Challenge: Design a Digital Mood Ring

In this project, you will investigate temperature measurement using a temperature sensor and then build a digital mood ring that changes the color of the TI-Innovator Hub's Red, Green, and Blue (RGB) light emitting diode (LED) as the temperature of your finger changes. You can even display what mood you are in on your calculator!

First, you will work through a series of "challenges" that will build your skills and knowledge to be able to achieve the final goal of creating a digital mood ring.

Background:
The mood ring was invented by Joshua Reynolds. Mood rings enjoyed fad popularity in the 1970s and are still around today. The stone of the ring changes color, supposedly according to the mood of the wearer. The 'stone' of a mood ring is really a hollow glass shell containing *thermotropic* liquid crystals. Modern mood rings are made from a flat plastic strip containing liquid crystals that is inserted into the hollow glass and mounted within the bezel of the mood ring. The crystals respond to changes in temperature by twisting in a regular way. The twisting changes their molecular geometry, which alters the wavelengths of light that are reflected from the crystals. Wavelengths of light are another way of saying color, so when the temperature of the liquid crystals changes, so does the color reflected from the stone. Thus, the ring changes color with the hand temperature of the wearer.

Activity Materials:

- Temperature Sensor
- TI-Innovator™ Hub (sketch vs. 1.3)
- TI-Nspire CX (OS version. 4.5 or later)
- Colored pipe cleaner (Chenille stem)
- A mood to be measured!

Challenges:

1. Use SET COLOR to explore using the color LED. Try to find the RGB values of all the colors in the mood chart. e.g. Send "SET COLOR 255 128 0" will make yellow.

2. Use DispAt command to display your name at several locations on the screen.

3. Use a For..EndFor loop to display the numbers 1 through 10.

4. Connect a temperature sensor to the TI-Innovator Hub and display the temperature on the calculator.

5. Use a loop to read and display temperature.

6. (Optional) Use a loop to read, display and log into list arrays for time and temperature. Graph the result.

7. Use a While..EndWhile loop along with getKey() command to monitor temperature and make a decision to display a message, "Hot" or "Cold". Then modify your program to include a 3rd level between hot and cold, "Nice".

8. **Final Challenge**: Build a mood ring to repeatedly read the temperature sensor, determine the mood of the person, display the temperature value and display the mood.

Example snippets of TI-BASIC Code for TI-Nspire CX:

Code snippet to set the RGB LED to blue on the Hub:

```
Send "SET COLOR 0 0 255"
```

Code snippet to set the RGB LED to yellow on the Hub:

```
Send "SET COLOR 255 128 0"
```

Code snippet to display a message on the calculator:

```
DispAt 5,"I AM HAPPY"
```

Code snippet to connect and read the temperature sensor:

```
Send "CONNECT TEMPERATURE 1 TO IN 1"
Send "READ TEMPERATURE 1"
Get t
DispAt 4, "TEMPERATURE IS ",t
```

Code snippet to read and display the temperature sensor every second until the Esc key is pressed:

```
key:=" "
While key≠"esc"
key:=getKey()
Send "READ TEMPERATURE 1"
Get t
DispAt 4, "TEMPERATURE IS",t,"◦C"
Wait 1
EndWhile
```

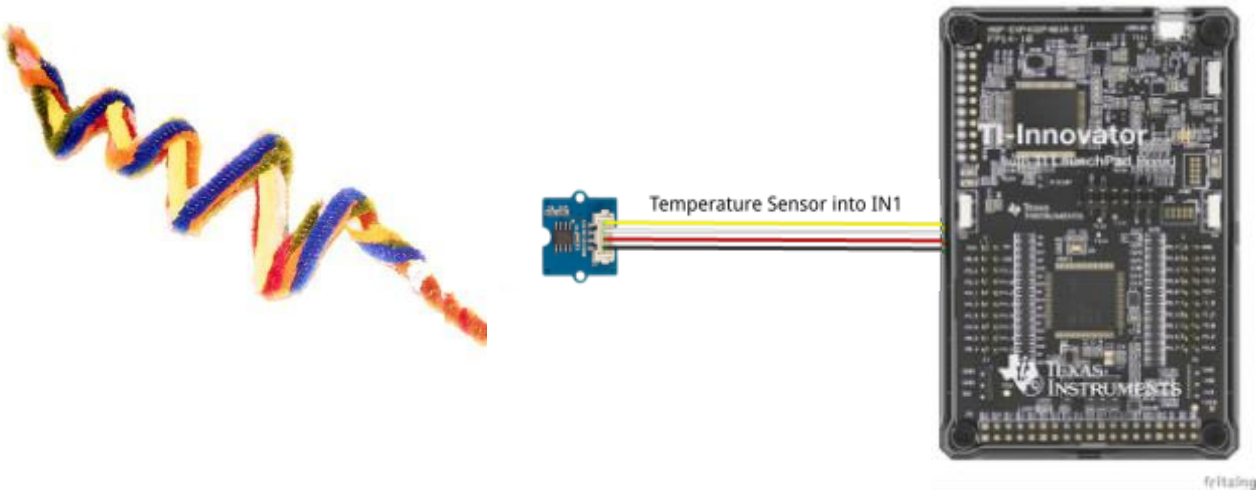
Code snippet to read temperature every .5 second for 1 minute and storing time into list named time and temperature into a list named *temp*. These lists may be graphed on a Data and Statistics page.

```
time:={}
temp:={}
For n,1,120
time[n]:=n*.5
Send "READ TEMPERATURE 1"
Get t
temp[n]:=t
DispAt 4,"TEMPERATURE IS", t
Wait .5
EndFor
```

Code snippet to make decisions using the If-ElseIf- Then-EndIf programming structure:

```
Send "READ TEMPERATURE 1"
Get t
If t<24 Then
 Send "SET COLOR 255 0 0"
EndIf
If t≥24 and t<27 Then
 Send "SET COLOR 0 255 0"
EndIf
If t≥27 Then
 Send "SET COLOR 0 0 255"
EndIf
```

Sensor and actuator Hub connections:



Calculator Notes:

- On the Home screen press 4:Current to return to your document file.
- On the Home screen Press 1:New to create a new document file.
- You create and edit programs in a Program Editor app. You run programs from within a Calculator app.
- Use the [menu] key to see the options for your current app.
- ctrl-b is the shortcut from the Check Syntax and Store menu to store changes to your program.
- ctrl-r is the shortcut from the Check Syntax and Store menu to store changes to your program and paste the name to a Calculator app
- Press [enter] to run a program named on a Calculator app entry line.
- The Calculator app “remembers” the last command entered. Press Enter after a program has run to run the program again.
- Find your program names in a Calculator app by pressing the [var] (variables) key.
- Move from page to page by using ctrl-left arrow and ctrl-right arrow or by using the touchpad pointer to click on the desired page tab.
- ctrl-doc (+page) will add a blank page to your document.
- ctrl-z will undo your last action.
- To stop (“break”) a program press and hold the ON key until you receive a dialogue box.
- ctrl-s is the shortcut for saving your entire document file. Do this periodically to save your work.