Design a Digital Mood Ring – Part 2

TI-NSPIRE™ CXII PYTHON

TI-INNOVATOR™ STEM PROJECT

TEACHER NOTES

Overview:

In this project, students build and program an introductory example of a feedback and control loop. This engineering principle is central to many industrial systems and consumer products. There are feedback and control principles in many Texas Instruments TI-Innovator™ Hub projects. During the project, students will engage with fundamental concepts in programming, such as variables, loops, conditional statements and Boolean operators.

The project includes a series of challenges that build the conceptual knowledge and skills needed for the final open-ended challenge.

Goals:

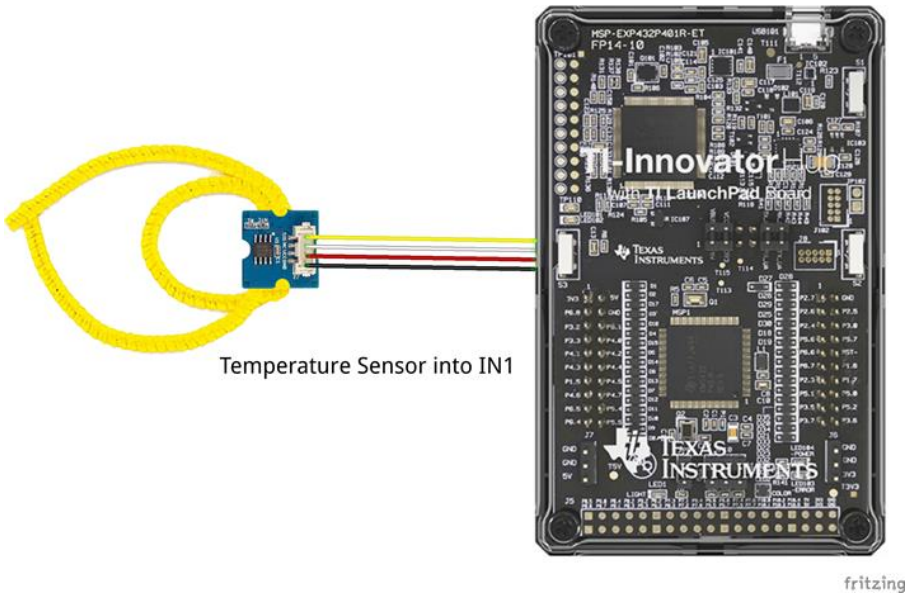
Students will:

- Create and edit a Python program on the calculator.
- Connect the Hub with a sensor to the calculator.
- Write programs that include many commonly used Hub and calculator commands.
- Build a simple feedback and control system.

Note: For programming commands for this project, refer to the "Python Syntax Quick Reference" document

Setup Project:

Students may work in groups of two or three.



Supplies:

- Calculator
- Unit to Unit Cable
- Hub
- Temperature sensor
- Grove Cable
- Chenille ("Fuzzy") Wire

Note: Students will need to push the wire through the holes on the temperature sensor board to create a ring. The sensing element is on the flat side of the sensor board. Expect (and encourage!) trial and error as students attempt to determine the most effective ring design.

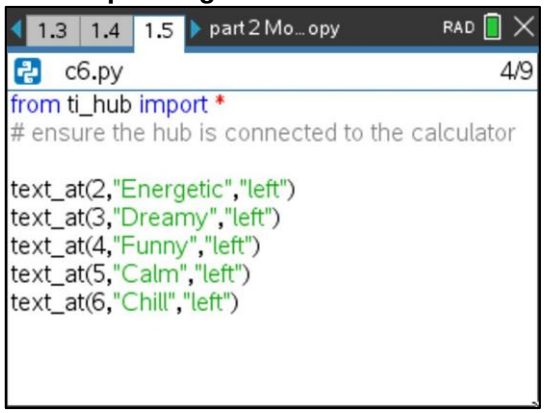
Student Activity:

Coding Challenge 6: Choose five mood messages from the list and write a program to display them, each on a different line, of the TI-Nspire CXII display.

Teacher Notes:

- Mood messages to choose from

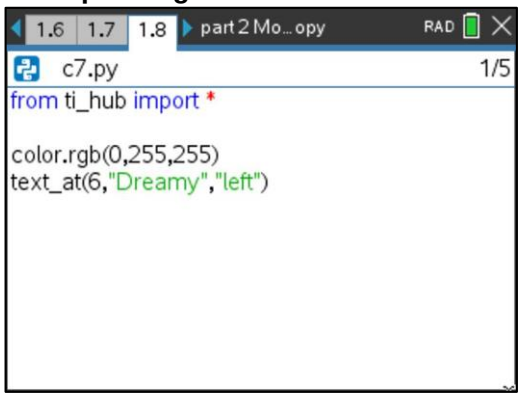
- **Example Program:**



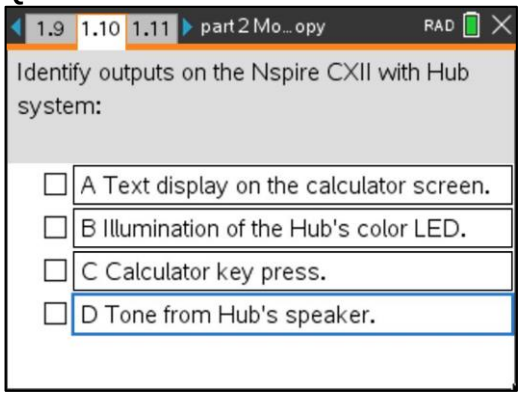
Coding Challenge 7: Combine your favorite mood color with your favorite mood message.

Write a program that displays both.

- **Example Program:**



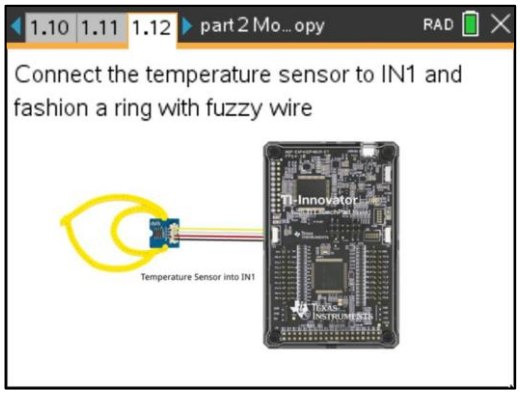
- **Question:**



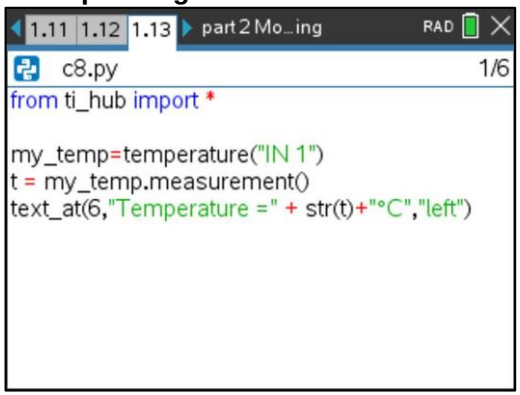
Coding Challenge 8: Connect the temperature sensor to IN1 and write a program that creates a temperature object named my_temp.

Measure the my_temp object and store value in the variable named temp. Display an appropriate prompt with the measurement value and units.

- Connect temperature sensor to port IN1



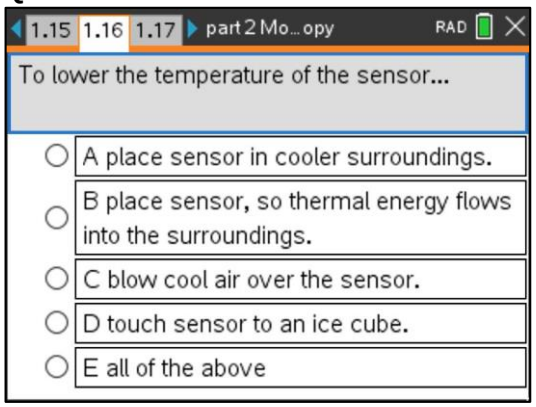
- **Example Program:**



Science Activity 10: Use the previous program to explore the temperatures around you.
-What is the temperature of the room?
-What is the temperature of your skin?
-How low of a reading can you measure?
-What is the temperature of an ice cube?

* do not submerge sensor in liquid

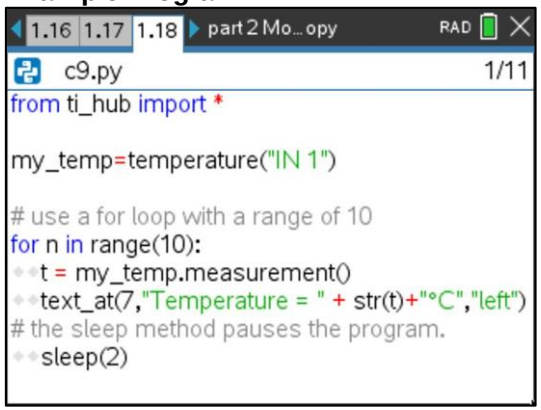
- **Question:**



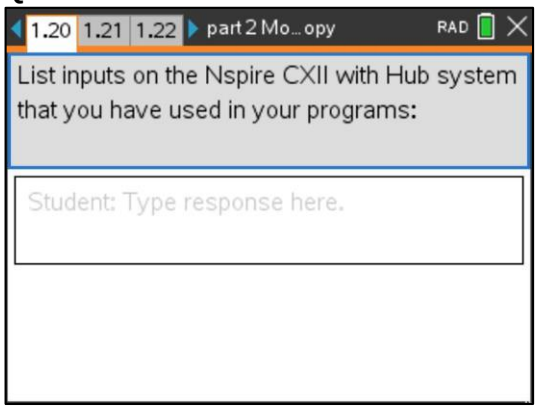
Coding Challenge 9: Write a program using a for loop that will read and display ten temperature measurements.

Use sleep(2) to pause for two seconds in each cycle.

- **Example Program:**



- **Question:**



Coding Challenge 10: Use a while loop to continuously monitor temperature.

Include an if conditional statement using appropriate temperatures to display the following text and LED color:
- "cool" - blue
- "just right" - green
- "hot" - red

- **Example Program:**

```
c10.py saved successfully
from ti_hub import *

my_temp=temperature("IN 1")

# use the while loop with escape key to exit
while get_key() != "esc":
    t = my_temp.measurement()
    text_at(5,"Temperature = " + str(t)+"°C","left")
# use the if statment to make a decision
    if t<20:
        text_at(8,"It is cool","left")
        color.rgb(0,0,255)
    if t>=20 and t <25:
        text_at(8,"It is just right","left")
        color.rgb(0,255,0)
    if t>=25:
        text_at(8,"It is hot","left")
        color.rgb(255,0,0)
    sleep(1)
```

Final Coding Challenge 11: Use the skills from all of the previous challenges to design and code your mood ring. Your program should display mood messages and colors over a range of finger temperatures.

Helpful tips:
-As a starting point, modify a copy of the previous program. Switch to that program editor page, and select [ctrl]+B, then [menu]->Actions->Create Copy.

- Include at least five if case intervals that change the mood color and message based on finger temperatures.

- Temperature intervals of about two degrees will help your ring respond to typical temperature measurements.

- **Example Program:**

c11.py saved successfully

```
from ti_hub import *

my_temp=temperature("IN 1")

while get_key() != "esc":
    t = my_temp.measurement()
    text_at(5,"Temperature = " + str(t)+"°C","left")
    if t<20:
        text_at(8,"I feel happy","left")
        color.rgb(255,0,0)
    if t>=20 and t<22:
        text_at(8,"I feel hopeful","left")
        color.rgb(0,255,0)
    if t>=22 and t<24:
        text_at(8,"I feel meh","left")
        color.rgb(0,0,255)
    if t>=24 and t<26:
        text_at(8,"I feel calm","left")
        color.rgb(255,255,0)
    if t>=26:
        text_at(8,"I feel chill","left")
        color.rgb(255,0,255)
    sleep(1)
```