Meet the TI-Innovator Hub

TI-Nspire CXII

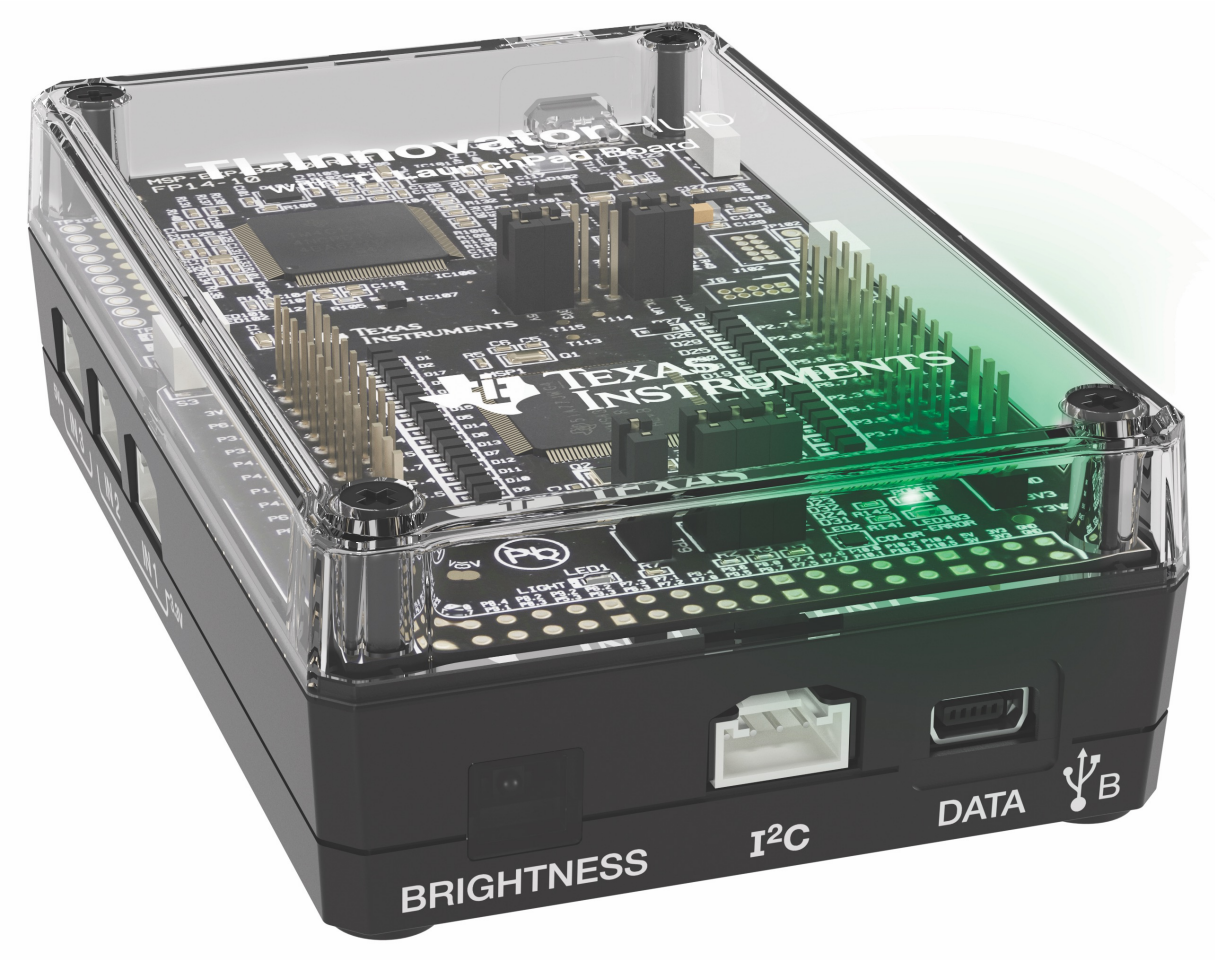
Python

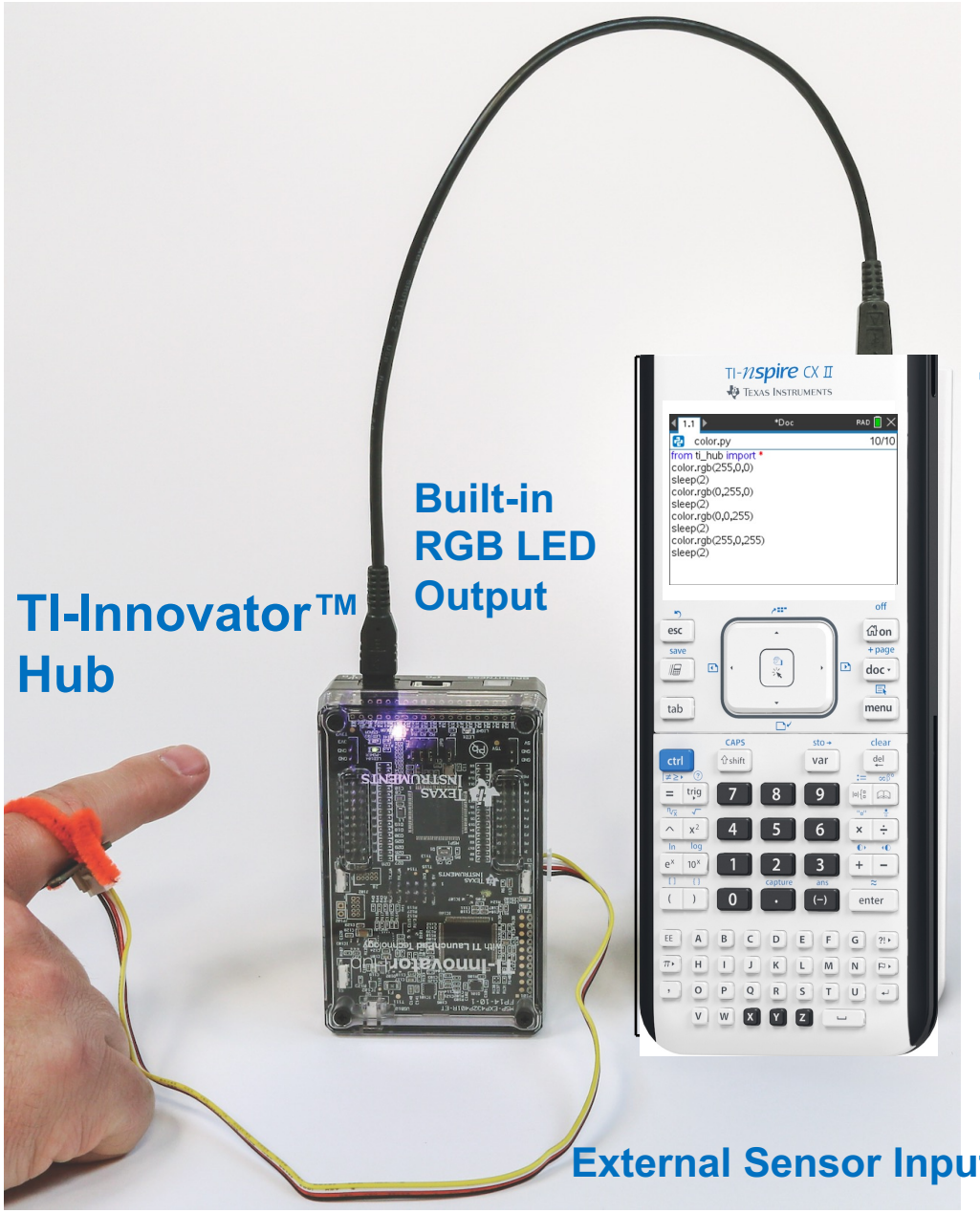
Texas Instruments

@ticalculators

www.TIstemProjects.com

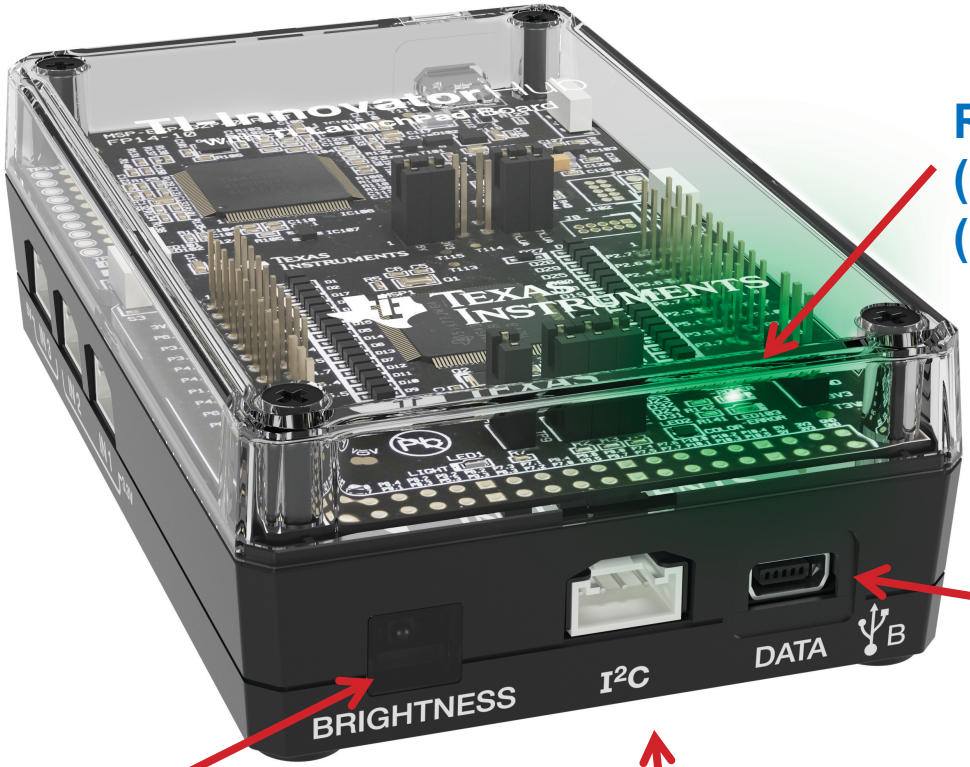
Meet the TI-Innovator Hub





TI Graphing Calculator

Hub from the Front



Red-Green-Blue (RGB) Color LED (Built-in)

USB Port (mini) to connect to calculator and computer

Brightness Sensor (Built-in)

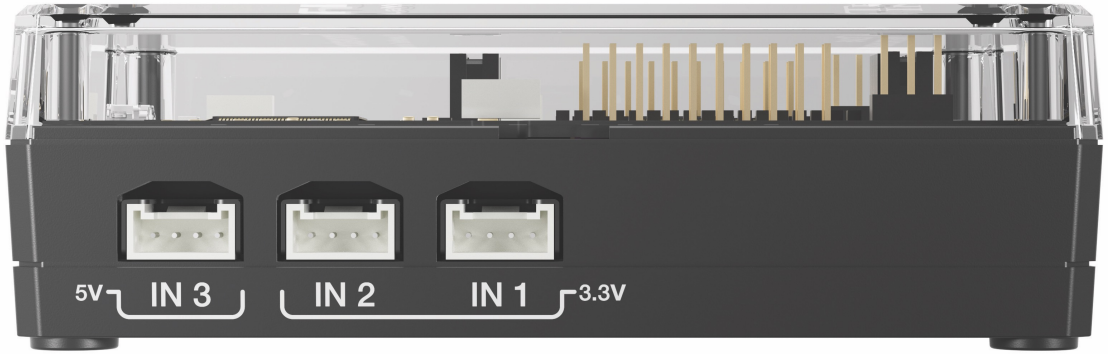
I2C Port used to connect to Rover

Hub from the Bottom

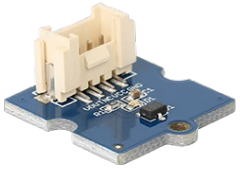
Speaker (Built-in)

Hub from the side – input ports

Input ports for external sensors with Grove connectors

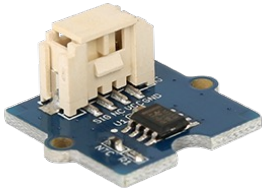


3.3 Volt ports, IN 1 and IN 2, required for Ranger and DHT

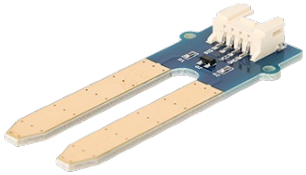


Magnetic Field (Hall Effect)

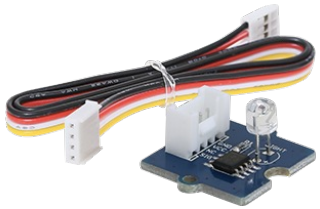
Vernier SensorLink Adaptor



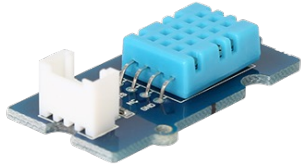
Temperature



Moisture



Light Level

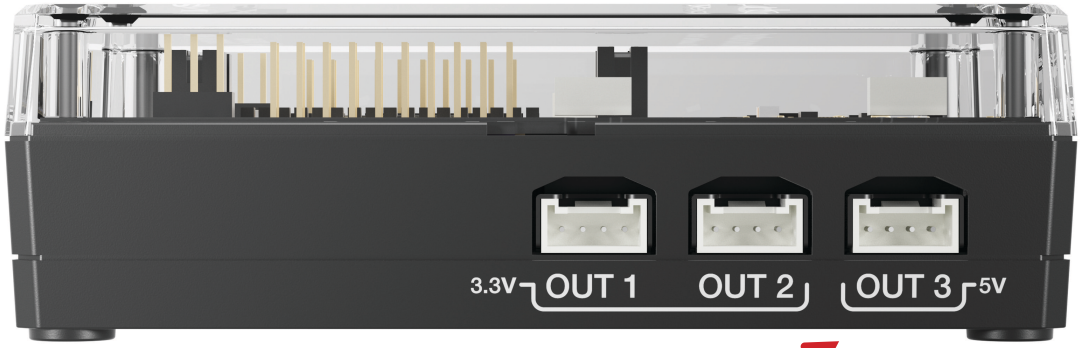


Digital Temperature and Humidity (DHT)

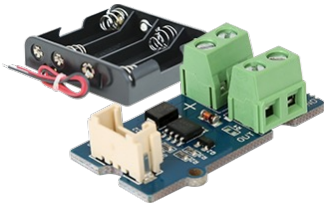
Ranger

Hub from the side – Output ports

Output ports for external motors and other outputs with Grove connectors

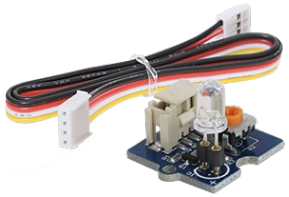


5 Volt port, OUT 3, required for motors



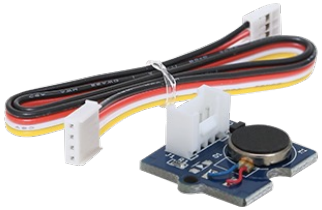
MOSFET controls power level to pump and other devices

Pump



External LED

Continuous Servo Motor



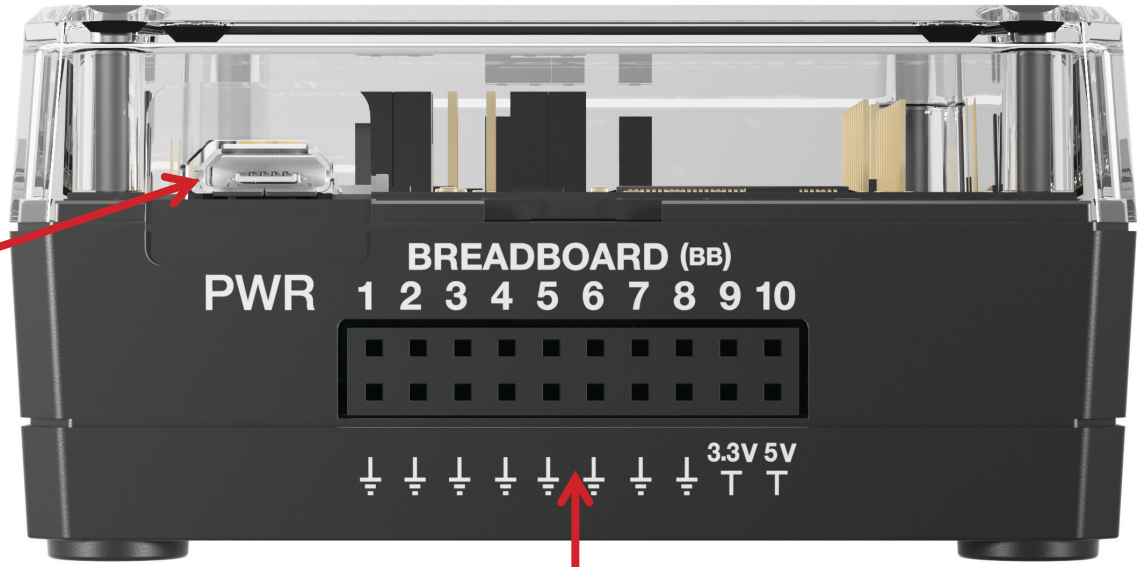
Vibration Motor

Texas Instruments

Hub from the Back – breadboard ports

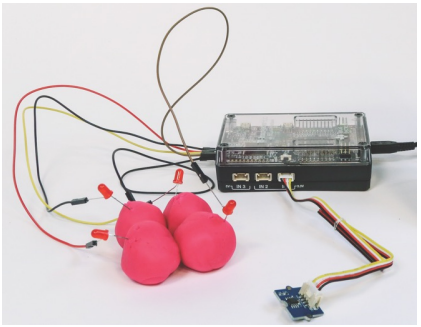
USB Port (micro) connects to external batteries and wall socket.

Connects to computer for updating Hub firmware.



Breadboard ports

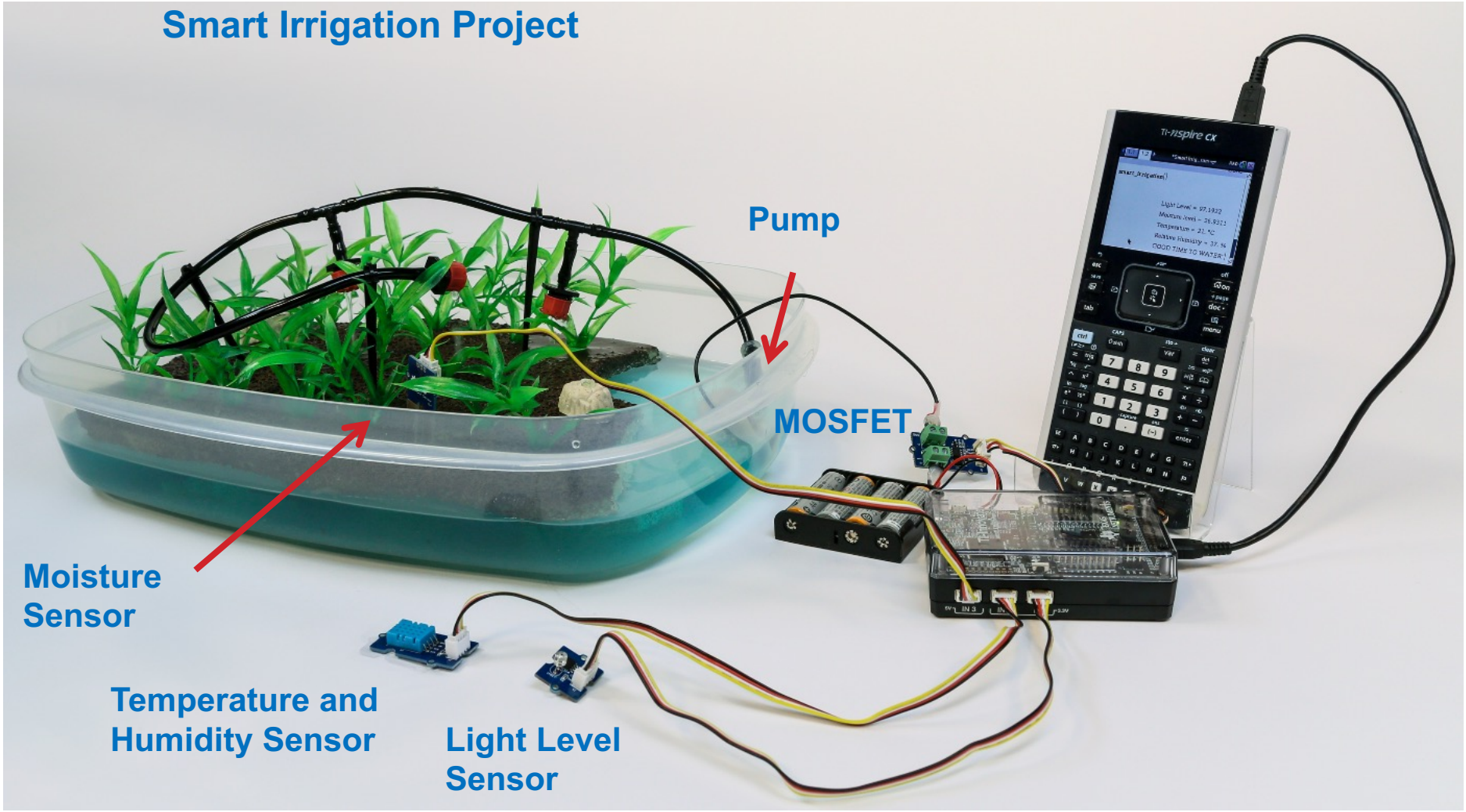
RGB Array



Four-Chambered Heart Project

Path to STEM Projects with breadboard

Putting Inputs and Outputs together in a Smart System



Connecting the Hub to your calculator

Note: The Hub is powered by the calculator. When the cables are connected the Hub goes through a brief "boot-up" process. During the boot-up the RGB LED displays a color that indicates the Hub firmware release, in this case, orange.

1

Plug B side of cable into USB B port of the Hub.

2

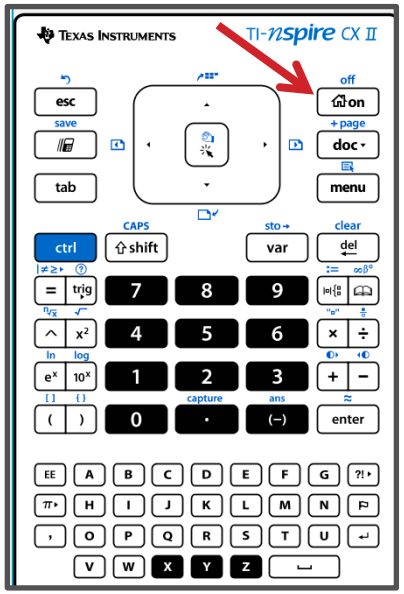
Plug A side of cable into port on calculator.

Unit-to-unit cable

TEXAS INSTRUMENTS

Creating a new TI-Nspire document

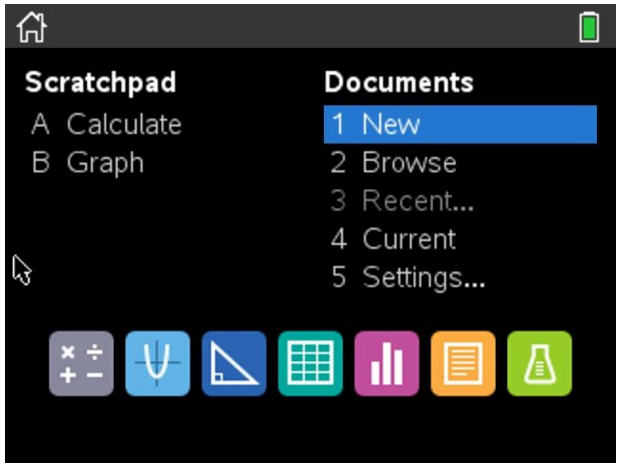
1



Press the **[home/on]** key to display the home screen.

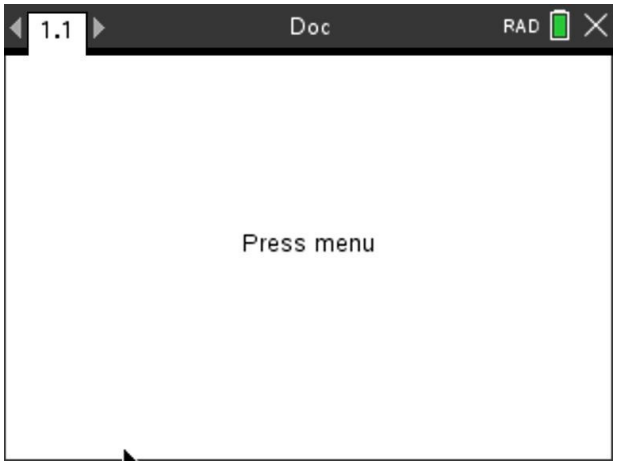
Note: If you have a document open, pressing the **[home/on]** key repeatedly toggles between the home screen and the document.

2



Use **arrow keys** and **[enter]** or Press **[1]** to select 1 New document.

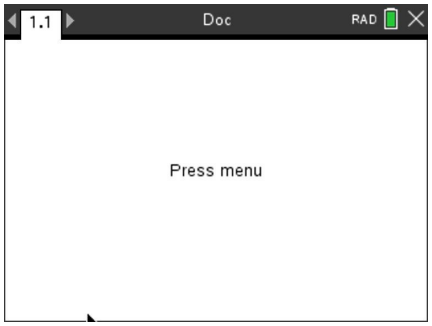
3



See next slide for steps to add a program.

Creating a Hub Program - 1

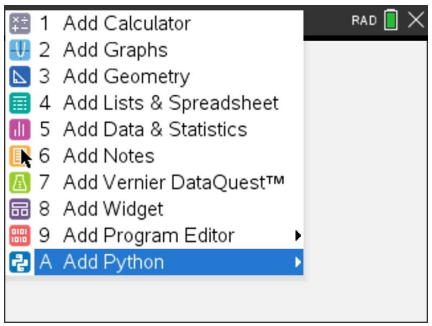
1



Press **[menu]** to bring up a menu of applications to add to the page.

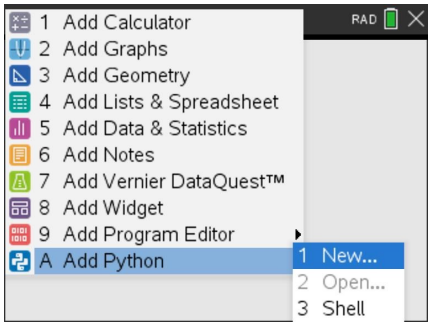
Note: You can also add a new page to the document by pressing **[ctlr] [doc] +page.**

2



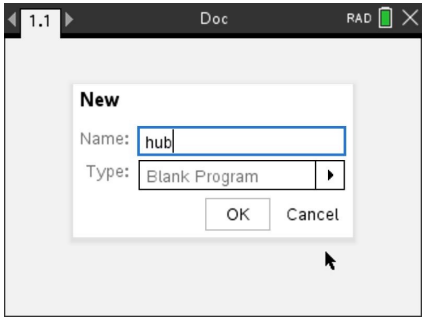
Press **down arrow** repeatedly then press **[enter]** or press **[A]** to select Add Python.

3



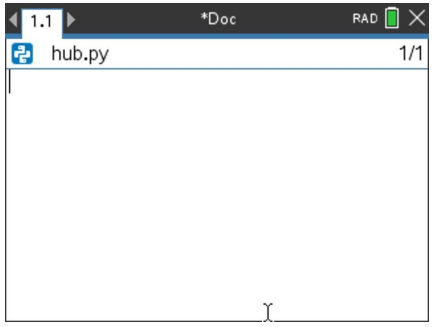
Select 1: New by pressing **[enter]** or **[1]**

4



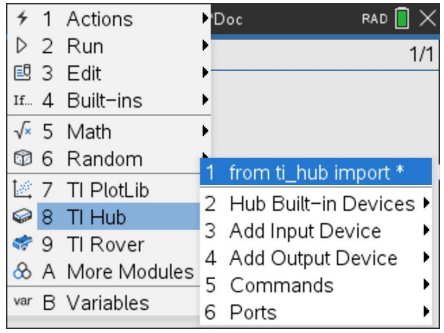
Type your program name and press **[enter]**.

5



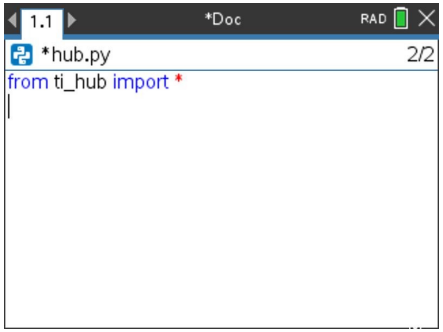
You begin at a blank edit screen.

6



Press **[menu]** then **[8]** TI Hub **[1]** Import ti_hub.

7

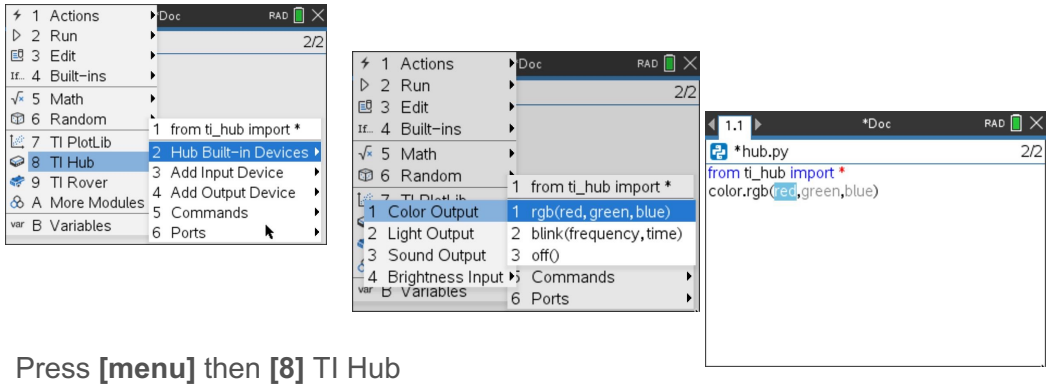


Importing the ti_hub module is required at the beginning of every Hub program.

See step 8 on next page.

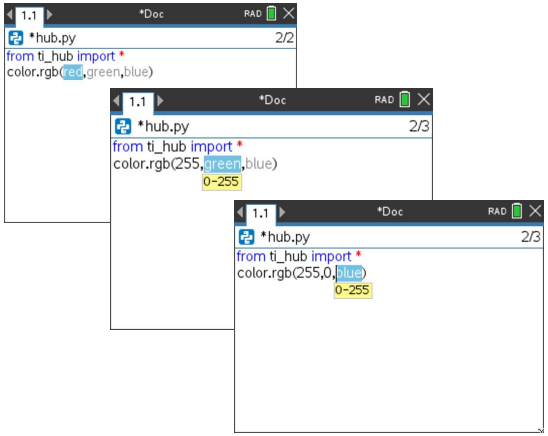
Creating a Hub Program - 2

8



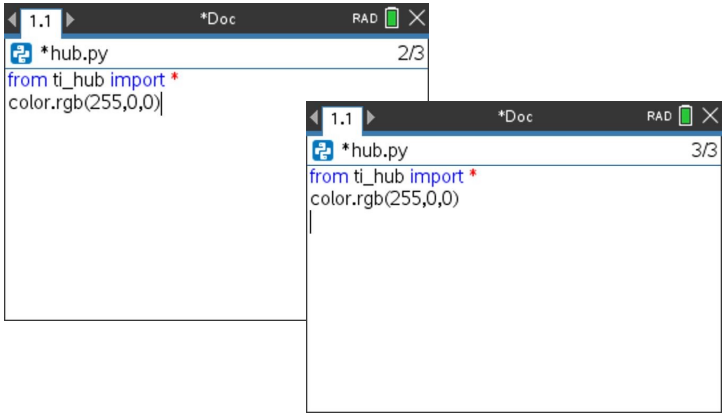
Press **[menu]** then **[8]** TI Hub
[2] Hub Built-in Devices **[1]** Color Output
[1] rgb(red,green,blue) to paste to the edit line.

9



You are prompted to enter values between 0 and 255 for the red, green and blue LED's.
255 is full power and 0 is off.
Press **[tab]** to move from prompt to prompt.

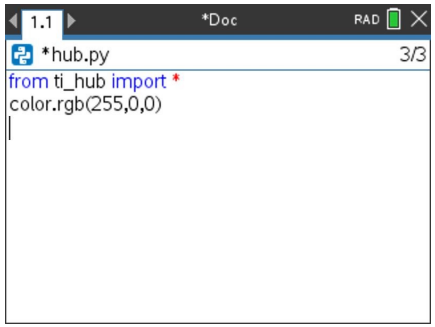
10



Right arrow to the end of the line and press **[enter]** to complete the statement.
A faster approach is to use **[ctrl] [enter]** from any place on a line to complete the statement and move the cursor to the beginning of the line below.
Note: It is important that each statement begin on a new line.

Running a Hub Program

1

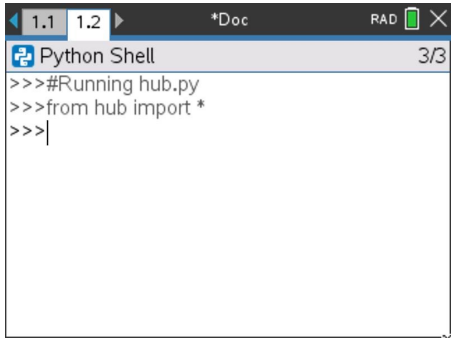


Press **[ctrl] [R]** to run the program from a Python shell on the next page.

Note: **[ctrl] [R]** also checks syntax and stores program changes. **[ctrl] [B] is another option for checking syntax and storing.** * before the program name indicates that changes have not been stored.

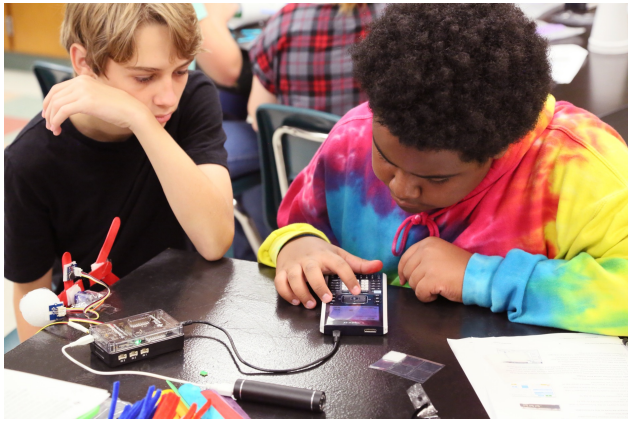
Before running the program make sure that the calculator and the Hub are connected.

2



Your program runs in a Python shell.

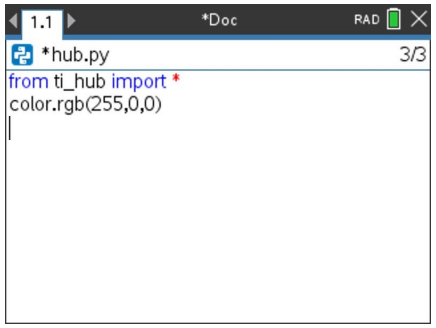
You can re-run the program from the shell by pressing **[ctrl] [R]** again.



Texas Instruments

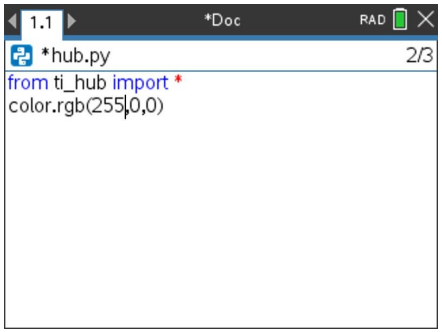
Editing a Hub Program

1



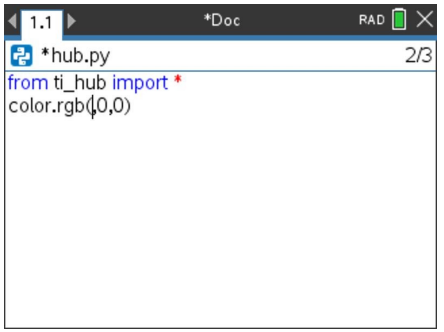
Press **[ctrl] left arrow** to go back to your Python editor page.

2



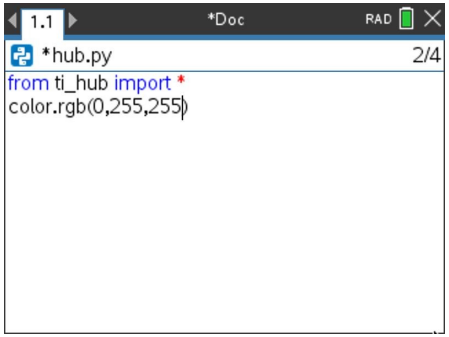
Use the **arrow keys** to position the cursor to change the red value. Put the cursor just after the current value.

3



Press **[del]** repeatedly to backspace over the 255 for the red value.

4

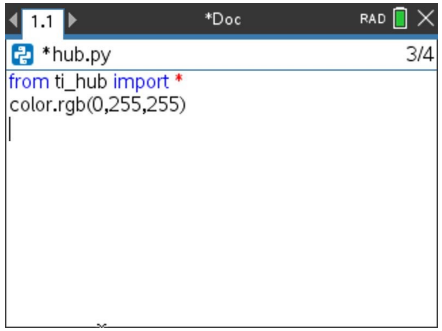


Type in a new value for red, then **right arrow** just past the 0 for green, use **[del]** to backspace over the 0 and type a new value for green, then **right arrow** just past the 0 for blue, use **[del]** to backspace over the 0 and type a new value for blue.

After entering the last change, **right arrow** to the end of the line and press **[enter]** to complete the statement.

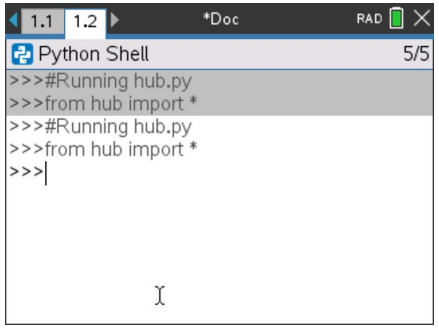
Note: A faster approach is to use **[ctrl] [enter]** from any place on a line to complete the statement and move the cursor to the beginning of the line below.

5



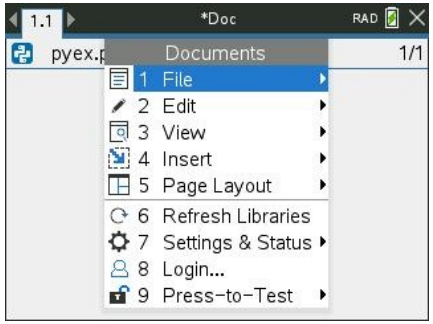
Press **[ctrl] [R]** to run the program again from a Python shell on the next page.

6



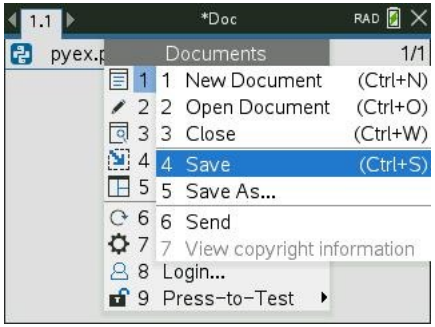
Saving a TI-Nspire document file

1



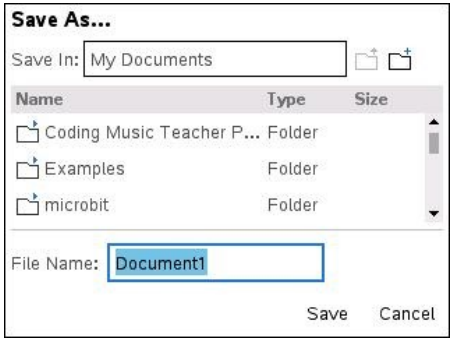
Press **[doc]** then select 1 File from the menu by pressing **[enter] or [1].**

2



Select 4 Save or 5 Save As… from the menu.

3

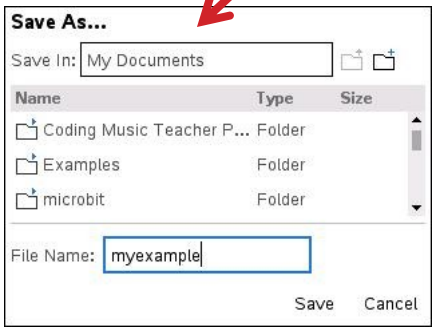


Type in your file name using alpha and numeric characters.

Note: The name must begin with an alpha character.

4

Folder where file will be saved.



Press **[enter]** to save the file to the folder indicated above.

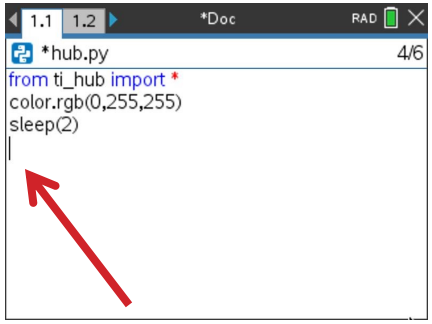
To change the folder press the **[UP]** arrow key and then use **arrows** and **[enter]** to select a folder before pressing **[enter]** to save the file.

Press **[esc]** to cancel the save dialogue.

You can use **[ctrl] [S]** as a shortcut to save the TI-Nspire document file.

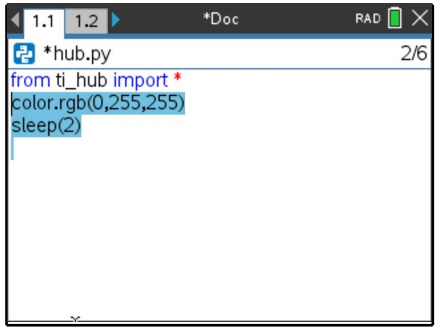
Copying and Pasting a Block of Code

1



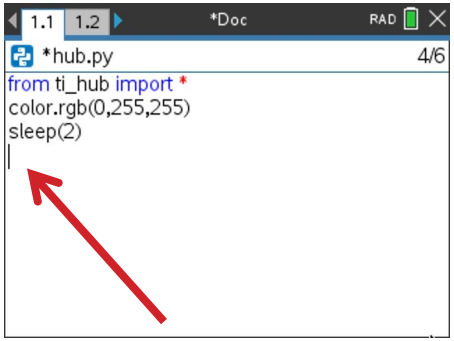
Use **arrow keys** to move the cursor to the beginning of row below the section of code that you want to copy.

2



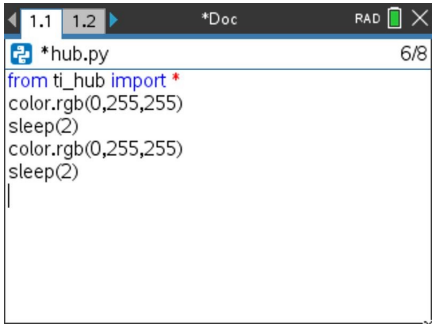
Press and hold **[shift]** then press **UP arrow** repeatedly to highlight the rows to be copied. Press **[ctrl] [C]** to copy the highlighted code.

3



Use **arrow keys** to move the cursor to the location that you want to paste from.

4

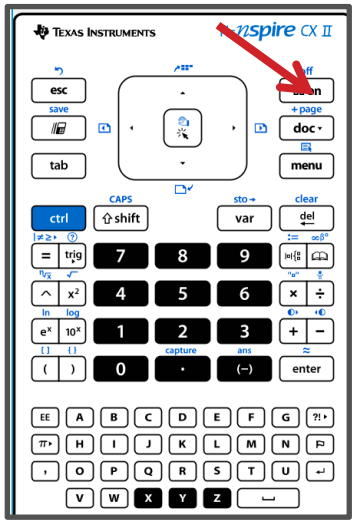


Press **[ctrl] [V]** to paste.

You can paste repeatedly.

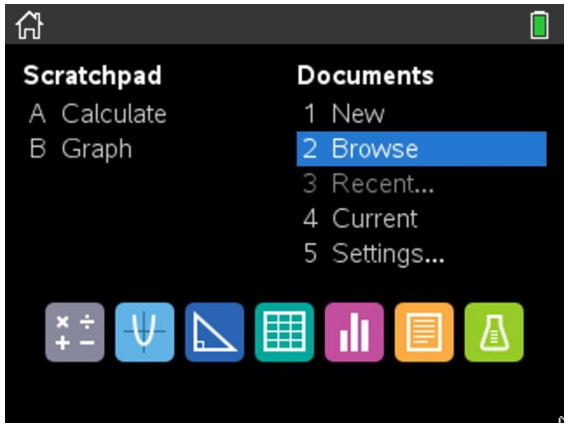
Opening an existing TI-Nspire document file

1



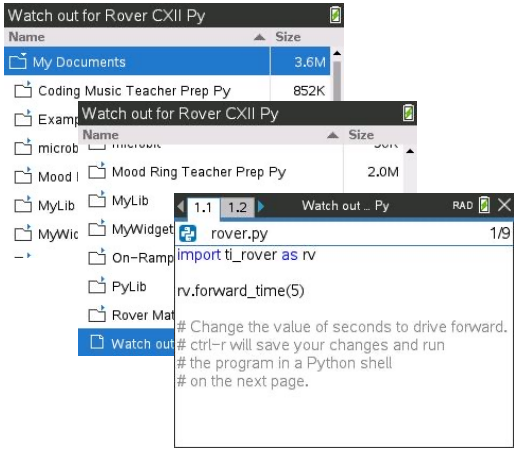
Press the **[home/on]** key to display the home screen.

2



Use **arrow keys** and **[enter]** or Press **[2]** to select 2 Browse files.

3

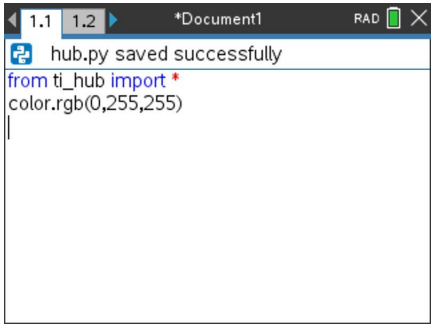


Use **arrow keys** and **[enter]** to select a folder and a file.

Note: Pressing the **[home/on]** key repeatedly toggles between the home screen and the current document.

Copying a Python Program

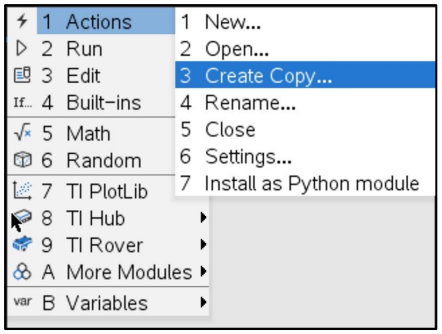
1



Press **[ctrl] [B]** to compile and save your program.

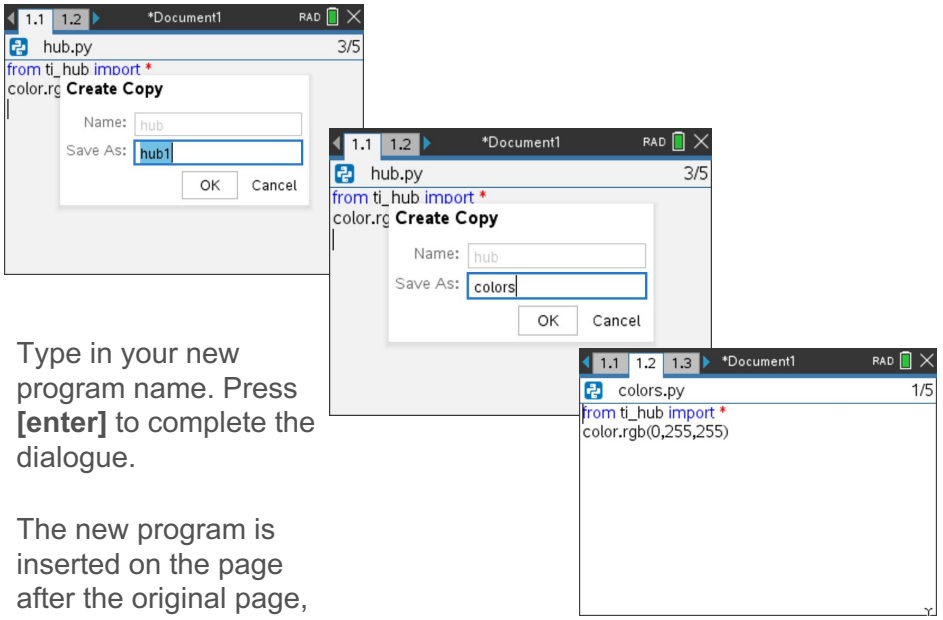
Note: You will not be able to copy the program if you have made changes since using **[ctrl] [R]** or **[ctrl] [B]**.

2



Press **[menu] [1]** Actions **[3]** Create Copy…

3

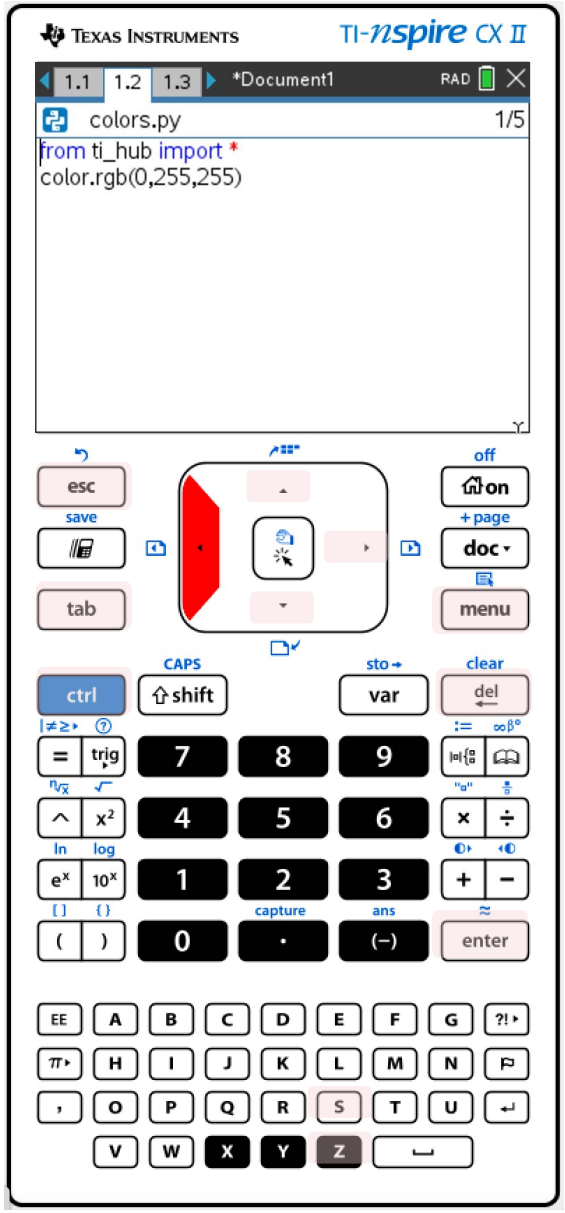


Type in your new program name. Press **[enter]** to complete the dialogue.

The new program is inserted on the page after the original page, in this case page 1.2.

Entry and Edit Tips

- Use **number key shortcuts** or **arrow keys** and **[enter]** to select from menus
- Use **[esc]** to back out of a menu or a dialogue.
- Use **[enter]** to complete a dialogue.
- Use **[tab]** to move to the next input when entering a function
- Use **arrow keys** to move the cursor around the screen
- Use **[del]** as a destructive backspace
- Use **[ctrl] [enter]** to complete a statement and move to the next line
- Use **[ctrl] [Z]** to undo an action
- Use **[ctrl] [S]** to save your file
- Use **[ctrl] [left arrow]** and **[ctrl] [right arrow]** to move from page to page
- Use **[menu]** to see options for the current application.



Texas Instruments

COLOR OUTPUTS

Set the color

New Program:

Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

Task: Set the color output of the RGB LED.

Each color takes a value of (0-255).

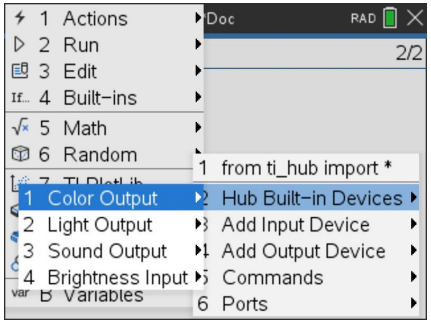
Challenge Tasks:

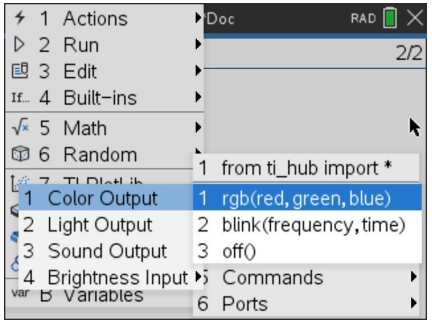
Try to make Yellow

Try to make Cyan

Try to make Magenta

Find the color_rgb( ) function on the Hub Built-in Devices, Color Output menu.





Texas Instruments

Create and Name a Color

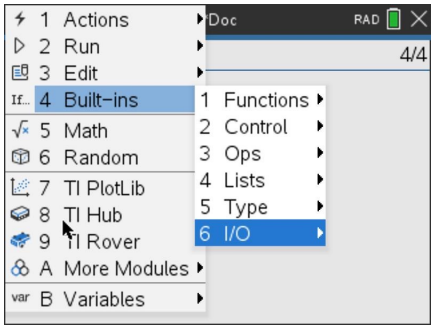
Add to previous Program:

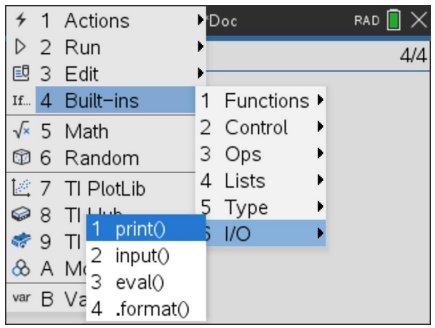
Task: Create your own color and give it a name.

Challenge Tasks:

Print the name of your color.

Find the print() function on the Hub Built-in Device I/O (Input/Output) menu. Print your message as a text string by enclosing the characters in quotes. Add quote marks by pressing [ctrl] [x] (multiply symbol).





Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

Extra

Display a series of colors.

Add to previous program:

Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

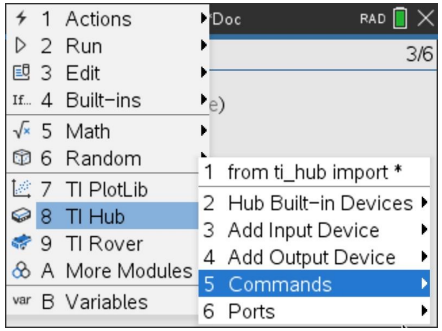
Task: Display a sequence of colors for 2 seconds each.

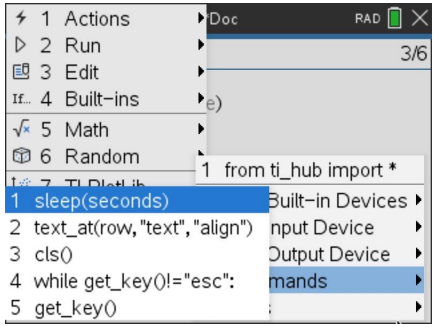
Challenge Tasks:

Try to have your LED match the pattern of a traffic light.

Try to have your LED turn off at the end of the sequence.

sleep() causes the program to wait the number of seconds entered before moving to the next statement.
Find the sleep() function on the Hub Commands menu.





Texas Instruments

Extra

Turn the LED ON and OFF

New Program:

Task: Set the RGB LED to a color then keep ON for 2 seconds then turn the LED off for 2 seconds.

Challenge Tasks:

Try to blink (turn on and turn off) 4 times

Try to blink 4 times in 8 seconds

What values for the red, green and blue inputs to the color.rgb() function will turn the LED off?
See below for steps to copy and paste a block of code.

Use **arrow keys** to move cursor to row below the section you want to copy

Press and hold **[shift]**, press **UP arrow** to highlight

[ctrl] [C] to copy, move cursor to location to paste from, **[ctrl] [V]** to paste.

Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

Extra

Blink the LED Repeatedly

New Program:

Task: Set the RGB LED to a color and then blink 10 times
(ON for 2 seconds then OFF for 2 seconds)

Challenge Tasks:
Try to blink 30 times in a minute
Try to blink 60 times in a minute

1.1 1.2 1.3 *Doc RAD

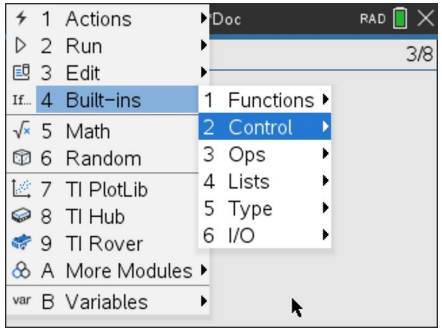
*blinkloop.py 10/10

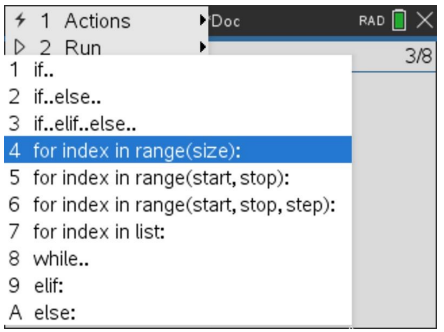
```
from ti_hub import *

#The loop repeats the statements in the
#indented block 10 times
for n in range(10):
    color.rgb(red,green,blue)
    sleep(2)
    color.rgb(0,0,0)
    sleep(2)
```

Enter *n* as the loop counter variable and 10 as the value for the loop size. *n* starts with a value of 0. Each time the loop completes *n* is increased by 1. If *n* is less than loop size value looping continues, otherwise looping stops and the program moves to the next statement after the block.

Find for index in range(size): on the Built-ins Control menu.





Texas Instruments

SOUND OUTPUTS

Play a Sound Tone

New Program:

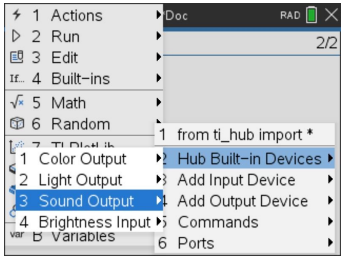
Task: Play a sound tone by entering a value for frequency (sound vibrations per second) and a value for time in seconds to play the tone.

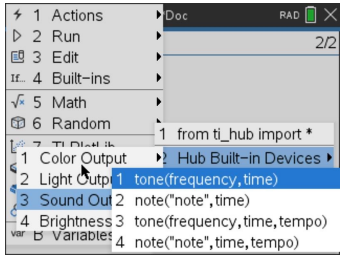
Note: Human voices tend to be in the 85 to 255 Hertz (vibrations per second) range.

What is the lowest tone that you can hear?

What is the highest tone that you can hear?

Find the sound.tone( ) function on the Hub Built-in Devices, sound Output menu.

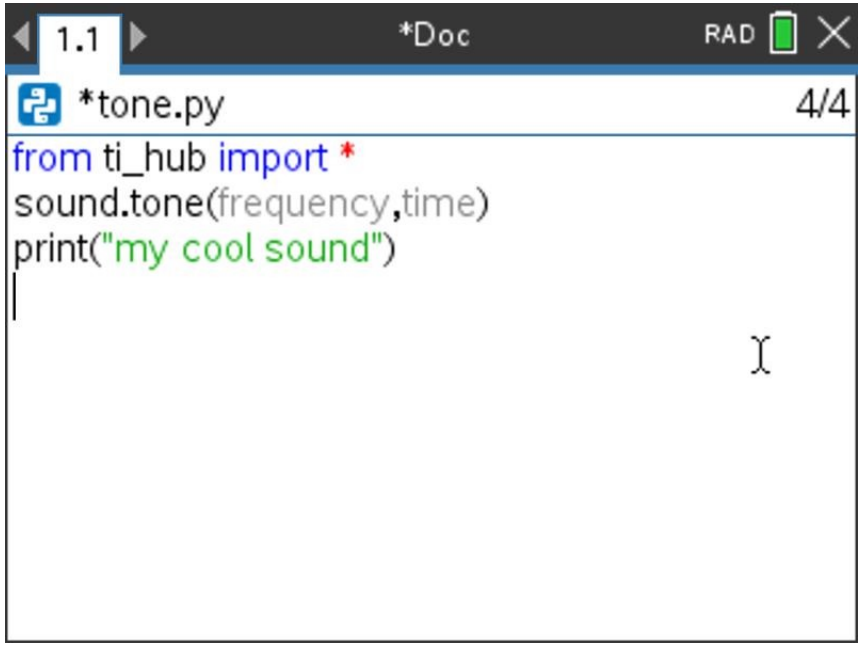




Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

Find your Favorite Sound Tone

Add to previous Program:



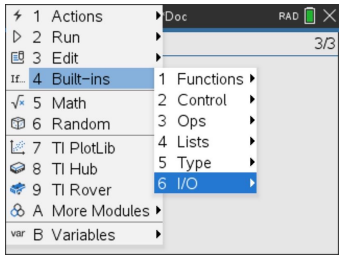
Task: Play a sound tone by entering a value for frequency (sound vibrations per second) and a value for time in seconds to play the tone.

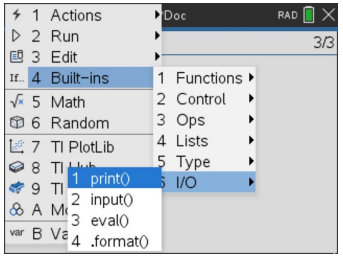
Find your favorite frequency.

Challenge Tasks:

Give your frequency a name and print the name.

Find the sound.tone( ) function on the Hub Built-in Devices, sound Output menu.
Find the print() function on the Hub Built-in Device I/O (Input/Output) menu. Print your message as a text string by enclosing the characters in quotes. Add quote marks by pressing **[ctrl] [x] (multiply symbol)**.

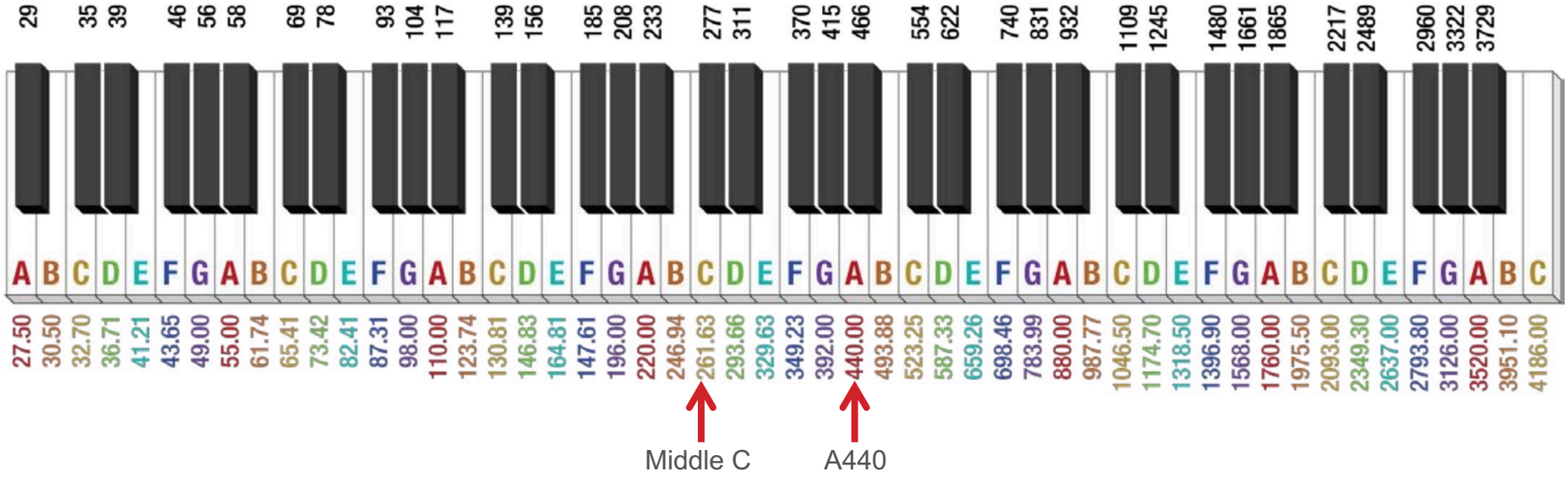




Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

Sound Frequencies and Musical Notes

Extra



Sound tone frequencies map to musical notes.
Middle C (C4) on the piano keyboard has a frequency of 261.6 Hertz.
A440 (A4) is used by orchestras for tuning.

See Coding the Sounds of Music project for an in-depth treatment of digital music with the Hub. https://education.ti.com/en/activities/stem-projects/making-music-with-code

Play a Musical Note

Extra

New Program:

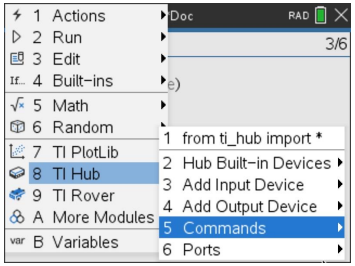
Task: Enter and play the doorbell tones using the sound.note() function.

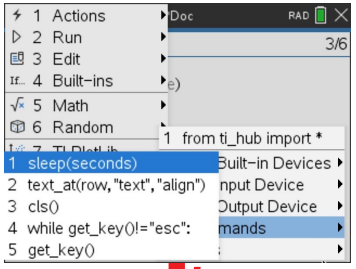
Create your own doorbell tone.

Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

It is important to pause the program with a sleep() function between sounds. This allows the sound to play completely before the program starts the next sound. (Try playing multiple sounds without sleep() functions.)
If you want to have a short silence between sounds add an additional .1 seconds to the sleep function, sleep(.5+.1) or insert an additional sleep(.1) between sounds.

Find the sleep() function on the Hub Commands menu.





Texas Instruments

Play the Notes of an Octave

Extra

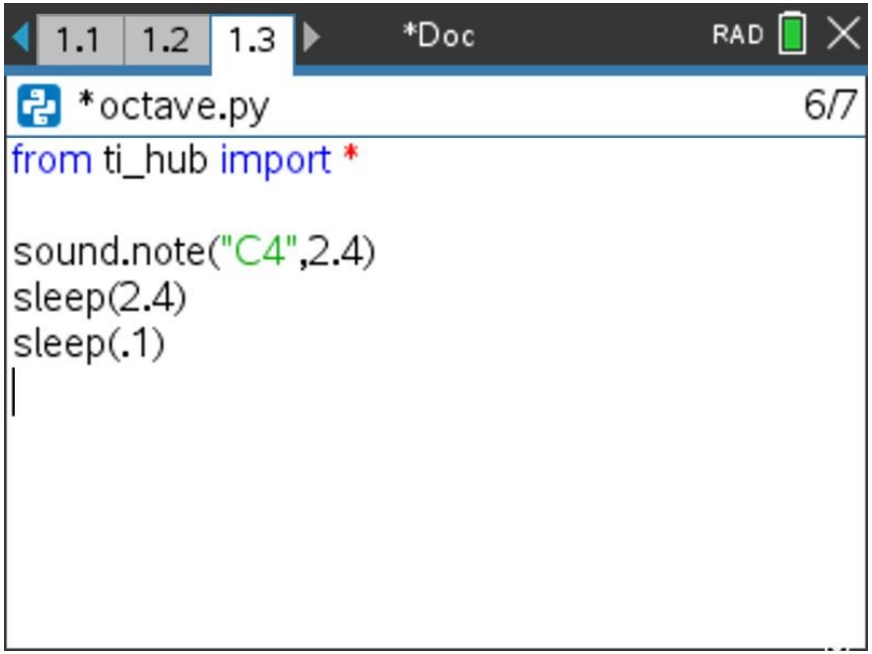
New Program:

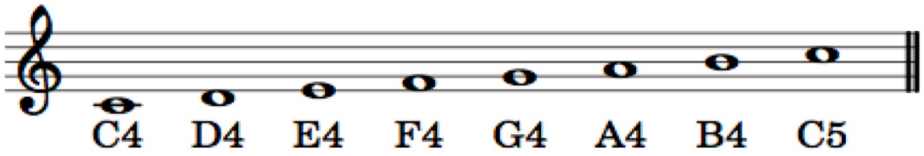
Task: Write a program to play each note of Do-Re-Mi-Fa-Sol-La-Si-Do as whole notes.

This is an entire octave.

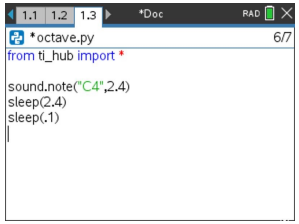
Recall at 100 BPM a whole note lasts for 2.4 seconds. The first note Do is "C4" and the last note Do is "C5".

Include a .1 second rest between notes.

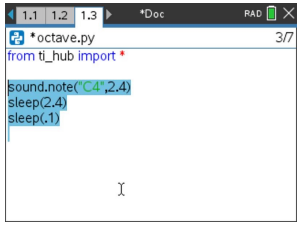




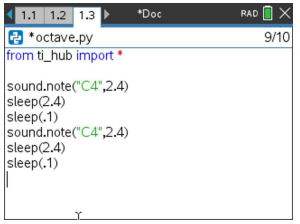
The steps for copying and pasting blocks of code may be helpful.



Use **arrow keys** to move cursor to row below the section you want to copy.



Press and hold **[shift]**, press **UP arrow** to highlight.



[ctrl] [C] to copy, move cursor to location to paste from, **[ctrl] [V]** to paste.

Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

BRIGHTNESS (LIGHT LEVEL) INPUTS

Measure Brightness

New Program:

```
1.1 1.2 *Doc RAD
*bright.py 7/8
from ti_hub import *

while get_key() != "esc":
    b=brightness.measurement()
    print("brightness= ",b)
    sleep(.25)
```

Brightness sensor

Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

Task: Enter and run the program to measure brightness.

What is the light level in your room?

Try shining a light on the brightness sensor.

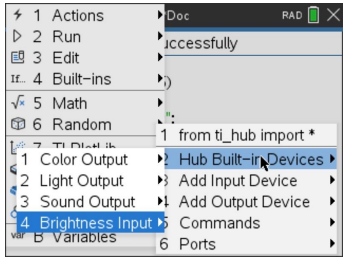
Try covering the brightness sensor.

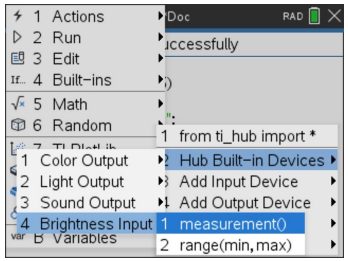
Find while get_key() != "esc": on the Hub Commands menu. This sets up a loop that will run until the escape key is pressed.

Find the brightness.measurement() function on the Hub Built-in Devices, Brightness Input menu.
Store the result to a variable named b.
Start your statements with b=, then add brightness.measurement().

Find print( ) on the Built-ins I/O (Inputs/Outputs) menu.

.





Texas Instruments

Extra

Control an RGB LED with Brightness Measurements

Add to previous Program:

```
1.1   *Doc   RAD
bright.py saved successfully
from ti_hub import *
brightness.range(0,255)

while get_key() != "esc":
    b=brightness.measurement()
    print("brightness= ",b)
    color.rgb(red,green,blue)
    sleep(.25)
```

Task: Add brightness.range(0,255) to the program to set brightness measurements to be from 0 to 255 instead of 0 to 100.

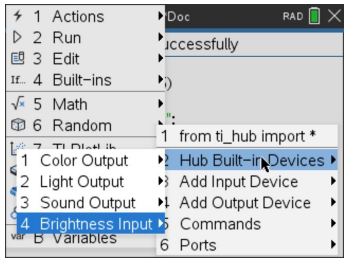
Use the brightness values stored in variable b as inputs for some or all of the color.rgb(red,green, blue) inputs.

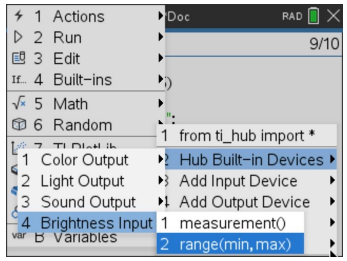
Try shining a light onto the Brightness sensor.

Challenge Task:

Try to make a night light that sets the color.rgb() to gets brighter as the light level dims.

Find the brightness.range() function on the Hub Built-in Devices, Brightness Input menu. If you do not set the range in your program the brightness values will be 0 to 100.





Press **[menu]** key to see Python Program Editor options.
Press **[tab]** key to move to the next input in a function.
Press **[crtl] [enter]** to complete a statement and move to the next line.
Press **[ctrl] [R]** to run the program from a Python shell on the next page.
Use **[ctrl] left** to move from the shell page back to the Python editor page.

Thank You

www.TIstemProjects.com

Contact stem-team@ti.com with questions

Texas Instruments