

Meet the TI-Innovator Hub

TI-84 Plus CE

Python

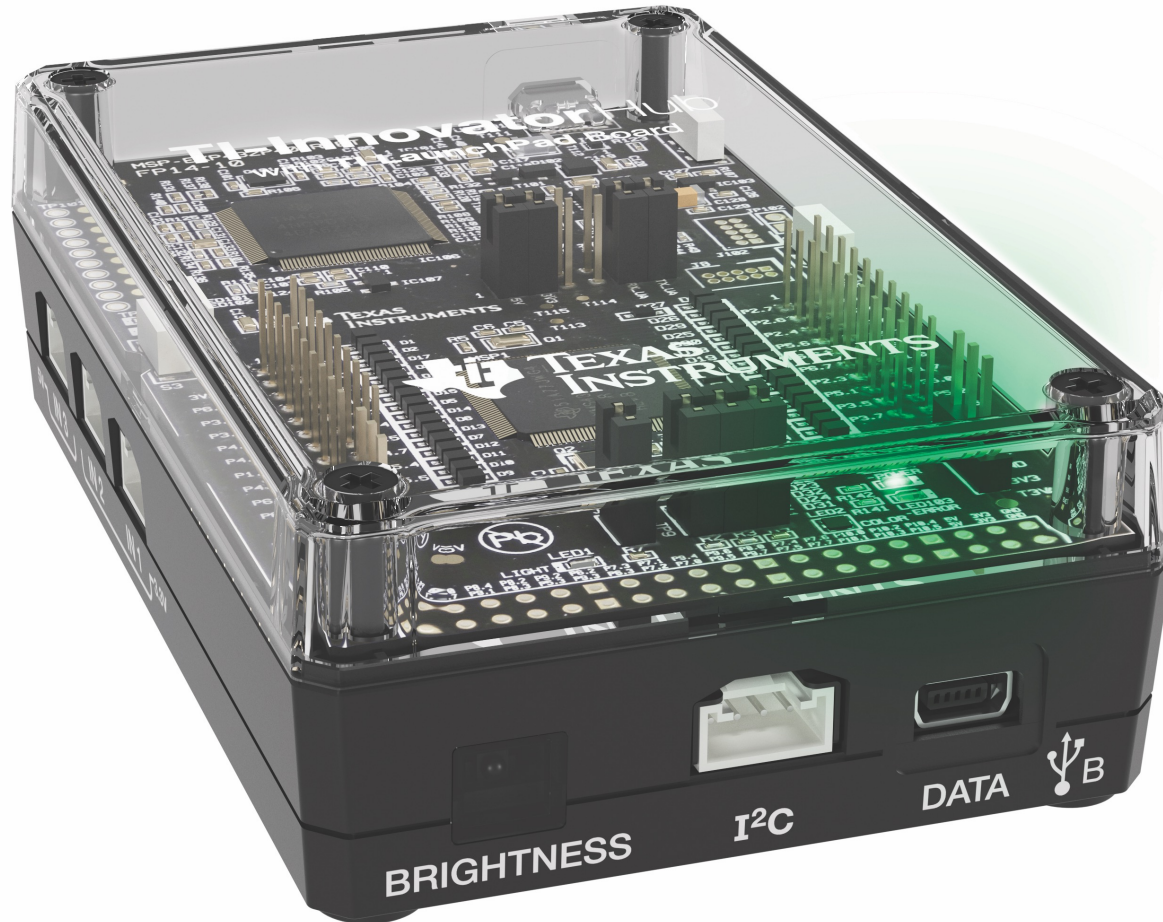
Texas Instruments

@ticalculators



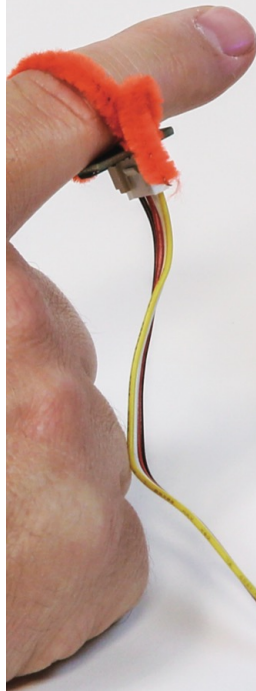
www.TIstemProjects.com

Meet the TI-Innovator Hub

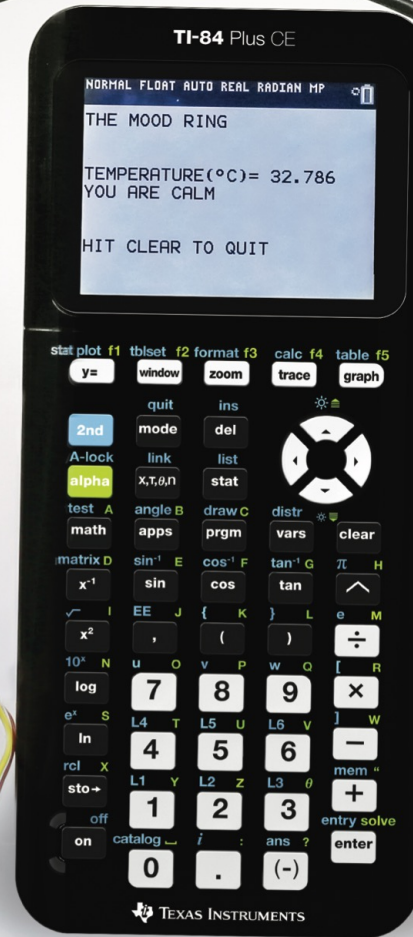


TI-Innovator™
Hub

Built-in
RGB LED
Output

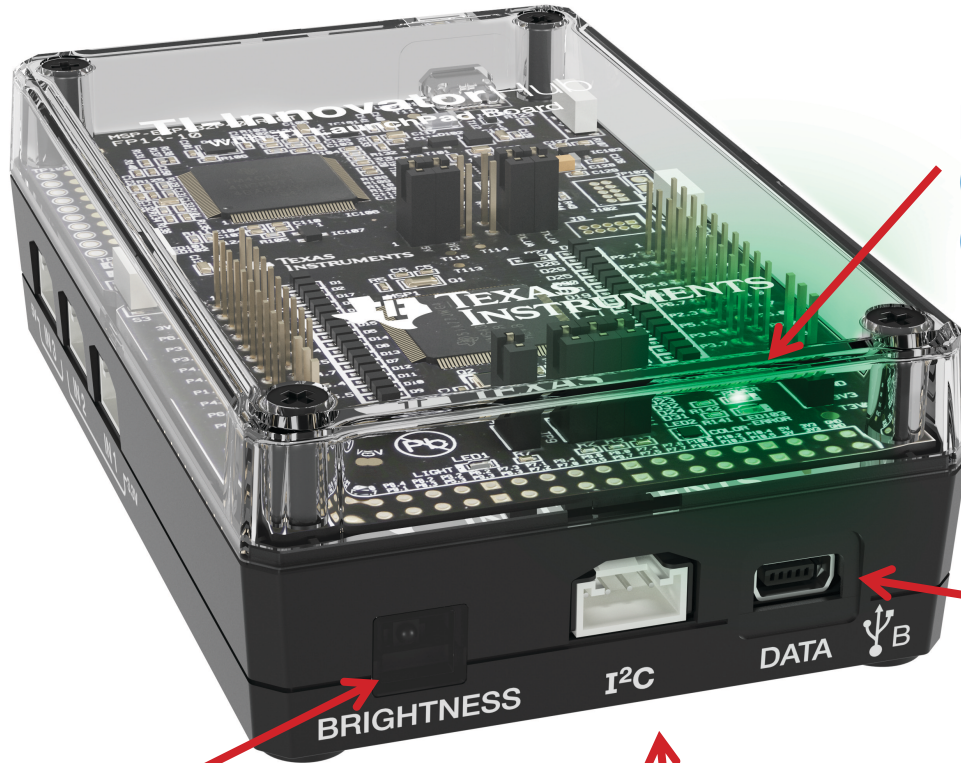


External Sensor Input



TI Graphing
Calculator

Hub from the Front



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

USB Port (mini) to connect to calculator and computer

I2C Port used to connect to Rover

Brightness Sensor (Built-in)

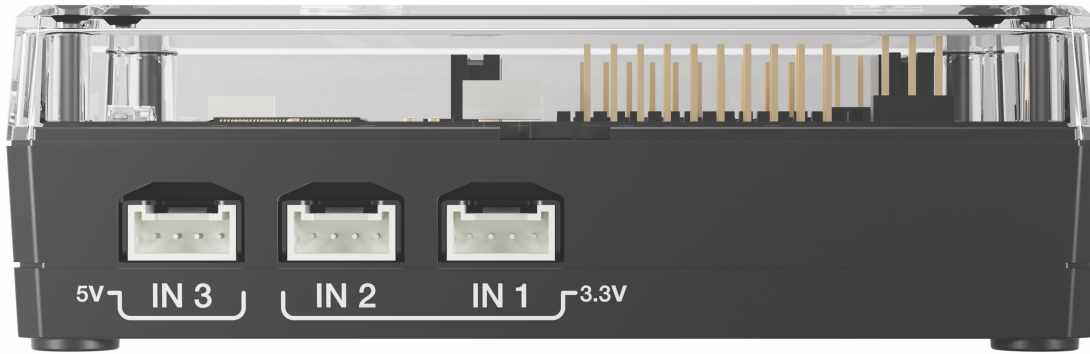
Hub from the Bottom



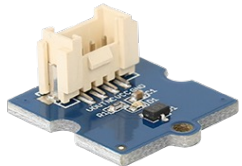
**Speaker
(Built-in)**

Hub from the side – input ports

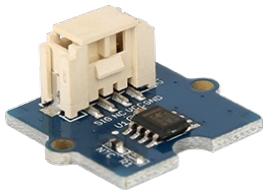
Input ports for external sensors with Grove connectors



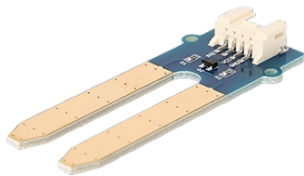
3.3 Volt ports, IN 1 and IN 2, required for Ranger and DHT



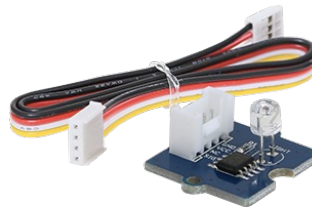
Magnetic Field (Hall Effect)



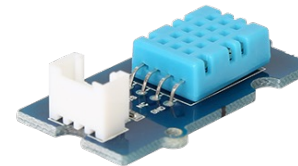
Temperature



Moisture



Light Level



Digital Temperature and Humidity (DHT)



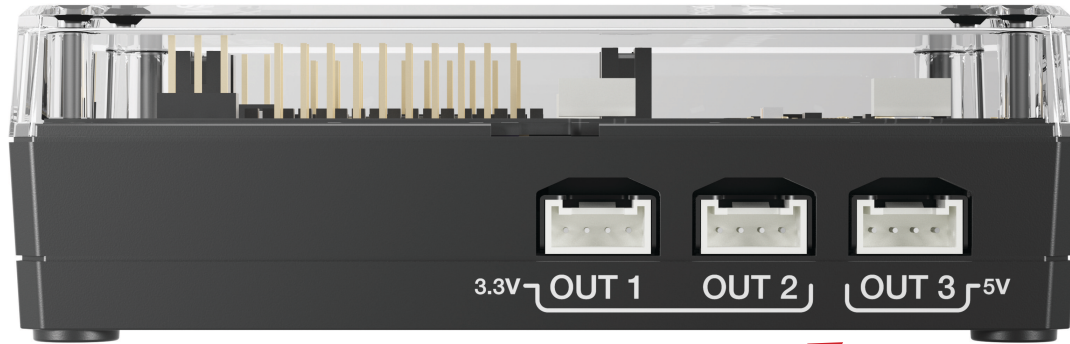
Ranger



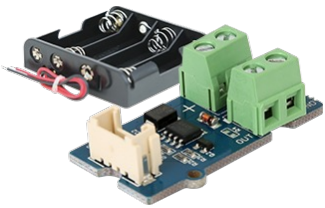
Vernier SensorLink Adaptor

Hub from the side – Output ports

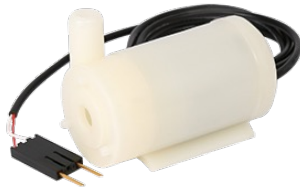
Output ports for external motors and other outputs with Grove connectors



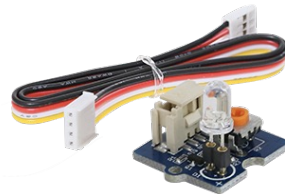
5 Volt port, OUT 3, required for motors



MOSFET controls power level to pump and other devices



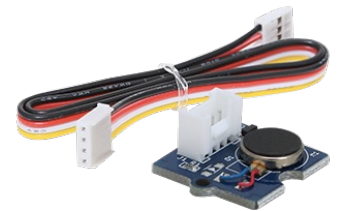
Pump



External LED



Continuous Servo Motor

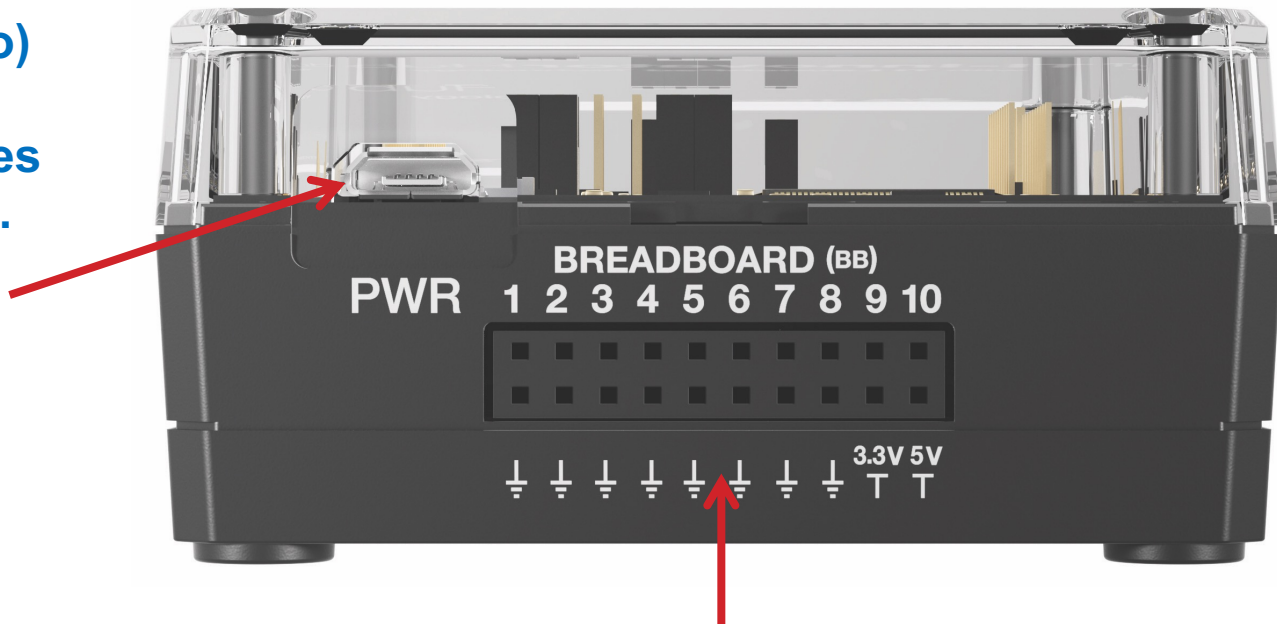


Vibration Motor

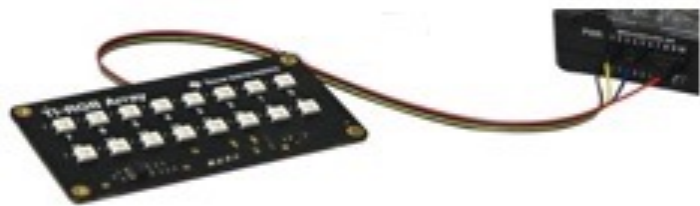
Hub from the Back – breadboard ports

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

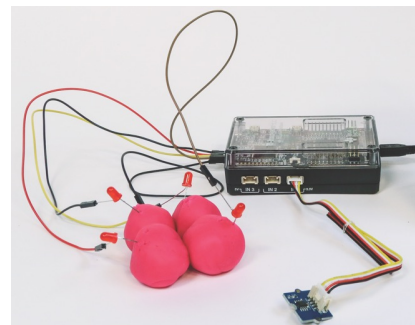
Connects to computer for updating Hub firmware.



Breadboard ports



RGB Array



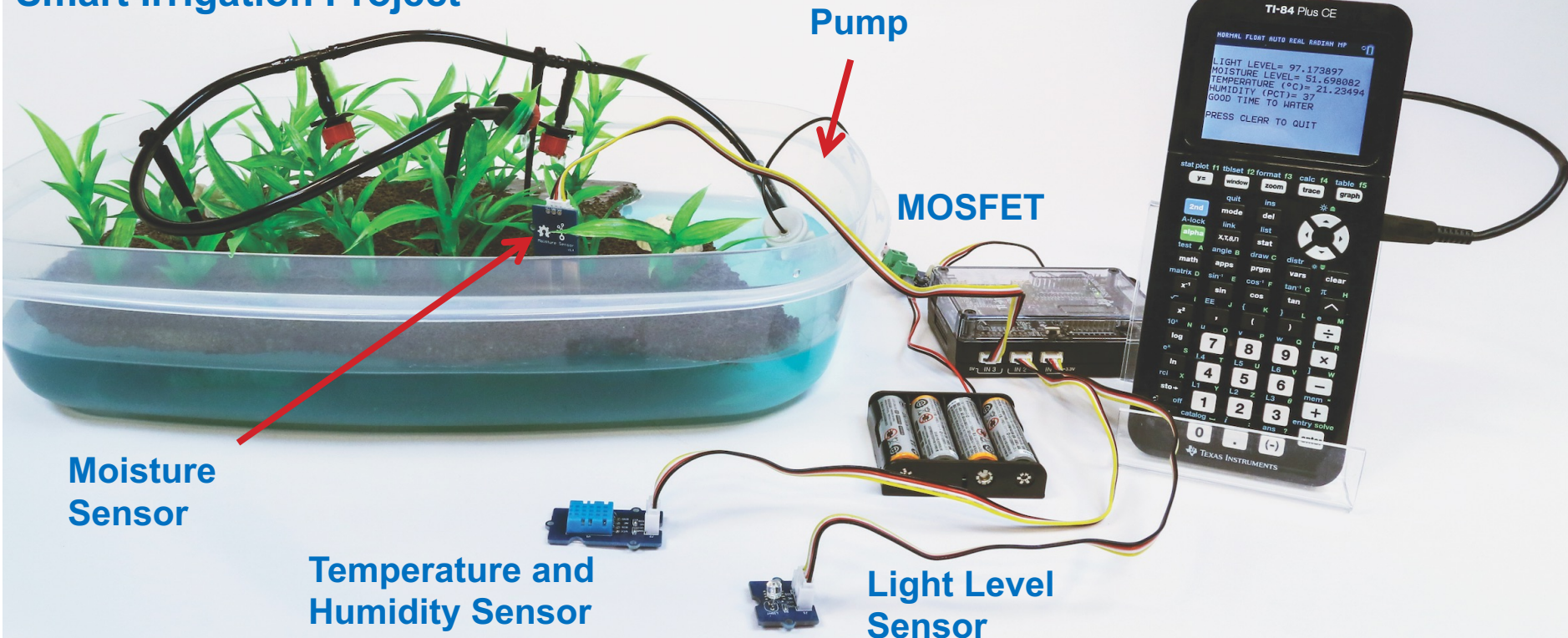
Four-Chambered Heart Project



Path to STEM Projects with breadboard

Putting Inputs and Outputs together in a Smart System

Smart Irrigation Project



Pump

MOSFET

Moisture Sensor

Temperature and Humidity Sensor

Light Level Sensor

Connecting the Hub to your calculator



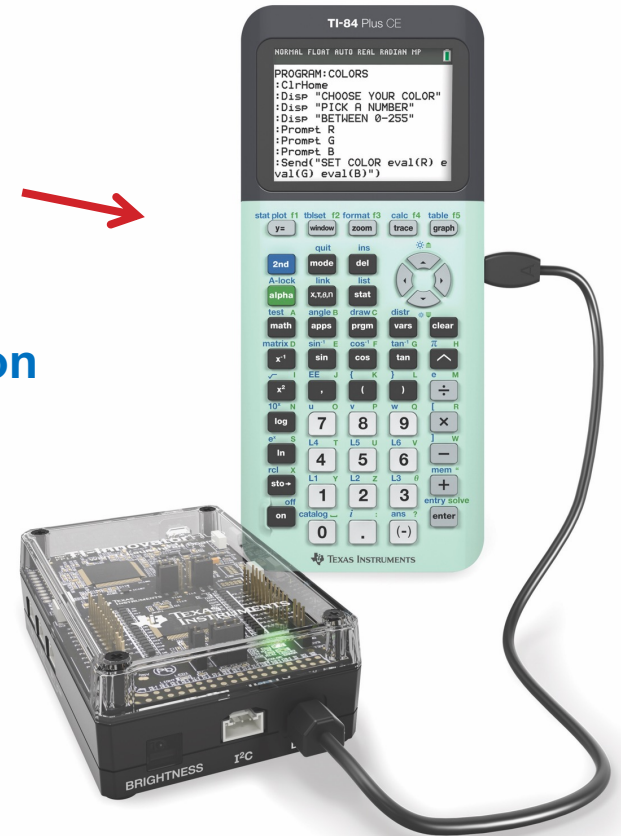
Note: The Hub is powered by the calculator. When the cables are connected the Hub goes through a brief “boot-up” process. During the boot-up the RGB LED displays a color that indicates the Hub firmware release, in this case, orange.

1
Plug B side of cable into USB B port of the Hub.

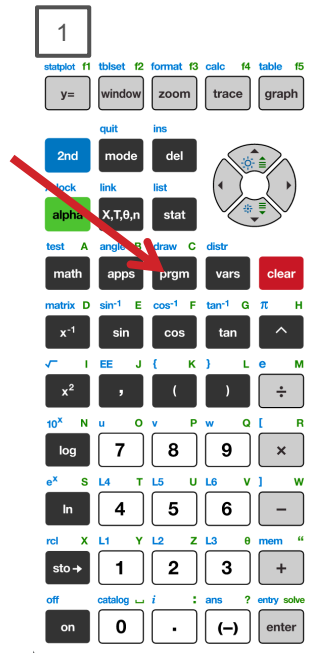
2
Plug A side of cable into port on calculator.



Unit-to-unit cable



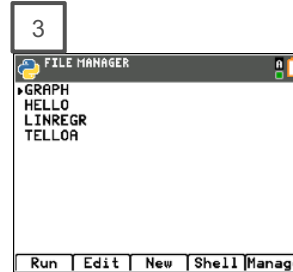
Creating a new Python Program



Press the [prgm] key to create, edit and execute TI-Python programs.

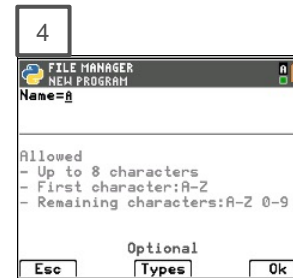


Press **down arrow** [enter] or press [2] to select 2: Python App

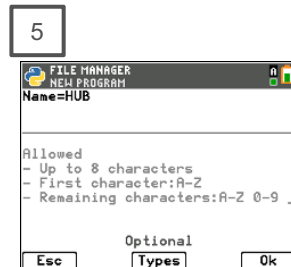


You have the option to run, edit, create or manage programs.

Press [New] softkey (zoom button)



You are prompted to enter a program name. The blinking A cursor shows that you are in alpha entry mode. The green alpha labels on the keys are active.



Type your program name and press [Ok].



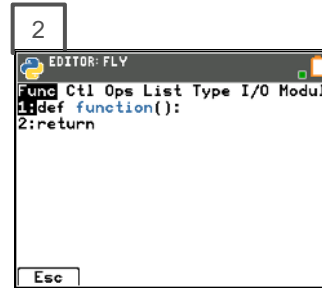
You are now in position to begin entering statements to your program.

Entering a TI-Innovator Hub Program – 1

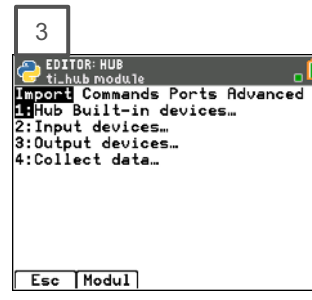
importing the ti_hub module



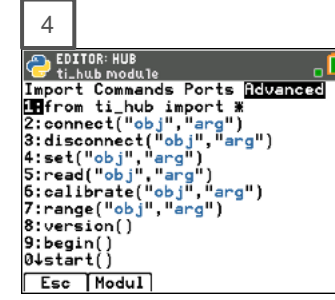
The Python program editor uses an insert cursor and a backspace delete. Press **[Fns...]** softkey to see functions to use in your program.



Press **right arrow** repeatedly or **left arrow** to move to the Modul menu.



You will see a menu of installed modules available to use functions from. Select **6:ti_hub...**



Press **right arrow** repeatedly or **left arrow** to move to the Advanced menu.



The `ti_hub` module import statement is pasted to your program. The `ti_hub` import statement brings in the functions and statements available on the `ti_hub` Commands and Advanced menus.

You will include additional import statements for inputs and outputs in your programs.

You are now ready to add statements to complete your program.

See the next slide.

Entering a TI-Innovator Hub Program - 2

1

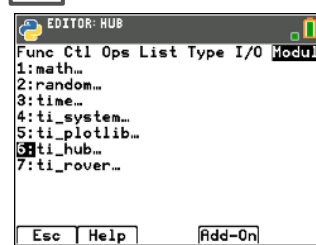


```
EDITOR: DRIVE
PROGRAM LINE 0002
import ti_rover as rv
_
```

Fns... | a A # | Tools | Run | Files

You are now ready to import modules for specific inputs and outputs. Navigate to the ti_hub menu by pressing [Fns...] then arrow to the Modul menu.

2



```
EDITOR: HUB
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
```

Esc | Help | Add-On

Then select **6:ti_hub...** to see options.

3

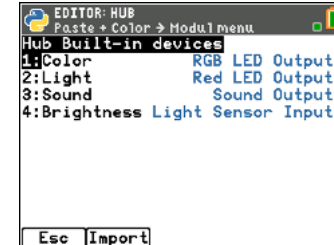


```
EDITOR: HUB
ti_hub module
Import Commands Ports Advanced
1:Hub Built-in devices...
2:Input devices...
3:Output devices...
4:Collect data...
```

Esc | Modul

Our first program will be to control the built-in Red-Green-Blue color LED. Select **1:Hub Built-in devices...**

4



```
EDITOR: HUB
Paste + Color -> Modul menu
Hub Built-in devices
1:Color RGB LED Output
2:Light Red LED Output
3:Sound Sound Output
4:Brightness Light Sensor Input
```

Esc | Import

Select **1:Color** to import the Color RGB LED module.

5

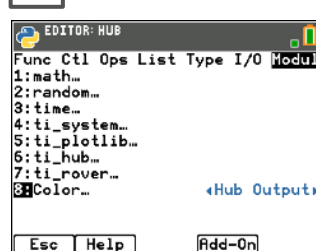


```
EDITOR: HUB
PROGRAM LINE 0003
from ti_hub import *
import color
```

Fns... | a A # | Tools | Run | Files

The import color statement brings the functions to control the Color RGB LED.

6



```
EDITOR: HUB
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Color... <Hub Output>
```

Esc | Help | Add-On

Navigate to the color menu by pressing [Fns...] then arrow to the Modul menu. You will now see a menu item for **Color...**

Select the **Color...** menu item to see the functions available for the Color RGB LED.

7

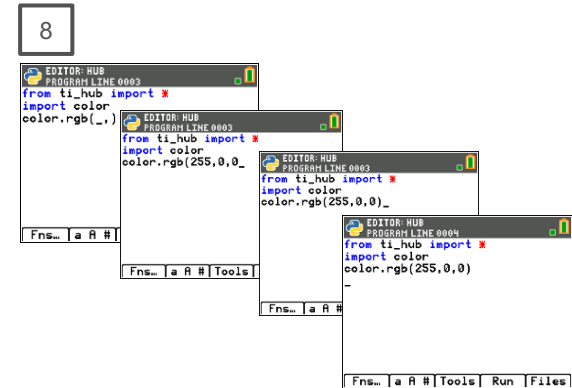


```
EDITOR: HUB
Color
1:rgb(r,g,b) 0-255
2:blink(freq,time)
3:off()
```

Esc | Modul

Select **1:rgb(r,g,b)** to paste the function into your editor.

8



```
EDITOR: HUB
PROGRAM LINE 0003
from ti_hub import *
import color
color.rgb(255,0,0)

EDITOR: HUB
PROGRAM LINE 0003
import color
color.rgb(255,0,0)

EDITOR: HUB
PROGRAM LINE 0004
from ti_hub import *
import color
color.rgb(255,0,0)
```

Fns... | a A # | Tools | Run | Files

Your cursor is positioned just before the first comma ready to enter the input value for red. Values range from 0 (off) to 255 (full power). To display red **type 255** then **right arrow** to position the cursor just before the second comma. **Type 0** for the value of green. The **right arrow** and **type 0** for the value for blue.

Right arrow to the end of the line and press [enter] to complete the statement.

A faster approach is to use [2nd] [enter] from any place on a line to complete the statement and move the cursor to the beginning of a blank line below.

Note: It is important that each statement begin on a new line.

Running a Hub Program

1

```
EDITOR: HUB
PROGRAM LINE 0004
from ti_hub import *
import color
color.rgb(255,0,0)
_
```

Fns... a A # Tools Run Files

Press **[Run]** (trace key) to run the program from the Python shell.

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

2

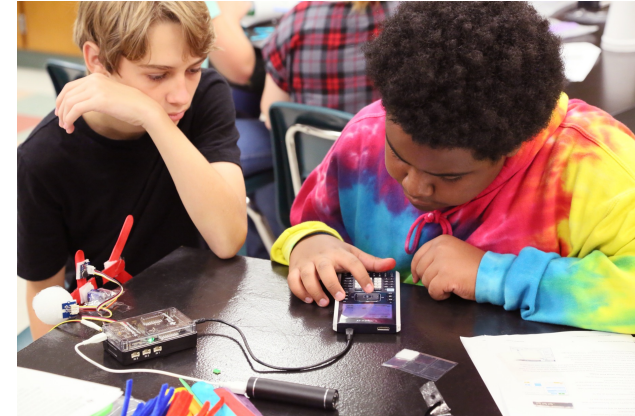
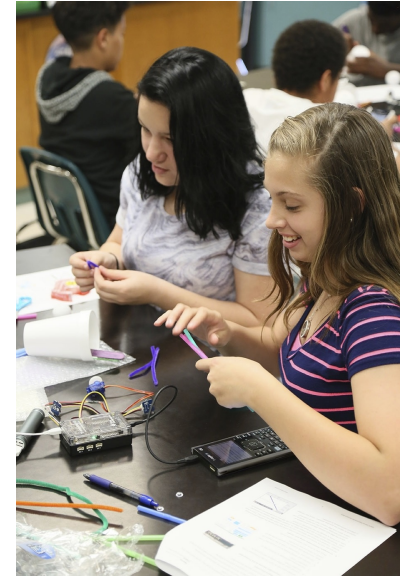
```
PYTHON SHELL

>>> # Shell Reinitialized
>>> # Running HUB
>>> from HUB import *
>>> |
```

Fns... a A # Tools Editor Files

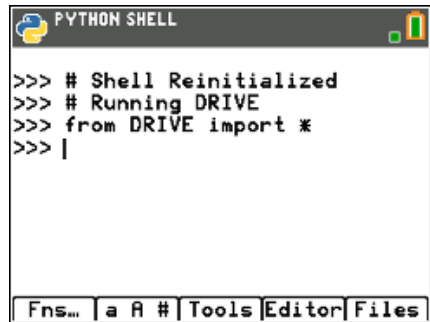
Your program runs in a Python shell.

You can re-run the program from the shell by pressing **[Tools]** (zoom key) then select **1:Rerun Last Program** from the menu.



Editing a Hub Program

1



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running DRIVE
>>> from DRIVE import *
>>> |
```

Press **[Editor]** to go back to your Python editor page.

2



```
EDITOR: HUB
PROGRAM LINE 0004
from ti_hub import *
import color
color.rgb(255,0,0)
-
```

To display cyan instead of red, use the **arrow keys** to position the cursor just after the red input of 255.

3



```
EDITOR: HUB
PROGRAM LINE 0003
from ti_hub import *
import color
color.rgb(255_0,0)
```

Press **[del]** to backspace over 255.


4



```
EDITOR: HUB
PROGRAM LINE 0003
from ti_hub import *
import color
color.rgb(0,255_0)
```

Type 0 for the red value. **Right arrow** to just after the 0 of the green input. Press **[del]** to backspace over 0. **Type 255** for the green input.

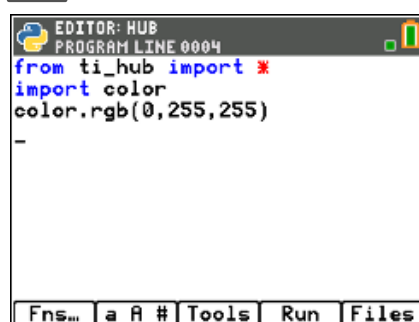
5



```
EDITOR: HUB
PROGRAM LINE 0003
from ti_hub import *
import color
color.rgb(0,255,255_
```

Right arrow to just after the 0 for the blue input. Press **[del]** to backspace over 0. **Type 255** for the blue input.

6

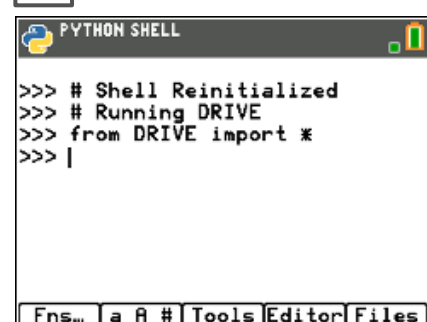


```
EDITOR: HUB
PROGRAM LINE 0004
from ti_hub import *
import color
color.rgb(0,255,255)
-
```

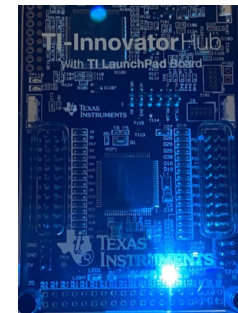
Press **[2nd] [enter]** to complete the statement.

Press **[Run]** to run the program in the Python shell.

7



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running DRIVE
>>> from DRIVE import *
>>> |
```



TI-Innovator Hub Module Menu

Import

```
EDITOR: AAA
ti_hub module
Import Commands Ports Advanced
1:Hub Built-in devices...
2:Input devices...
3:Output devices...
4:Collect data...
Esc Modul
```

```
EDITOR: AAA
Paste + Color → Modul menu
Hub Built-in devices
1:Color RGB LED Output
2:Light Red LED Output
3:Sound Sound Output
4:Brightness Light Sensor Input
Esc Import
```

```
EDITOR: AAA
PROGRAM LINE 0005
from ti_hub import *
import color
import sound
import brightns
Fns... a R # Tools Run Files
```

Commands

```
EDITOR: AAA
ti_system module
Import Commands Ports Advanced
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align")
4:disp_clr() clear text screen
5:disp_wait() [clear]
6:disp_cursor() 0=off 1=on
7:while not escape(): [clear]
Esc Modul
```

Ports

```
EDITOR: AAA
ti_hub module
Import Commands Ports Advanced
1:OUT 1
2:OUT 2
3:OUT 3
4:IN 1
5:IN 2
6:IN 3
7:BB 1
8:BB 2
9:BB 3
0:BB 4
A:BB 5
B:BB 6
C:BB 7
D:BB 8
E:BB 9
F:BB 10
G:I2C
```

Advanced

```
EDITOR: AAA
ti_hub module
Import Commands Ports Advanced
1:from ti_hub import *
2:connect("obj","arg")
3:disconnect("obj","arg")
4:set("obj","arg")
5:read("obj","arg")
6:calibrate("obj","arg")
7:range("obj","arg")
8:version()
9:begin()
0:start()
A:about()
B:isti()
C:what()
D:who()
E:last_error()
F:sleep(seconds)
```

Color Menus*

```
EDITOR: AAA
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Color... <Hub Output>
9:Sound... <Hub Output>
0:Brightness... <Hub Input>
Esc Help Add-On
```

```
EDITOR: AAA
Color
1:rgb(r,g,b) 0-255
2:blink(freq,time)
3:off()
Esc Modul
```

* Importing a Hub object module adds to the Modul menu.

Sound Menus*

```
EDITOR: AAA
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Color... <Hub Output>
9:Sound... <Hub Output>
0:Brightness... <Hub Input>
Esc Help Add-On
```

```
EDITOR: AAA
Sound
1:tone(freq,time)
2:note("string",time)
3:tone(freq,time,tempo)
4:note("string",time,tempo)
Esc Modul
```

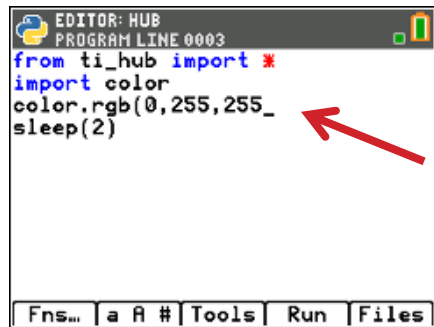
Brightness Menus*

```
EDITOR: AAA
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Color... <Hub Output>
9:Sound... <Hub Output>
0:Brightness... <Hub Input>
Esc Help Add-On
```

```
EDITOR: AAA
Brightness
1:measurement()
2:range(min,max)
Esc Modul
```


Copying and Pasting a Line of Code

1

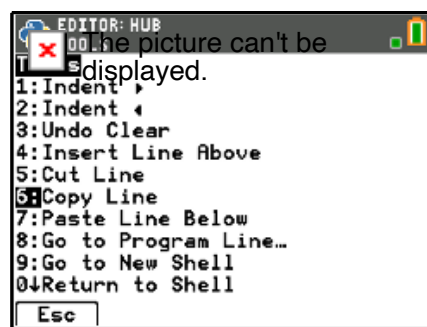


```
EDITOR: HUB
PROGRAM LINE 0003
from ti_hub import *
import color
color.rgb(0,255,255_
sleep(2)
```

Fns... | a A # | Tools | Run | Files

Use **arrow keys** to move the cursor to a position anywhere on the line that you would like to copy.

2



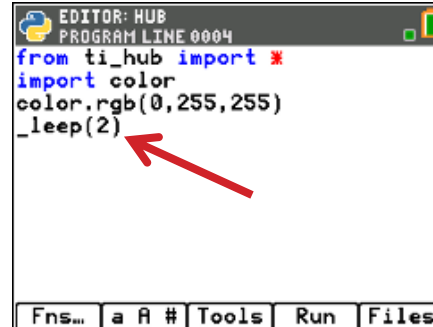
```
EDITOR: HUB
PROGRAM LINE 0004
The picture can't be
displayed.
1:Indent >
2:Indent <
3:Undo Clear
4:Insert Line Above
5:Cut Line
6:Copy Line
7:Paste Line Below
8:Go to Program Line...
9:Go to New Shell
0↓Return to Shell
```

Esc

Press **[Tools]** then select **6:Copy Line** from the menu.

After you select you will be returned to the editor.

3

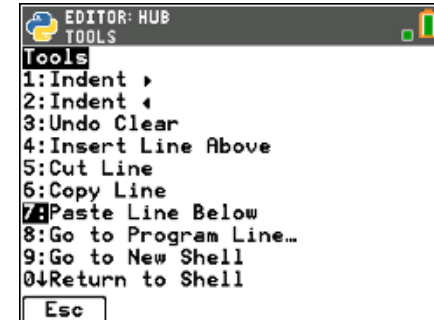


```
EDITOR: HUB
PROGRAM LINE 0004
from ti_hub import *
import color
color.rgb(0,255,255)
sleep(2)
```

Fns... | a A # | Tools | Run | Files

Use **arrow keys** to move the cursor to any location on the line above where you would like to insert the copied line.

4

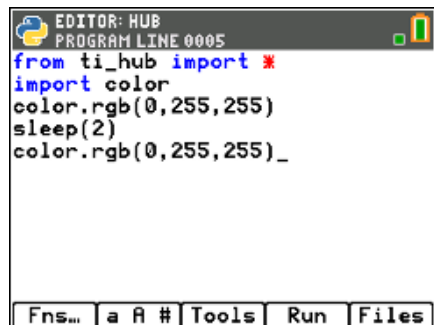


```
EDITOR: HUB
TOOLS
Tools
1:Indent >
2:Indent <
3:Undo Clear
4:Insert Line Above
5:Cut Line
6:Copy Line
7:Paste Line Below
8:Go to Program Line...
9:Go to New Shell
0↓Return to Shell
```

Esc

Press **[Tools]** then select **7:Paste Line Below** from the menu. The copied line will be pasted.

5

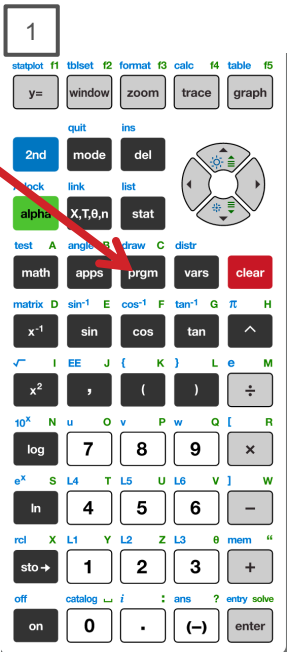


```
EDITOR: HUB
PROGRAM LINE 0005
from ti_hub import *
import color
color.rgb(0,255,255)
sleep(2)
color.rgb(0,255,255)_
```

Fns... | a A # | Tools | Run | Files

You can paste again by returning to the **[Tools]** menu and selecting **7:Paste Line Below**.

Opening an existing Python Program File



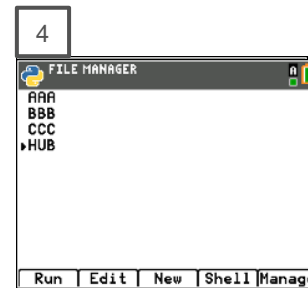
Press the **[prgm]** key to create, edit and execute TI-Python programs.



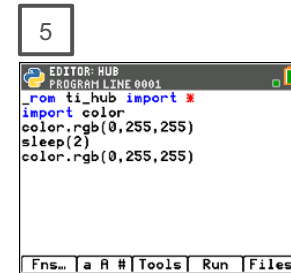
Press **[enter]** or
Press **[2]** to
select 2: Python App



To edit an existing
program, use the **Up**
and **Down Arrow** keys
to select a program.



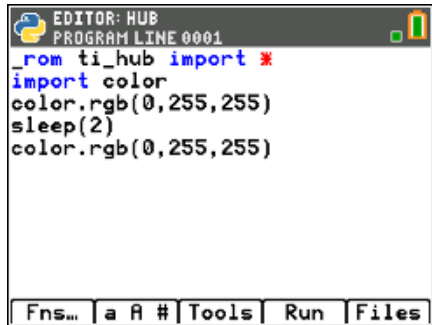
Press **[Edit]** to open
with Python Editor with
the selected program.



You can now make
changes to the
program or run the
program.

Copying/Replicating a Python Program File

1

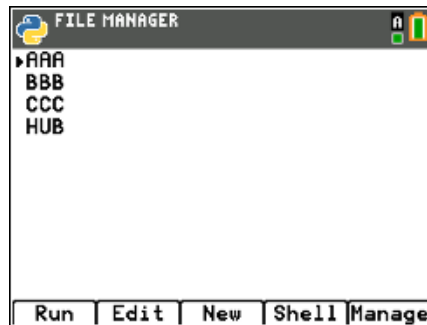


```
EDITOR: HUB
PROGRAM LINE 0001
_rom ti_hub import *
import color
color.rgb(0,255,255)
sleep(2)
color.rgb(0,255,255)
```

Fns... | a A # | Tools | Run | Files

Press **[Files]** to return to the file management screen.

2

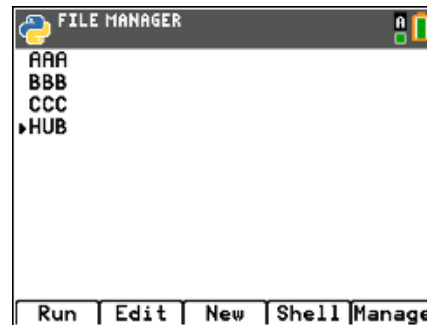


```
FILE MANAGER
AAA
BBB
CCC
HUB
```

Run | Edit | New | Shell | Manage

Use the **Up and Down Arrow** keys to select a program.

3

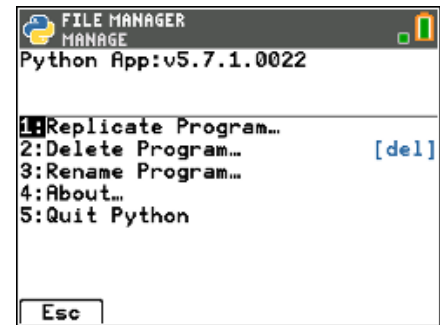


```
FILE MANAGER
MANAGE
AAA
BBB
CCC
HUB
```

Run | Edit | New | Shell | Manage

Press **[Manage]** to see file options.

4

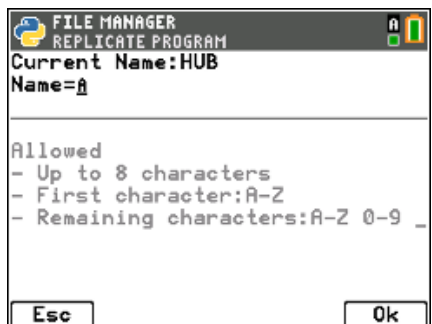


```
FILE MANAGER
MANAGE
Python App:v5.7.1.0022
1:Replicate Program...
2:Delete Program... [del]
3:Rename Program...
4:About...
5:Quit Python
```

Esc

Select **1:Replicate Program** to receive a prompt.

5



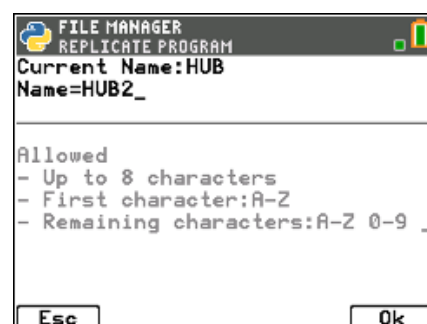
```
FILE MANAGER
REPLICATE PROGRAM
Current Name:HUB
Name=_
```

Allowed
- Up to 8 characters
- First character:A-Z
- Remaining characters:A-Z 0-9 _

Esc | Ok

Type in the name of the new program using the green alpha key labels.

6



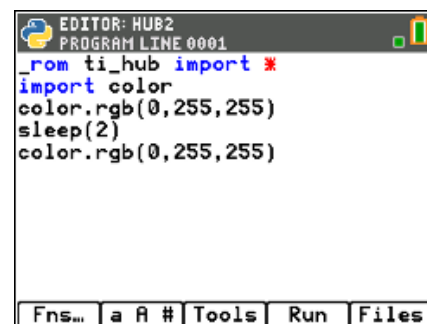
```
FILE MANAGER
REPLICATE PROGRAM
Current Name:HUB
Name=HUB2_
```

Allowed
- Up to 8 characters
- First character:A-Z
- Remaining characters:A-Z 0-9 _

Esc | Ok

To use a number in the name, exit alpha mode by pressing **[2nd] [alpha]** then a **number** key.
Press **[Ok]** to finish the dialogue.

7



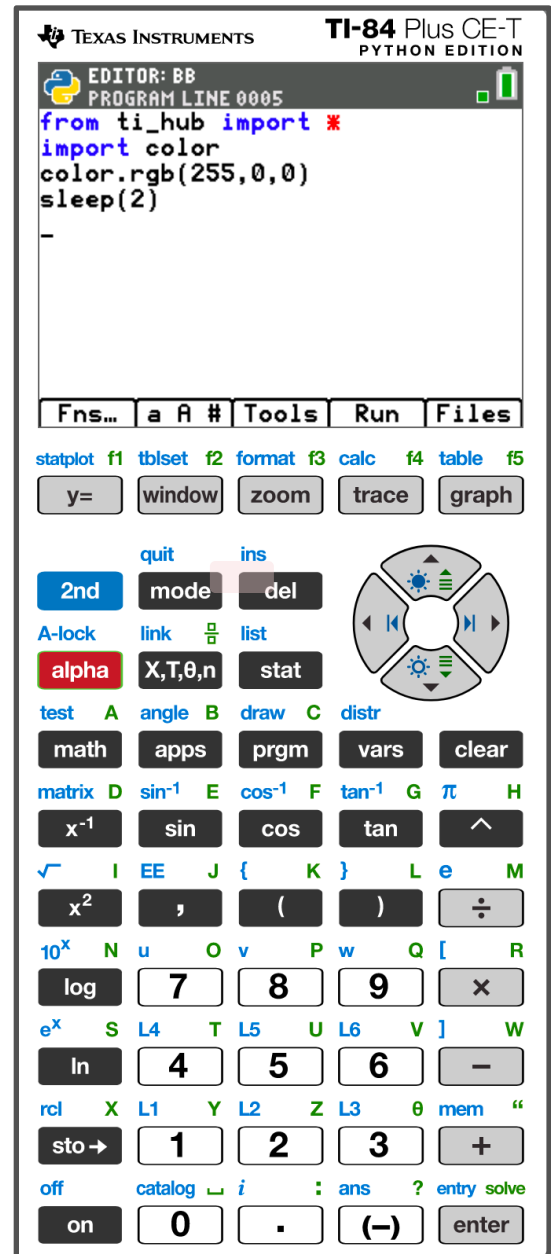
```
EDITOR: HUB2
PROGRAM LINE 0001
_rom ti_hub import *
import color
color.rgb(0,255,255)
sleep(2)
color.rgb(0,255,255)
```

Fns... | a A # | Tools | Run | Files

You are now in the editor ready to make changes or to run the new program.

Entry and Edit Tips

- » Use **number key shortcuts** or **arrow keys** and **[enter]** to select from menus
- » Use **arrow keys** to move the cursor around the screen.
- » Use **[alpha]** **repeatedly** to cycle from numeric, to lower case alpha to upper case alpha entry mode. The cursor indicates the current mode.
- » Use **[2nd] [A-lock]** to lock to alpha entry or to return to numeric entry.
- » Use **[Fns...]** **softkey** to bring up Python function menus, including the **Modul (modules)** menu.
- » Use **[clear]** or **[Esc]** **softkey** to back out of a menu.
- » Use **[del]** as a destructive backspace
- » Use **[2nd] [enter]** from any place on a line to complete the statement and move the cursor to the beginning of a blank line below.
- » Use **[Tools]** **softkey** menu to undo a clear and to copy, cut, paste and more.
- » Use **[Editor]** **softkey** to return to the editor from the Shell.
- » Use **[2nd] [quit]** to leave the Python app and return to the calculator.



COLOR OUTPUTS

Task: Set the color output of the Red, Green, Blue (RGB) LED.

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

Challenge Tasks:

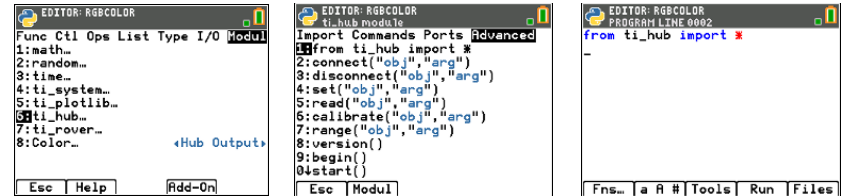
Try to make Yellow

Try to make Cyan

Try to make Magenta

Set the color

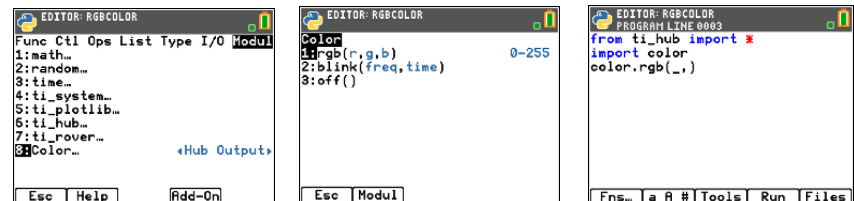
Press **[Fns...]** (Functions), Left Arrow to Modul (Modules) menu, select 6:ti_hub..., then arrow to Advanced to import ti_hub.



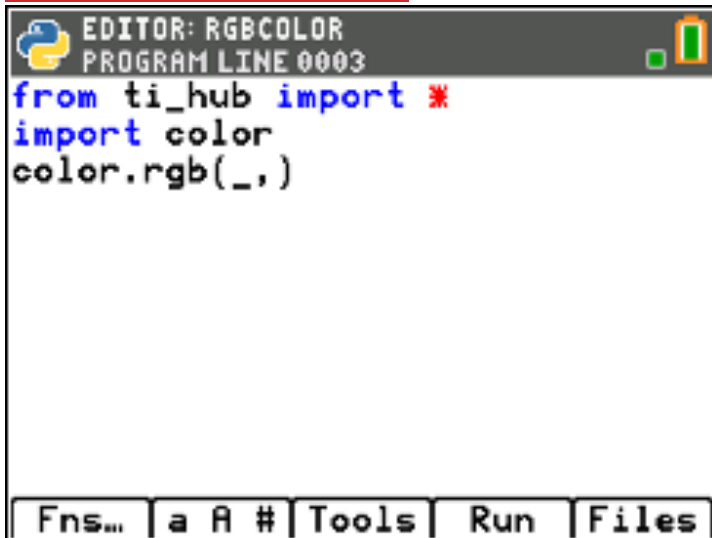
Press **[Fns...]** (Functions), Left Arrow to Modul (Modules) menu, select 6:ti_hub..., Import 1: Hub Built-in Devices, 1:Color to import Color.



Press **[Fns...]** (Functions), Left Arrow to Modul (Modules) menu, select Color..., select 1: rgb(r,g,b). Type in values for red, green and blue between commas. When done with number entry, press **[2nd]** **[enter]** to complete the statement and move the cursor to the next row.



New Program:



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

Challenge Tasks:

Print the name of your color.

Create and Name a Color

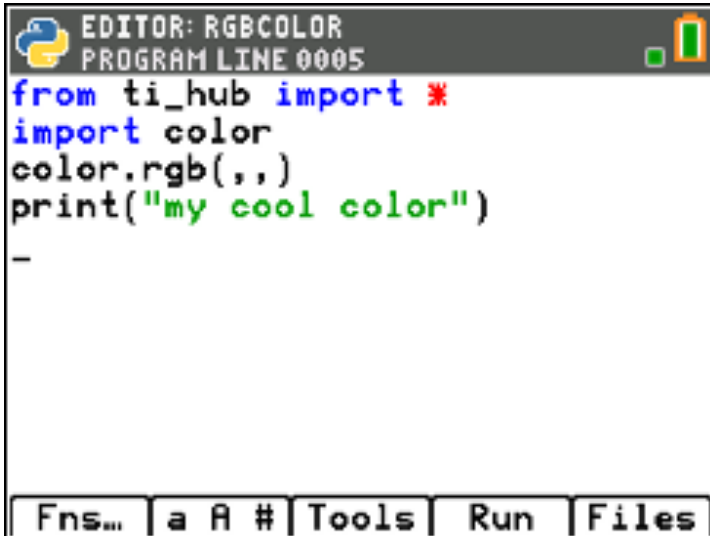
To add the **print()** function to the bottom of your program, move your cursor to the beginning of the empty row at the bottom of your program.

Press [**Fns...**] (Functions), Right Arrow to I/O (Inputs/Outputs) menu, select 1:print().

Press [**2nd**] [**A-lock**] to lock into alpha entry mode. Enclose your color name in quotes [**+**] **key**. Space is the [**0**] **key**.

Press [**2nd**] [**enter**] to complete the statement and move to the next line.

Add to previous Program:



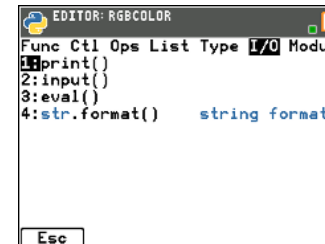
```
EDITOR: RGBCOLOR
PROGRAM LINE 0005
from ti_hub import *
import color
color.rgb(,,)
print("my cool color")
_
```

Fns... | a A # | Tools | Run | Files



```
EDITOR: RGBCOLOR
PROGRAM LINE 0003
from ti_hub import *
import color
color.rgb(,,)
```

Fns... | a A # | Tools | Run | Files



```
EDITOR: RGBCOLOR
PROGRAM LINE 0004
Func Ctl Ops List Type I/O Modul
1:print()
2:input()
3:eval()
4:str.format() string format
```

Esc



```
EDITOR: RGBCOLOR
PROGRAM LINE 0004
from ti_hub import *
import color
color.rgb(,,)
print(_
```

Fns... | a A # | Tools | Run | Files

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.

Display a series of colors

To remove the print statement, move your cursor to the row with print() and press **[clear]**.

The **sleep()** function pauses the program for the number of seconds that you enter as an input.

Press **[Fns...]** (Functions), Left Arrow to Modul (Modules) menu, select 6:ti_hub..., then arrow to Commands and select 2:sleep().

Enter a value for the number of seconds to pause the program. You can use decimal values.

Edit previous Program:



```
EDITOR: RGBCOLOR
PROGRAM LINE 0009
from ti_hub import *
import color
color.rgb(,,)
sleep(2)
color.rgb(,,)
sleep(2)
color.rgb(,,)
sleep(2)
-
```

Fns... | a A # | Tools | Run | Files



```
EDITOR: RGBCOLOR
ti_system module
Import Commands Ports Advanced
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align")
4:disp_clr() clear text screen
5:disp_wait() [clear]
6:disp_cursor() 0=off 1=on
7:while not escape(): [clear]
```

Esc | Modul



```
EDITOR: RGBCOLOR
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Color... <Hub Output>
```

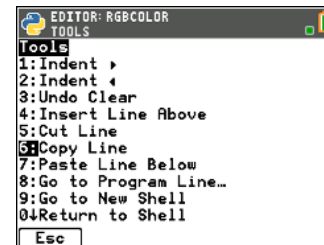
Esc | Help | Add-On



```
EDITOR: RGBCOLOR
PROGRAM LINE 0004
from ti_hub import *
import color
color.rgb(,,)
sleep(_
```

Fns... | a A # | Tools | Run | Files

You can use copy line and paste to line below from the **[Tools]** menu to speed up editing.



```
EDITOR: RGBCOLOR
TOOLS
Tools
1:Indent >
2:Indent <
3:Undo Clear
4:Insert Line Above
5:Cut Line
6:Copy Line
7:Paste Line Below
8:Go to Program Line...
9:Go to New Shell
0:Return to Shell
```

Esc

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

Turn the LED ON and OFF

The `sleep()` function pauses the program for the number of seconds that you enter as an input.

What values for the red, green and blue inputs to the `color.rgb()` function will turn the LED off?

Press **[Fns...]** (Functions), Left Arrow to Modul (Modules) menu, select 6:ti_hub..., then arrow to Commands and select 2:sleep().

Enter a value for the number of seconds to pause the program. You can use decimal values.

New Program:

```

EDITOR: BLINK
PROGRAM LINE 0007
from ti_hub import *
import color
color.rgb(,,)
sleep(2)
color.rgb(,,)
sleep(2)
-
Fns... | a A # | Tools | Run | Files
  
```

```

EDITOR: RGBCOLOR
ti_system module
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Color...
<Hub Output>
Esc | Help | Add-On
  
```

```

EDITOR: RGBCOLOR
ti_system module
Import Commands Ports Advanced
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align")
4:disp_clr() clear text screen
5:disp_wait() [clear]
6:disp_cursor() 0=off 1=on
7:while not escape(): [clear]
Esc | Modul
  
```

```

EDITOR: RGBCOLOR
PROGRAM LINE 0004
from ti_hub import *
import color
color.rgb(,,)
sleep(_
Fns... | a A # | Tools | Run | Files
  
```

You can use copy line and paste to line below from the **[Tools]** menu to speed up editing.

```

EDITOR: RGBCOLOR
TOOLS
Tools
1:Indent >
2:Indent <
3:Undo Clear
4:Insert Line Above
5:Cut Line
6:Copy Line
7:Paste Line Below
8:Go to Program Line...
9:Go to New Shell
0|Return to Shell
Esc
  
```

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

New Program:

```

EDITOR: BLNKLOOP
PROGRAM LINE 0011
from ti_hub import *
import color
#The loop repeats the
#statements in the indented
#block 10 times.
for i in range(10):
    **color.rgb(,,)
    **sleep(2)
    **color.rgb(0,0,0)
    **sleep(2)
-
Fns... | a A # | Tools | Run | Files
  
```

Blink the LED Repeatedly

`for i in range(size):` sets up a loop that will repeat the statements in a block of code for the number of times that you enter for the size variable.

Press **[Fns...]** (Functions), **Right Arrow** to Ctl (Control) menu, select 4:for i in range(size):

Enter a value for the variable `size`. Size sets the number of times that the loop will repeat.

After entering the value for size press [2nd] [enter] to complete the the statement and move to the beginning of the next row.

The next row is indented. The indented rows are part of the loop block of statements that will be repeated.

```

EDITOR: BLNKLOOP
Func Ctl Ops List Type I/O Modul
1:if ..
2:if .. else ..
3:if .. elif .. else
4:for i in range(size):
5:for i in range(start,stop):
6:for i in range(strt,stp,step):
7:for i in list:
8:while condition:
9:elif :
0:else:
Esc
  
```

```

EDITOR: BLNKLOOP
PROGRAM LINE 0004
from ti_hub import *
import color
for i in range(_:
**
Fns... | a A # | Tools | Run | Files
  
```

```

EDITOR: BLNKLOOP
PROGRAM LINE 0005
from ti_hub import *
import color
for i in range(10):
**
Fns... | a A # | Tools | Run | Files
  
```

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

SOUND OUTPUTS

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?

New Program:

```

EDITOR: TONE
PROGRAM LINE 0004
from ti_hub import *
import sound
sound.tone(,)
-
Fns... | a A # | Tools | Run | Files
    
```

Play a Sound Tone

Press **[Fns...]** (Functions), Left Arrow to Modul (Modules) menu, select 6:ti_hub..., then arrow to Advanced to import ti_hub.

```

EDITOR: RGBCOLOR
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Color...
<Hub Output>
Esc | Help | Add-On

EDITOR: RGBCOLOR
ti_hub module
Import Commands Ports Advanced
1:from ti_hub import *
2:connect("obj","arg")
3:disconnect("obj","arg")
4:set("obj","arg")
5:read("obj","arg")
6:calibrate("obj","arg")
7:range("obj","arg")
8:version()
9:begin()
0:start()
Esc | Modul

EDITOR: RGBCOLOR
PROGRAM LINE 0002
from ti_hub import *
-
Fns... | a A # | Tools | Run | Files
    
```

Press **[Fns...]** (Functions), Left Arrow to Modul (Modules) menu, select 6:ti_hub..., Import 1: Hub Built-in Devices, 3:Sound to import Sound.

```

EDITOR: RGBCOLOR
ti_hub module
Import Commands Ports Advanced
1:Hub Built-in devices...
2:Input devices...
3:Output devices...
4:Collect data...
Esc | Modul

EDITOR: TONE
ti_hub module
Import Commands Ports Advanced
1:Color RGB LED Output
2:Light Red LED Output
3:Sound Sound Output
4:Brightness Light Sensor Input
Esc | Import

EDITOR: TONE
PROGRAM LINE 0003
from ti_hub import *
import sound
-
Fns... | a A # | Tools | Run | Files
    
```

Press **[Fns...]** (Functions), Left Arrow to Modul (Modules) menu, select Sound..., select 1: tone(freq,time). Type in values for sound frequency and time duration separated by commas. When done with number entry, press **[2nd] [enter]** to complete the statement and move the cursor to the next row.

```

EDITOR: TONE
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Sound...
<Hub Output>
Esc | Help | Add-On

EDITOR: TONE
ti_hub module
Import Commands Ports Advanced
1:tone(freq,time)
2:note("string",time)
3:tone(freq,time,tempo)
4:note("string",time,tempo)
Esc | Modul

EDITOR: TONE
PROGRAM LINE 0003
from ti_hub import *
import sound
sound.tone(,)
-
Fns... | a A # | Tools | Run | Files
    
```

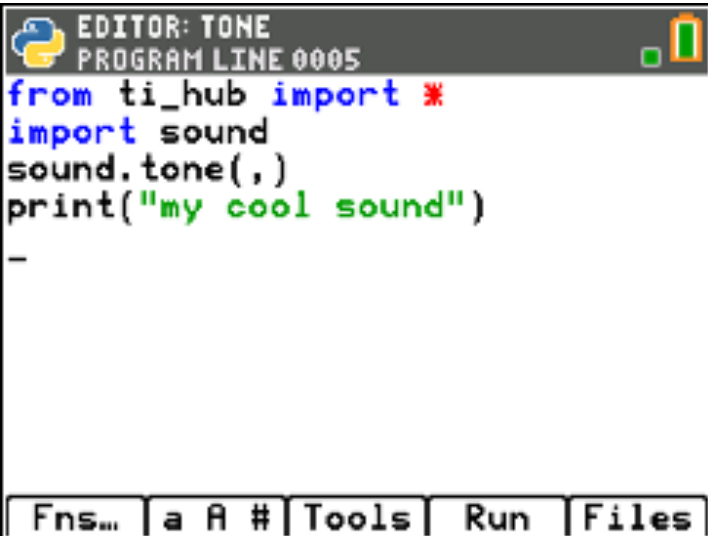
Task: Play a sound tone by entering a value for frequency (sound vibrations per second) and a value for time in seconds to play the tone.

Find your favorite frequency.

Challenge Tasks:

Give your frequency a name and print the name.

Add to previous Program:



```
EDITOR: TONE
PROGRAM LINE 0005
from ti_hub import *
import sound
sound.tone(,)
print("my cool sound")
_
```

Fns... | a A # | Tools | Run | Files

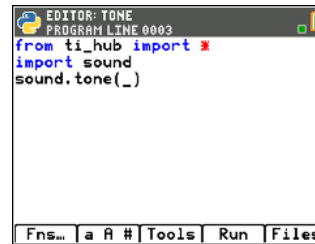
Find your Favorite Sound Tone

To add the **print()** function to the bottom of your program, move your cursor to the beginning of the empty row at the bottom of your program.

Press [**Fns...**] (Functions), Right Arrow to I/O (Inputs/Outputs) menu, select 1:print().

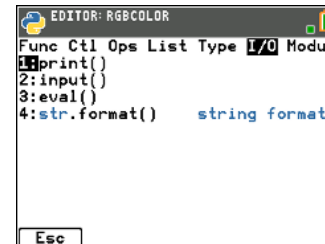
Press [**2nd**] [**A-lock**] to lock into alpha entry mode. Enclose your color name in quotes [**+**] **key**. Space is the [**0**] **key**.

Press [**2nd**] [**enter**] to complete the statement and move to the next line.



```
EDITOR: TONE
PROGRAM LINE 0003
from ti_hub import *
import sound
sound.tone(_)
```

Fns... | a A # | Tools | Run | Files



```
EDITOR: RGBCOLOR
Func Ctl Ops List Type I/O Modul
1:print()
2:input()
3:eval()
4:str.format() string format
```

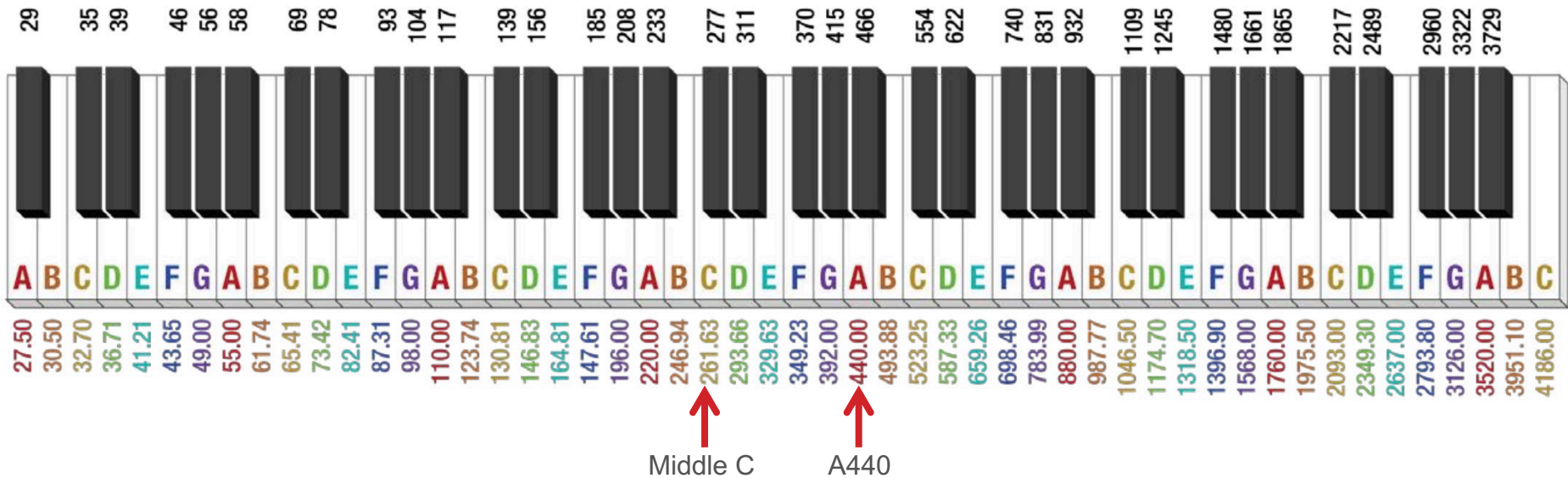
Esc



```
EDITOR: TONE
PROGRAM LINE 0004
from ti_hub import *
import sound
sound.tone(,)
print(_)
```

Fns... | a A # | Tools | Run | Files

Sound Frequencies and Musical Notes



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 Making Music with Code project for an in-depth treatment of digital music with the Hub. https://resources.tistemprojects.com/tistemprojects-home/?resource_id=2403

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

Create your own doorbell tone.

Play a Musical Note

`sound.note()` is on the Sound menu.

Press [alpha] then the key corresponding with the desired letter to enter notes. The number represents the octave.

The first screenshot shows the TI-84 Plus CE calculator editor with the 'Sound' menu highlighted. The menu options are: 1:tone(freq,time), 2:note("string",time), 3:tone(freq,time,tempo), and 4:note("string",time,tempo). The second screenshot shows the same editor with the 'note' command entered: `note("string",time)`.

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.

`sleep()` is on the Hub Commands menu.

The first screenshot shows the TI-84 Plus CE calculator editor with the 'Hub Commands' menu highlighted. The menu options are: 1:from ti_system import *, 2:sleep(seconds), 3:disp_at(row,"text","align"), 4:disp_clr() clear text screen, 5:disp_wait() [clear], 6:disp_cursor() 0=off 1=on, and 7:while not escape(): [clear]. The second screenshot shows the same editor with the `sleep(seconds)` command entered: `sleep(seconds)`.

Note: 6:Copy Line and 7:Paste to Line Below from the Tools menu may help you enter your program faster.

New Program:

The screenshot shows the TI-84 Plus CE calculator editor with the following code entered:

```

from ti_hub import *
import sound

sound.note("e4",.5)
sleep(.5)
sound.note("c4",1)
sleep(1)
  
```

The editor also shows the 'Tools' menu at the bottom with options: Fns..., a A #, Tools, Run, and Files.

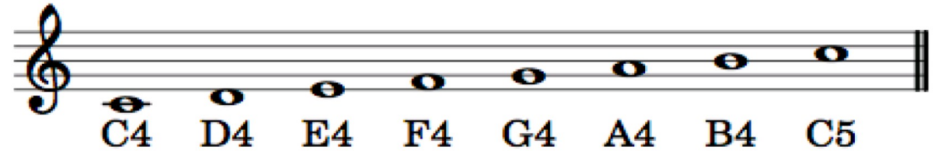
Task: Write a program to play each note of Do-Re-Mi-Fa-Sol-La-Si-Do as whole notes.

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.

Play the Notes of an Octave



`sound.note()` is on the Sound menu.

Press [alpha] then the key corresponding with the desired letter to enter notes. The number represents the octave.

New Program:

```

EDITOR: OCTAVE
PROGRAM LINE 0006
from ti_hub import *
import sound

sound.note("c4",2.4)
sleep(2.4+.1)
-
  
```

Fns... a A # Tools Run Files

```

EDITOR: BELL
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Sound...
  
```

Esc Help Add-On

```

EDITOR: BELL
Sound
1:tone(freq,time)
2:note("string",time)
3:tone(freq,time,tempo)
4:note("string",time,tempo)
  
```

Esc Modul

`sleep()` is on the Hub Commands menu.

```

EDITOR: BELL
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Sound...
  
```

Esc Help Add-On

```

EDITOR: BELL
ti_system module
Import Commands Ports Advanced
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align")
4:disp_clr() clear text screen
5:disp_wait() [clear]
6:disp_cursor() 0=off 1=on
7:while not escape(): [clear]
  
```

Esc Modul

Note: 6:Copy Line and 7:Paste to Line Below from the Tools menu may help you enter your program faster.

BRIGHTNESS (LIGHT LEVEL) INPUTS

Measure Brightness

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.

Brightness sensor



To **import** `brightns`, the Hub built-in light level sensor, press **[Fns...]** (Functions), **Left Arrow** to Modul (Modules) menu, select 6:ti_hub..., Import 1: Hub Built-in Devices, 4: Brightness.

```
EDITOR: BRIGHT
ti_hub module
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
Esc Help Add-On
```

```
EDITOR: BRIGHT
ti_system module
Import Commands Ports Advanced
1:Hub Built-in devices...
2:Input devices...
3:Output devices...
4:Collect data...
Esc Modul
```

```
EDITOR: BRIGHT
Postke = Brightness -> Modul menu.
Hub Built-in devices
1:Color RGB LED Output
2:Light Red LED Output
3:Sound Sound Output
4: Brightness Light Sensor Input
Esc Import
```

```
EDITOR: BRIGHT
PROGRAM LINE 0003
from ti_hub import *
import brightns
Fns... a A # Tools Run Files
```

Find **while not escape()**: on the Hub Commands menu. This sets up a loop that will run until the **[clear]** key is pressed. To use **while not escape()**: you will need to import the ti-system module that is available on the Commands menu. You may use the Insert Line Above command from the Tools menu. **sleep()** is also available from the Hub Commands menu.

New Program:

```
EDITOR: BRIGHT
PROGRAM LINE 0009
from ti_hub import *
import brightns
from ti_system import *

while not escape():
  **b=brightns.measurement()
  **print("brightness= ",b)
  **sleep(.25)
-
```

```
EDITOR: BRIGHT
ti_system module
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
8: Brightness...
Esc Help Add-On
```

```
EDITOR: BRIGHT
ti_system module
Import Commands Ports Advanced
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align")
4:disp_clr() clear text screen
5:disp_wait() [clear]
6:disp_cursor() 0=off 1=on
7:while not escape(): [clear]
Esc Modul
```

```
EDITOR: BRIGHT
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
8: Brightness...
Esc Help Add-On
```

```
EDITOR: BRIGHT
ti_system module
Import Commands Ports Advanced
1:from ti_system import *
2:sleep(seconds)
3:disp_at(row,"text","align")
4:disp_clr() clear text screen
5:disp_wait() [clear]
6:disp_cursor() 0=off 1=on
7:while not escape(): [clear]
Esc Modul
```

The measurement value is stored to a variable named **b**. **=** is available from the test menu at **[2nd][math]** or by pressing **[sto ->]**. **brightns.measurement()** is available from the brightness menu. **print()** is available from the I/O menu. Quotes are available by pressing **[alpha][+]**.

```
EDITOR: BRIGHT
Ops
1:≙ [sto ->]
2:≙= equal
3:≙≠ not equal
4:≙>
5:≙<
6:≙<=
7:≙<≠
8:and
9:or
0:not
Esc
```

```
EDITOR: BRIGHT
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
8: Brightness...
Esc Help Add-On
```

```
EDITOR: BRIGHT
Brightness
1:measurement()
2:range(min,max)
Esc Modul
```

```
EDITOR: BRIGHT
Func Ctl Ops List Type I/O Modul
1:print()
2:input()
3:eval()
4:str.format() string format
Esc
```

Control an RGB LED with Brightness Measurements

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 `color.rgb(255-b,0,0)`. How does the behavior change?

Add to previous Program:

```

EDITOR: BRIGHT
PROGRAM LINE 0001
from ti_hub import *
import brightns
from ti_system import *
import color
brightns.range(0,255)

while not escape():
    b=brightns.measurement()
    print("brightness= ",b)
    color.rgb(b,0,0)
    sleep(.25)
  
```

Brightness sensor



Set the range of brightness measurements to match the output range values of the `color.rgb` LED, 0-255.

Find the `brightness.range()` function on the Hub Built-in Devices, Brightness Input menu.

Note: The default brightness range is 0 to 100.

```

EDITOR: BRIGHT
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:ti_brightness...
9:Color...
  
```

```

EDITOR: BRIGHT
Brightness
1:measurement()
2:range(min,max)
  
```

You will also need to import `color` from the Hub, Built-in devices... menu.

Then you will insert a `color.rgb()` statement into the program.

You may use the Insert Line Above command from the Tools menu.

```

EDITOR: BRIGHT
Paste + color -> Modul menu
Hub Built-in devices
1:Color RGB LED Output
2:Light Red LED Output
3:Sound Sound Output
4:ti_brightness Light Sensor Input
  
```

```

EDITOR: BRIGHT
PROGRAM LINE 0010
from ti_hub import *
import brightns
from ti_system import *
import color
brightns.range(0,255)

while not escape():
    b=brightns.measurement()
    print("brightness= ",b)
    sleep(.25)
  
```

```

EDITOR: BRIGHT
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:ti_brightness...
9:Color...
  
```

Thank You



www.TIstemProjects.com

Contact stem-team@ti.com with questions