampling for sound-level analog sensor.
Type or Addressable Component:	Sensor

LIGHTLEVEL i n

Command:	LIGHTLEVEL i n
Command Syntax:	**AVERAGE LIGHTLEVEL i n**
Range:	– where **n** ranges from 1 to 25
Describe:	Set the number of readings from the ADC to be used for the external light

Command:	LIGHTLEVEL i n
	sensor connected to an analog input. Does not support I^{2}C light sensors.
Result:	Set oversampling for off-board light sensor (analog).
Type or Addressable Component:	Sensor

TEMPERATURE i n

Command:	TEMPERATURE i n
Command Syntax:	**AVERAGE TEMPERATURE i n**
Range:	Where **n** ranges from 1 to 25
Describe:	Set the number of readings from the ADC to be used for the external temperature sensor connected to an analog input. Does not support I^{2}C or digital temperature sensors.
Result:	When using an analog-style thermistor temperature sensor, oversample this many times.
Type or Addressable Component:	Sensor

POTENTIOMETER i n

Command:	POTENTIOMETER i n
Command Syntax:	**AVERAGE POTENTIOMETER i n**
Range:	Where **n** ranges from 1 to 25
Describe:	Set the number of readings from the ADC to be used with an external potentiometer, either a linear or rotary model.
Result:	Set oversampling for rotary/linear potentiometers.
Type or Addressable Component:	Sensor

MOISTURE i n

Command:	MOISTURE i n
Command Syntax:	**AVERAGE MOISTURE i n**
Range:	– where **n** ranges from 1 to 25
Describe:	Set the number of readings from the ADC to be used with an external moisture sensor.
Result:	Set oversampling for soil moisture analog sensor.
Type or Addressable Component:	Sensor

THERMISTOR i n

Command:	THERMISTOR i n
Command Syntax:	**AVERAGE THERMISTOR i n**
Range:	Where **n** ranges from 1 to 25
Describe:	Set the number of readings from the ADC to be used with an external thermistor connected to an analog input.
Result:	Set oversampling for thermistor device analog input.
Type or Addressable Component:	Sensor

ANALOG.IN i n

Command:	ANALOG.IN i n
Command Syntax:	**AVERAGE ANALOG.IN i n**

Command:	**ANALOG.IN i n**
Range:	Where **n** ranges from 1 to 25
Describe:	Set the number of readings from the ADC to be used for the analog sensor attached to this generic analog item.
Result:	Sets oversampling count for generic analog input.
Type or Addressable Component:	Sensor

PERIOD n

Command:	**PERIOD n**
Command Syntax:	**PERIOD n**
Range:	
Describe:	The **AVERAGE** command is somewhat unique for **PERIOD** in that it specifies how many distinct periods are to be measured and averaged together to obtain the desired measurement. Up to 25 samples may be taken to obtain the period measurement for a given pin.
Result:	Set number of samples of frequency to take to be average together to generate period.
Type or Addressable Component:	Sensor

DISCONNECT-Output

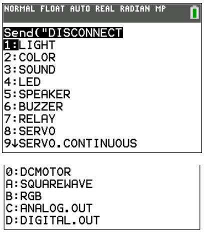
DISCONNECT breaks the association between a specified control or sensor and the pin/port it is associated with. If the specified sensor or control is not currently connected to anything, an error is generated.

The **DISCONNECT** command does not generate an active response, other than possible error responses. Pins associated with an actively connected sensor, or control, are released from use and, in general, are set to a digital input state with no enabled pullup/pulldown.

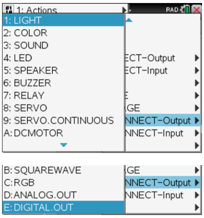
DISCONNECT - disconnect something that has been connected, by index if needed.

Command:	**DISCONNECT-Output**
Command Syntax:	**DISCONNECT**
Range:	
Describe:	Removes the association of a sensor or control with a pin, or set of pins, if such association exists. Places the pin(s) back to an **OUTPUT** state.
Result:	.
Type or Addressable Component:	

TI-84 Plus CE



TI-Nspire™ CX



LIGHT

Command:	**LIGHT**
Command Syntax:	**DISCONNECT LIGHT**

Command:	**LIGHT**
Range:	
Describe:	Disconnect the on-board **RED LED** used for direct program control from the system.
Result:	On-board **LED** disconnected
Type or Addressable Component:	Control

COLOR

Command:	**COLOR**
Command Syntax:	**DISCONNECT COLOR**
Range:	
Describe:	Disconnects the on-board **RGB LED** item from use. This action (in the initial release of the TI-Innovator™) frees three (3) hardware map-able **PWM** signals for use on other pins..
Result:	Disconnect on-board **RGB LED**.
Type or Addressable Component:	Control

SOUND

Command:	**SOUND**
Command Syntax:	**DISCONNECT SOUND**
Range:	
Describe:	Disconnect the on-board speaker from its digital pin.
Result:	Disconnects on-board speaker.

Command:	**SOUND**
Type or Addressable Component:	Control

LED i

Command:	**LED i**
Command Syntax:	**DISCONNECT LED i**
Range:	
Describe:	Disconnect an external **LED** object from the system.
Result:	**LED i** is disconnected
Type or Addressable Component:	Control

SPEAKER i

Command:	**SPEAKER i**
Command Syntax:	**DISCONNECT SPEAKER i**
Range:	
Describe:	Disconnect an external speaker from its digital pin.
Result:	Disconnect a speaker from a digital output pin.
Type or Addressable Component:	Control

BUZZER i

Command:	**BUZZER i**
Command Syntax:	**DISCONNECT BUZZER i**
Range:	
Describe:	Disconnect an active buzzer from the system. Active buzzers play a tone when their signal is set high/on, and stop the tone when the signal is dropped to ground. **DISCONNECT BUZZER i**
Result:	**ACTIVE** buzzers disconnected from a digital pin.
Type or Addressable Component:	Control

RELAY i

Command:	**RELAY i**
Command Syntax:	**DISCONNECT RELAY i**
Range:	
Describe:	Disconnect a digital relay interface from the system.
Result:	Relay disconnected.
Type or Addressable Component:	Control

SERVO i

Command:	**SERVO i**
Command Syntax:	**DISCONNECT SERVO i**

Command:	**SERVO i**
Range:	
Describe:	Disconnect a sweep or continuous **SERVO** motor from the digital pin associated with the motor.
Result:	Servo motor disconnected.
Type or Addressable Component:	Control

SERVO CONTINOUS i

Command:	**SERVO CONTINOUSi**
Command Syntax:	**DISCONNECT SERVO CONTINOUSi**
Range:	
Describe:	Disconnect a sweep or continuous **SERVO** motor from the digital pin associated with the motor.
Result:	Servo motor disconnected.
Type or Addressable Component:	Control

DCMOTOR i

Command:	**DCMOTOR i**
Command Syntax:	**DISCONNECT DCMOTOR i**
Range:	
Describe:	Disconnects a **DCMOTOR** object from the system. **DCMOTOR**, **ANALOG.OUT**, and **SQUAREWAVE** all share the same number space of items. **DCMOTOR** requires external power.
Result:	Disconnect **DCMOTOR** from pin.

Command:	**DCMOTOR i**
Type or Addressable Component:	Control

SQUAREWAVE i

Command:	**SQUAREWAVE i**
Command Syntax:	**DISCONNECT SQUAREWAVE i**
Range:	
Describe:	Disconnect the software generated squarewave generator from an associated digital output pin. The pin reverts to digital input upon disconnect.
Result:	Disconnect squarewave function from pin(s), stops squarewave generation.
Type or Addressable Component:	Control

RGB i

Command:	**RGB i**
Command Syntax:	**DISCONNECT RGB i**
Range:	
Describe:	Disconnect an external **RGB LED** from the system. These objects use three hardware **PWM** signals to properly operate, so in the initial product release, the on-board **COLOR** object must be disconnected to connect one of these objects.
Result:	Disconnect **RGB** and free up **PWM** outputs for use elsewhere.
Type or Addressable Component:	Control

ANALOG.OUT i

Command:	**ANALOG.OUT i**
Command Syntax:	**DISCONNECT ANALOG.OUT i**
Range:	
Describe:	Disconnects the connected generic analog output device specified, freeing a hardware map-able **PWM** if it is in use with the object.
Result:	Disconnect generic analog **PWM** output from pin.
Type or Addressable Component:	Control

DIGITAL.OUT i

Command:	**DIGITAL.OUT i**
Command Syntax:	**DISCONNECT DIGITAL.OUT i**
Range:	
Describe:	Disconnect a generic **DIGITAL** object. The associated pin is reverted to a digital **INPUT** pin with no enabled pullup or pulldown. The **DIGITAL** object number can be used to refer the same pin in either input, or output form…
Result:	Disconnect digital input object.
Type or Addressable Component:	Control/Sensor

DISCONNECT-Input

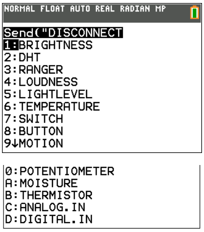
DISCONNECT breaks the association between a specified control or sensor and the pin/port it is associated with. If the specified sensor or control is not currently connected to anything, an error is generated.

The **DISCONNECT** command does not generate an active response, other than possible error responses. Pins associated with an actively connected sensor, or control, are released from use and, in general, are set to a digital input state with no enabled pullup/pulldown.

DISCONNECT - disconnect something that has been connected, by index if needed.

Command:	**DISCONNECT-Input...**
Command Syntax:	**DISCONNECT**
Range:	
Describe:	Removes the association of a sensor or control with a pin, or set of pins, if such association exists. Places the pin(s) back to an **INPUT** state.
Result:	.
Type or Addressable Component:	

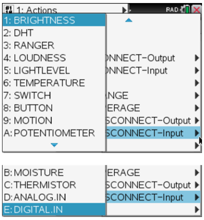
TI-84 Plus CE



```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("DISCONNECT
1:BRIGHTNESS
2:DHT
3:RANGER
4:LOUDNESS
5:LIGHTLEVEL
6:TEMPERATURE
7:SWITCH
8:BUTTON
9↓MOTION

0:POTENTIOMETER
A:MOISTURE
B:THERMISTOR
C:ANALOG.IN
D:DIGITAL.IN
```

TI-Nspire™ CX



```
1: Actions                RAD
1: BRIGHTNESS
2: DHT
3: RANGER
4: LOUDNESS          )NNECT−Output
5: LIGHTLEVEL        )NNECT−Input
6: TEMPERATURE
7: SWITCH            NGE
8: BUTTON            ERAGE
9: MOTION            SCONNECT−Output
A: POTENTIOMETER     SCONNECT−Input

B: MOISTURE          ERAGE
C: THERMISTOR        SCONNECT−Output
D: ANALOG.IN         SCONNECT−Input
E: DIGITAL.IN
```

BRIGHTNESS

Command:	**BRIGHTNESS**
Command Syntax:	**DISCONNECT BRIGHTNESS**
Range:	

Command:	BRIGHTNESS
Describe:	Disconnects the internal connection to the on-board **BRIGHTNESS** (light sensor) object.
Result:	Disconnect on-board **LIGHT** sensor.
Type or Addressable Component:	Sensor

DHT i

Command:	DHT i
Command Syntax:	**DISCONNECT DHT i**
Range:	Temperature reading default is in Celsius Humidity reading from 0 to 100 %
Describe:	Disconnects the specified digital humidity **DHT** and temperature sensor from the system. This also removes that object from the period scan list of style sensors in the DHT task.
Result:	Digital humidity/temperature sensor(s) disconnected.
Type or Addressable Component:	Sensor

RANGER i

Command:	RANGER i
Command Syntax:	**DISCONNECT RANGER i**
Range:	
Describe:	Disconnect a digital ultrasonic ranging sensor from the two digital pins it uses.
Result:	Ultrasonic ranging sensor disconnected.

Command:	**RANGER i**
Type or Addressable Component:	Sensor

LOUDNESS i

Command:	**LOUDNESS i**
Command Syntax:	**DISCONNECT LOUDNESS i**
Range:	
Describe:	Disconnect an analog sound intensity (**LOUDNESS**) sensor.
Result:	Analog sound level sensor disconnected
Type or Addressable Component:	Sensor

LIGHTLEVEL i

Command:	**LIGHTLEVEL i**
Command Syntax:	**DISCONNECT LIGHTLEVEL i**
Range:	
Describe:	Disconnect an external light sensor.
Result:	Light sensor disconnected.
Type or Addressable Component:	Sensor

TEMPERATURE i

Command:	TEMPERATURE i
Command Syntax:	**DISCONNECT TEMPERATURE i**
Range:	Temperature reading default is in Celsius. The range depends on the specific temperature sensor being used. Humidity reading from 0 to 100 %
Describe:	Disconnect a connected temperature sensor from the system. **TEMPERATURE** sensors can be either analog (thermistor-style). Disconnecting from the analog or digital reverts the associated pins to INPUT.
Result:	Disconnect temperature sensor.
Type or Addressable Component:	Sensor

SWITCH

Command:	SWITCH
Command Syntax:	**DISCONNECT SWITCH i**
Range:	
Describe:	Disconnect a switch from its digital pin. The pin reverts to INPUT state, and the switch is removed from the scanning sequence in the BUTTON task.
Result:	disconnect switch object from pin
Type or Addressable Component:	Sensor

BUTTON i

Command:	BUTTON i
Command	**DISCONNECT BUTTON i**

Command:	BUTTON i
Syntax:	
Range:	
Describe:	Disconnects the specified button object from the system and removes it from the list of scanned buttons/switches in the **BUTTON** task.
Result:	Digital button/switch is disconnected.
Type or Addressable Component:	Sensor

MOTION i

Command:	MOTION i
Command Syntax:	**DISCONNECT MOTION i**
Range:	
Describe:	Disconnects a digital **PIR** (passive infrared) **MOTION** detector and removes the object from the scanning list in the **BUTTON** task.
Result:	Disconnect passive **I/R** motion detectors
Type or Addressable Component:	Sensor

POTENTIOMETER i

Command:	POTENTIOMETER i
Command Syntax:	**DISCONNECT POTENTIOMETER i**
Range:	
Describe:	Disconnect an analog variable resistor (**POTENTIOMETER**) from the system

Command:	**POTENTIOMETER i**
Result:	Disconnect a rotary/linear potentiometer sensors
Type or Addressable Component:	Sensor

MOISTURE i

Command:	**MOISTURE i**
Command Syntax:	**DISCONNECT MOISTURE i**
Range:	
Describe:	Disconnect an analog moisture sensor.
Result:	Disconnect analog moisture sensors
Type or Addressable Component:	Sensor

THERMISTOR i

Command:	**THERMISTOR i**
Command Syntax:	**DISCONNECT THERMISTOR i**
Range:	
Describe:	Disconnect an analog thermistor sensor from the associated pin.
Result:	disconnect analog thermistor
Type or Addressable Component:	Sensor

ANALOG.IN i

Command:	**ANALOG.IN i**
Command Syntax:	**DISCONNECT ANALOG.IN i**
Range:	
Describe:	Disconnects the connected generic analog input device specified.
Result:	Disconnect generic analog input from pin.
Type or Addressable Component:	Sensor

DIGITAL.IN i

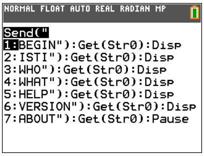
Command:	**DIGITAL.IN i**
Command Syntax:	**DISCONNECT DIGITAL.IN i**
Range:	
Describe:	Disconnect a generic **DIGITAL** object. The associated pin is reverted to a digital **INPUT** pin with no enabled pullup or pulldown. The **DIGITAL** object number can be used to refer the same pin in either input, or output form.
Result:	Disconnect digital input object.
Type or Addressable Component:	Control/Sensor

MANAGE

The **Manage** menu pastes a **Send(** command with the following management items.

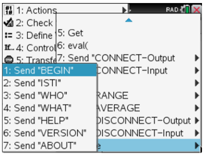
Str0 is displayed on Home Screen with information if requested in the command.

TI-84 Plus CE



```
NORMAL FLOAT AUTO REAL RADIAN MP
Send("
1:BEGIN"):Get(Str0):Disp
2:ISTI"):Get(Str0):Disp
3:WHO"):Get(Str0):Disp
4:WHAT"):Get(Str0):Disp
5:HELP"):Get(Str0):Disp
6:VERSION"):Get(Str0):Disp
7:ABOUT"):Get(Str0):Pause
```

TI-Nspire™ CX



BEGIN

The **BEGIN** command disconnects all connected sensors and controls, re-initializes all sensor/control memory within the sketch, and resets the sensor average default value, error formatting, and flow control defaults. Additionally, all **IN***n* port pins, and the breadboard connector (**BB***n*) pins are set to the **INPUT** pin mode. All **OUT***n* port pins are set to the **INPUT** state, and allowed to float, including **OUT3** which will read as high due to a pullup resistor from the 5V supply on this pin.

When the entire process completes, a response of **READY** is sent to the host system. This response must be waited for by the host before any further operations are performed. Additional commands may be in the command queue to be executed, but will not be acted upon until this command completes.

BEGIN

Command:	**BEGIN**
Command Syntax:	**SEND("BEGIN"**
Describe:	Disassociates sensors from ports or pins, and resets all settings back to defaults. Disconnects any connected sensor objects and restores system to state as if **RESET** button pressed.
Result:	Responds with a "**READY**" when completed.
Type or Addressable Component:	Not Applicable

Note: The [:] is used to sequence command lines on one command line. The **Manage…** menu pastes a convenient set of commands to then display the information in **Str0** on the home screen.

ISTI

The **ISTI** command is used to synchronize communications with the sketch. The response to this command must be **TISTEM**. Responses may have a leading ***NUL*** (0) character on initial power-on of the Innovator hub. All responses from the Innovator hub will be followed with a *CR/LF* pair that may or may not be stripped by software layers in the host system prior to the response being received by the application layer on the host system.

ISTI

Command:	**ISTI**
Command Syntax:	**ISTI**
Describe:	Send "ISTI", and get response "TISTEM".
Result:	Handshake command used to determine presence of a supported “sketch” on the TI-Innovaotr™ Hub.
Type or Addressable Component:	

WHO

WHO is an identification command (similar to the **ISTI** handshake command below) that can be used to determine what product is present and running the sketch.

The correct response to **WHO** is “**TI INNOVATOR ON MSP432**” when this command is sent to the TI-Innovator Hub.

WHO

Command:	**WHO**
Command Syntax:	**WHO**
Describe:	Identification command to determine what product is running the sketch. Send (“WHO”) Get Str0 Disp Str0
Result:	Identify the product - TI INNOVATOR ON MSP432.
Type or Addressable Component:	

WHAT

The **WHAT** command is an identification command. The response to **WHAT** for TI-Innovator is “**TI INNOVATOR HUB**”.

WHAT

Command:	**WHAT**
Command Syntax:	**WHAT**
Describe:	Product name query. Identify the product - "**TI INNOVATOR HUB**" Send (“WHAT”) Get Str0 Disp Str0
Result:	Identify the product.
Type or Addressable Component:	

HELP

HELP is used to obtain quick information about each of these commands. The **HELP command-name** is sent, and generates a string response with a one-line description of the given command.

HELP

Command:	**HELP**
Command Syntax:	**HELP**
Describe:	Provides per command quick help information. i.e. HELP SET, etc.
Result:	
Type or Addressable Component:	

VERSION

The **VERSION** command has a response that represents the current version of the sketch running on the TI-Innovator™ Hub.

The version will be of the *major.minor.patch.build* form in released products; for example, 1.0.0.

VERSION

Command:	**VERSION**
Command Syntax:	**VERSION**
Describe:	Returns version number (and possibly Accurev stream name from which sketch was built).
Result:	Report the version of the sketch in format *major.minor.patch.build*. Send (“VERSION”) Get Str0 Disp Str0
Type or Addressable Component:	

ABOUT

The **ABOUT** command response is the product line name along with a copyright date and owner. The current response to this command is “**TI INNOVATOR (C)2015-2016 TEXAS INSTRUMENTS**”.

ABOUT

Command:	**ABOUT**
Command Syntax:	**ABOUT**
Describe:	Product name and copyright information returned. Send (“ABOUT”) Get Str0 Disp Str0
Result:	Returns copyright string. "**TI INNOVATOR (C)2015-2016 TEXAS INSTRUMENTS**"
Type or Addressable Component:	

Additional Supported Commands

The following sets of supported commands are not found in the Hub Menus.

Additional SET Commands

FORMAT ERROR STRING/NUMBER

Command:	**FORMAT ERROR STRING/NUMBER** **Advanced user**
Command Syntax:	**SET FORMAT ERROR STRING/NUMBER**
Range:	
Describe:	Used for setting error return format and optional audible tone on error. **SET FORMAT ERROR STRING/NUMBER** – returned error codes in string or numeric format.
Result:	Sets the format for the return of error information (numbers, or strings).
Type or Addressable Component:	Setting

FORMAT ERROR NOTE/QUIET

Command:	**FORMAT ERROR NOTE/QUIET** **Advanced user**
Command Syntax:	**SET FORMAT ERROR NOTE/QUIET**
Range:	
Describe:	Used for setting error return format and optional audible tone on error. **SET FORMAT ERROR NOTE/QUIET** – error display flash accompanied by speaker sound or no sound.
Result:	Enables tones, or disables tones in addition to the string/number reporting

Command:	**FORMAT ERROR NOTE/QUIET** **Advanced user**
	above.
Type or Addressable Component:	Setting

FLOW [TO] ON/OFF

Command:	**FLOW [TO] ON/OFF** **Advanced user**
Command Syntax:	**SET FLOW [TO] ON/OFF**
Range:	
Describe:	Enables (**ON**) or disables (**OFF**) the software flow control mechanism between the sketch and the communications hardware. **NOTE:** When the **SEGDISP** module is **CONNECT**ed, this setting determines whether or not the display module shows error information (flow control disabled), or command queue depth (flow control enabled).
Result:	Turn on xon/xoff flow control, or turn off (no flow control)
Type or Addressable Component:	Setting

OUT1/2/3 [TO]

Command:	**OUT1/2/3 [TO]**

Command Syntax:	**OUT1/2/3 [TO] ...**
	SET OUTn 0-255
	SET OUTn HIGH/ON
	SET OUTn LOW/OFF

Command:	**OUT1/2/3 [TO]**
Range:	Set analog PWM value on **OUT** port(s) of the TI-Innovator™Hub
Describe:	Direct output of information to a given output port. These are PWM outputs on the TI-Innovator™ Hub. Set analog PWM value on TI-Innovator™ Hub **OUT** port(s). **SET OUTn 0-255** – 0=off, 255=on, anything else is a PWM signal @ 500 Hz with duty cycle high from 1 to 254, where that range provides a percentage of the high-time signal of the waveform. **SET OUTn HIGH/ON** – same as 255 **SET OUTn LOW/OFF** – same as 0
Result:	Set analog **PWM** value on **OUT** port(s) of the TI-Innovator™ Hub
Type or Addressable Component:	Port

BUZZER i

Command:	**BUZZER i**
Command Syntax:	**READ BUZZER i**
Range:	
Describe:	Returns the current state of the active buzzer specified; 0 = *silent*, 1 = *playing tone*.
Result:	Returns state of active buzzer, 0=silent, 1=on
Type or Addressable Component:	Control

COLOR

Command:	**COLOR**
Command Syntax:	**READ COLOR**
Range:	
Describe:	Read the current output state of the on-board **COLOR RGB LED** with sub-components **.RED**, **.GREEN**, **.BLUE**. When reading the entire item, a list of three values is returned, with values between 0 and 255 where 0=off, 255=full on, and values in between indicate **PWM** levels. **READ COLOR** – returns list of 3 values representing { red, green, blue } PWM levels **READ COLOR.RED** **READ COLOR.GREEN** **READ COLOR.BLUE** See Also: **RGB i**
Result:	Returns list of 3 values representing { red, green, blue } **PWM** levels.

Command:	**COLOR**
	Returns **RED/GREEN/BLUE** values for on-board **RGB** (color) **LED**.
Type or Addressable Component:	Control

COLOR.RED

Command:	**COLOR RED**
Command Syntax:	**READ COLOR.RED**
Range:	
Describe:	Read the current output state of the on-board **COLOR RGB LED** with sub-components **.RED**, **.GREEN**, **.BLUE**. When reading the entire item, a list of three values is returned, with values between 0 and 255 where 0=off, 255=full on, and values in between indicate **PWM** levels. **READ COLOR.RED**
Result:	Returns values representing {red} **PWM** levels. Returns **RED** values for on-board **RGB** (color) **LED**.
Type or Addressable Component:	Control

COLOR.GREEN

Command:	**COLOR GREEN**
Command Syntax:	**READ COLOR.GREEN**
Range:	

Command:	**COLOR GREEN**
Describe:	Read the current output state of the on-board **COLOR RGB LED** with sub-components **.RED**, **.GREEN**, **.BLUE**. When reading the entire item, a list of three values is returned, with values between 0 and 255 where 0=off, 255=full on, and values in between indicate **PWM** levels. **READ COLOR.GREEN**
Result:	Returns list of 3 values representing { red, green, blue } **PWM** levels. Returns **RED/GREEN/BLUE** values for on-board **RGB** (color) **LED**.
Type or Addressable Component:	Control

COLOR.BLUE

Command:	**COLOR BLUE**
Command Syntax:	**READ COLOR.BLUE**
Range:	
Describe:	Read the current output state of the on-board **COLOR RGB LED** with sub-components **.RED**, **.GREEN**, **.BLUE**. When reading the entire item, a list of three values is returned, with values between 0 and 255 where 0=off, 255=full on, and values in between indicate **PWM** levels. **READ COLOR.BLUE**
Result:	Returns list of 3 values representing { red, green, blue } **PWM** levels. Returns **RED/GREEN/BLUE** values for on-board **RGB** (color) **LED**.
Type or Addressable Component:	Control

DCMOTOR i

Command:	**DCMOTOR i**
Command Syntax:	**READ DCMOTOR i**
Range:	
Describe:	Motor that converts direct current electrical power into mechanical power.
Result:	Returns whether dcmotor is running (1) or stopped (0).
Type or Addressable Component:	Control

DIGITAL.OUT i

Command:	**DIGITAL.OUT i**
Command Syntax:	**READ DIGITAL.OUT i**
Range:	
Describe:	Returns the current state of the digital pin connected to the DIGITAL object, or the cached state of the digital output value last SET to the object.
Result:	Return 0 (output low), 1 (output high).
Type or Addressable Component:	Control/Sensor

FORMAT

Command:	FORMAT **Advanced user**
Command Syntax:	**READ FORMAT**
Range:	
Describe:	Return the current formatting flags for error reporting. The value returned is a byte value indicating various flags. Masking with values indicates what error reporting options are active. 1 = ERROR strings reported 2 = ERROR numbers reported +4 = ERROR TONE enabled, if not set, errors are reported silently.
Result:	Read error format (1=strings, 2=numbers, +4 to either: tones enabled).
Type or Addressable Component:	Setting

FLOW

Command:	FLOW **Advanced user**
Command Syntax:	**READ FLOW**
Range:	
Describe:	Returns the current flow control setting; 0=*disabled*, 1=*enabled*.
Result:	Read current flow control, 0 = none, 1 = xon/xoff
Type or Addressable Component:	Setting

IN1/IN2/IN3

Command:	**IN1/IN2/IN3**
Command Syntax:	**READ IN1** **READ IN2** **READ IN3**
Range:	
Describe:	Read the value present on the indicated port, and return that value to the host.
Result:	Read value of analog port on TI STEM board
Type or Addressable Component:	Port

LAST ERROR

Command:	**LAST ERROR**
Command Syntax:	**READ LAST ERROR**
Range:	
Describe:	Returns the last reported error from the last operation. Depending on the **FORMAT ERROR** setting, the response may be a **STRING** or a **NUMBER**.
Result:	Return last encountered error, resets automatically to 0, no error.
Type or Addressable Component:	Setting

LED i

Command:	**LED i**
Command Syntax:	**READ LED i**
Range:	
Describe:	Read the current state of the specified **LED**. If the **LED** is digital, a 0 or 1 is returned indicating the **LED** is off or on. If the **LED** is connected to a **PWM** output, a value from 0 to 255 will be returned, indicating the current **PWM** level where 0 is off, 255 is full on, and values in between indicate the current **PWM** setting.
Result:	Get state of **LED**, 0 or 1 if digital, 0-255 if **PWM** on analog.
Type or Addressable Component:	Control

LIGHT

Command:	**LIGHT**
Command Syntax:	**READ LIGHT**
Range:	
Describe:	Returns the state of the on-board **RED LED** (digital only). A value of 0 is off, and 1 is on.
Result:	Get current state of on-board red **LED** (0=off, 1=on).
Type or Addressable Component:	Control

OUT1/2/3

Command:	**OUT1/2/3**
Command Syntax:	**READ OUT1** **READ OUT2** **READ OUT3**
Range:	
Describe:	Read value of current port as input (may be a digital read since these do not support analog-input. **READ OUT1/OUT2/OUT3**
Result:	Read value of analog port on **TI STEM** board.
Type or Addressable Component:	Port

PWR

Command:	**PWR**
Command Syntax:	**READ PWR**
Range:	
Describe:	Returns the current state of presence of external power connected to the **PWR** port. The **PWR** port is read, and a status value of 0 (not present) or 1 (present) is returned, based on whether or not external power is available. **READ PWR**
Result:	Returns state of external power presence on **PWR** port (0=not present, 1=ext pwr present).
Type or Addressable Component:	Status

RELAY i

Command:	**RELAY i**
Command Syntax:	**READ RELAY i**
Range:	
Describe:	Return the current state of the specified relay. 0 = OFF, 1 = ON.
Result:	Read state of relay - 0=not active 1=active.
Type or Addressable Component:	Control

RESOLUTION

Command:	**RESOLUTION**
Command Syntax:	**READ RESOLUTION**
Range:	
Describe:	Returns the bit resolution used by the system for ADC readings.
Result:	Returns ADC resolution in use, in bits (default is 14).
Type or Addressable Component:	Setting

RGB i

Command:	**RGB i**
Command Syntax:	**READ RGB i**

Command:	**RGB i**
Range:	
Describe:	Same as the **COLOR** object referenced above, and has sub-objects named **RED**, **GREEN**, and **BLUE**. This command returns the current **PWM** level that the specified object is using. **READ RGB i** – returns a 3 element list, consisting of the { red, green, blue } color level. **READ RED i** – returns just the current red-component level. **READ GREEN i** **READ BLUE i**
Result:	Get state of **RGB LED**, {r,g,b} list values
Type or Addressable Component:	Control

RED i

Command:	**RED i**
Command Syntax:	**READ RED i**
Range:	
Describe:	Same as the **COLOR** object referenced above, and has sub-objects named **RED**, **GREEN**, and **BLUE**. This command returns the current PWM level that the specified object is using. **READ RGB i** – returns a 3 element list, consisting of the { red, green, blue } color level. **READ RED i** – returns just the current red-component level.
Result:	Get state of **RGB RED** component.
Type or Addressable Component:	Control

GREEN i

Command:	**GREEN i**
Command Syntax:	**READ GREEN i**
Range:	
Describe:	Same as the **COLOR** object referenced above, and has sub-objects named **RED**, **GREEN**, and **BLUE**. This command returns the current PWM level that the specified object is using. **READ RGB i** – returns a 3 element list, consisting of the { red, green, blue } color level. **READ GREEN i** – returns just the current green-component level.
Result:	Get state of **RGB GREEN** component.
Type or Addressable Component:	Control

BLUE i

Command:	**BLUE i**
Command Syntax:	**READ BLUE i**
Range:	
Describe:	Same as the **COLOR** object referenced above, and has sub-objects named **RED**, **GREEN**, and **BLUE**. This command returns the current PWM level that the specified object is using. **READ RGB i** – returns a 3 element list, consisting of the { red, green, blue } color level. **READ BLUE i** – returns just the current blue-component level
Result:	Get state of **RGB BLUE** component.
Type or Addressable Component:	Control

SERVO i

Command:	**SERVO i**
Command Syntax:	**READ SERVO i**
Range:	
Describe:	Returns the current position of a sweep servo in the range -90 to 90, OR the current speed of rotation of a continuous servo motor. Additionally, the current “calibration” setting for the servo which consists of a 2-element list representing the lower and upper microsecond pulse widths corresponding to the sweep/rotation ranges may be read. **READ SERVO i** – get current sweep position or rotation speed/direction. **READ SERVO i CALIBRATION** – get current microsecond range for sweep or rotation.
Result:	Return current servo position in degrees from -90 to +90.
Type or Addressable Component:	Control

SERVO i CALIBRATION

Command:	**SERVO i CALIBRATION** **Advanced user**
Command Syntax:	**READ SERVO i CALIBRATION**
Range:	
Describe:	Returns the current position of a sweep servo in the range -90 to 90, OR the current speed of rotation of a continuous servo motor. Additionally, the current “calibration” setting for the servo which consists of a 2-element list representing the lower and upper microsecond pulse widths corresponding to the sweep/rotation ranges may be read. **READ SERVO i CALIBRATION** – get current microsecond range for sweep or rotation.
Result:	Return current servo position in degrees from -90 to +90.
Type or Addressable Component:	Control

SOUND

Command:	**SOUND**
Command Syntax:	**READ SOUND**
Range:	
Describe:	Returns a value indicating whether sound is currently being played (1) or not (0) through the on-board speaker.
Result:	Return whether on-board speaker is playing a tone (1) or is silent(0).
Type or Addressable Component:	Control

SPEAKER i

Command:	**SPEAKER i**
Command Syntax:	**READ SPEAKER i**
Range:	
Describe:	Returns a value indicating whether sound is currently being played (1) or not (0) through an external speaker.
Result:	Return whether speaker is playing a tone (1) or silent (0).
Type or Addressable Component:	Control

SQUAREWAVE i

Command:	**SQUAREWAVE i**
Command Syntax:	**READ SQUAREWAVE i**
Range:	
Describe:	Returns a 0 the current squarewave object is not active. A value of 1 is returned if the object is actively generating an output.
Result:	Returns whether squarewave is active (1) or not active (0).
Type or Addressable Component:	Control

PERIOD n

Command:	PERIOD n
Command Syntax:	**PERIOD n**
Range:	
Describe:	The **AVERAGE** command is somewhat unique for **PERIOD** in that it specifies how many distinct periods are to be measured and averaged together to obtain the desired measurement. Up to 25 samples may be taken to obtain the period measurement for a given pin.
Result:	Set number of samples of frequency to take to be average together to generate period.
Type or Addressable Component:	Sensor

CALIBRATE

CALIBRATE is used to set various sensor and control values that do not otherwise fit within a means of setting any other way. For thermistors and temperature sensors that use an analog input port, it can be used to adjust the coefficients of the Steinhart-Hart equation used to map thermistor readings to temperature values. For servo motors, it is used to adjust the PWM pulse width within the range for a servo motor, where the zero position is set at 1500 microseconds. It is also used to set the calibration frequency for the DDS signal generator module (default is 24MHz).

For sensors supporting calibration, the value(s) may be obtained by **READ sensor [i] CALIBRATION**.

SERVO i / SERVO.CONTINUOUS i

Command:	**SERVO i /SERVO.CONTINUOUS i minimum maximum** **Advanced user**
Command Syntax:	**CALIBRATE SERVO i minimum maximum**
Range:	
Describe:	Servos operate by using pulse modulation where the high pulse width determines both direction of servo operation and possibly the speed of operation. The time between pulses is generally 20 milliseconds and is not adjustable by this command. The pulse width generally varies around a mid-point of 1.5 milliseconds (1500 microseconds). Pulse widths less than 1.5 milliseconds cause servo operation in one direction, while pulse widths greater than 1.5 milliseconds cause operation in the opposite direction. The **CALIBRATE** command for **SERVO** allows programmable changes to the minimum and maximum pulse widths. Parameters are pulse width times in microseconds. Current defaults are minimum 600 and maximum 2400 microseconds.
Result:	Set minimum and maximum pulse width for servo motor, values in microseconds, default 600 and 2400.
Type or Addressable Component:	Control

TEMPERATURE i C1 C2 C3 R1

Command:	**TEMPERATURE i C1 C2 C3 R1** **Advanced user**
Command Syntax:	**CALIBRATE TEMPERATURE i C1 C2 C3 R1**
Range:	
Describe:	The **CALIBRATE** command for analog temperature sensors allows changing the default Steinhart-Hart equation coefficients to match those of the thermistor element in the sensor being used. The default values are: C1: 8.76741e-8 C2: 2.34125e-4 C3: 1.129148e-3 R1: 10000.0 (reference resistor value = 10kΩ)
Result:	When using an analog-style thermistor temperature sensor.
Type or Addressable Component:	Sensor

THERMISTOR i C1 C2 C3 R1

Command:	**THERMISTOR i C1 C2 C3 R1** **Advanced user**
Command Syntax:	**CALIBRATE THERMISTOR i C1 C2 C3 R1**
Range:	
Describe:	The **CALIBRATE** command for analog thermistors allows changing the default Steinhart-Hart equation coefficients to match those of the thermistor element in the sensor being used. The default values are: C1: 1.33342e-7 C2: 2.22468e-4 C3: 1.02119e-3 R1: 15000.0 (reference resistor value = 15kΩ)

Command:	THERMISTOR i C1 C2 C3 R1 Advanced user
Result:	Where c1/c2/c3 are float constants for the Steinhart-Hart equation. … that models the thermistor, and r is resistance for the reference. … resistor used to create a voltage divider with the thermistor.
Type or Addressable Component:	Sensor