August 8, 2017

# Innovate Your Classroom with the TI Innovator ™ Hub

Todd Graba
tgraba@d155.org
Stacy Thibodeaux
svthibodeaux@lpssonline.com

| Practices in Mathematics, Science, and English Language Arts* | | |
|---|---|---|
| **M1.** Make sense of problems and persevere in solving them. | **S1.** Asking questions (for science) and defining problems (for engineering). | **E1.** They demonstrate independence. |
| **M2.** Reason abstractly and quantitatively. | **S2.** Developing and using models. | **E2.** They build strong content knowledge. |
| **M3.** Construct viable arguments and critique the reasoning of others. | **S3.** Planning and carrying out investigations. | **E3.** They respond to the varying demands of audience, task, purpose, and discipline. |
| **M4.** Model with mathematics. | **S4.** Analyzing and interpreting data. | **E4.** They comprehend as well as critique. |
| **M5.** Use appropriate tools strategically. | **S5.** Using mathematics, information and computer technology, and computational thinking. | **E5.** They value evidence. |
| **M6.** Attend to precision. | **S6.** Constructing explanations (for science) and designing solutions (for engineering). | **E6.** They use technology and digital media strategically and capably. |
| **M7.** Look for and make use of structure. | **S7.** Engaging in argument from evidence. | **E7.** They come to understanding other perspectives and cultures. |
| **M8.** Look for and express regularity in repeated reasoning. | **S8.** Obtaining, evaluating, and communicating information. | |

*The common Core English Language Arts uses the term "student capacities" rather than the term "practices" used in Common Core Mathematics and the Next Generation Science Standards.

# CCSS + NGSS = TI ❤

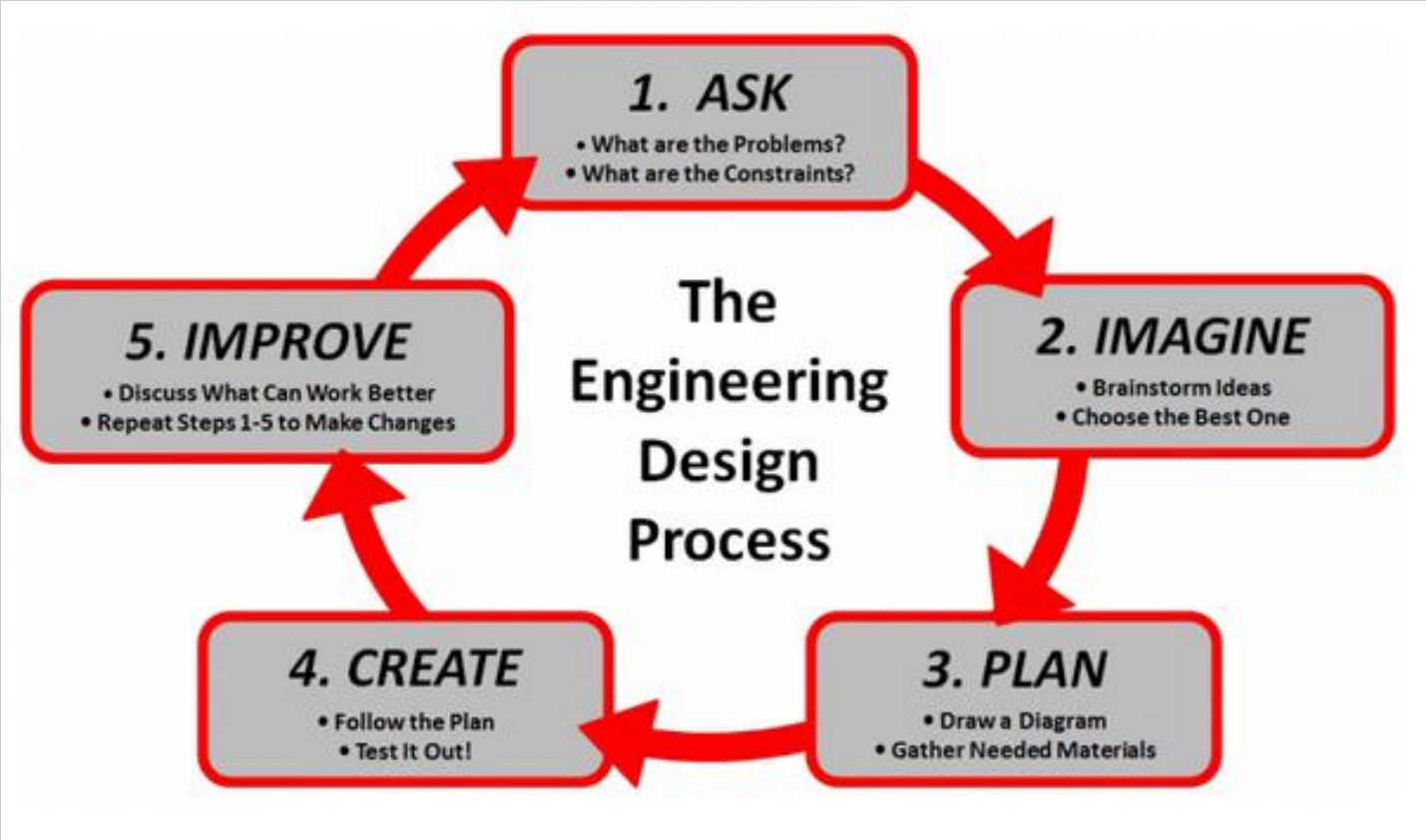
## CCSS Says

- Make sense of problems, persevere in solving them
- Reason abstractly and quantitatively
- Construct viable arguments and critique the reasoning of others
- Model with mathematics
- Use appropriate tools strategically
- Attend to precision

## NGSS Says

- Asking questions and defining problems
- Using mathematical and computational thinking
- Engaging in argument from evidence
- Developing and using models
- Planning and carrying out investigations
- Analyzing and interpreting data

# Engineering Design Process



# Innovator

Innovator contains:

- 3 input and 3 output ports
- Breadboard connector with 20 labeled pins (we will use late)
- $I^2C$ port
- Mini-USB port (connect to TI 84 CE, TI Nspire CX, computer running TI Nspire software)
- Micro-USB power port
- Red-LED
- Red-Green-Blue (RGB) LED
- Light Brightness Sensor
- Speaker

# Up to date?

The operating systems you should have on your devices are:

TI-84 CE - 5.2.2

Hub app needed for 84 CE (same link for CE update above)

TI NSpire CX - 4.4.0.532

TI Innovator Hub - 1.0.0.16

# Programs:

Let’s program the Innovator’s Red light on the TI 84 CE (Connect Software) (send signal to Hub)

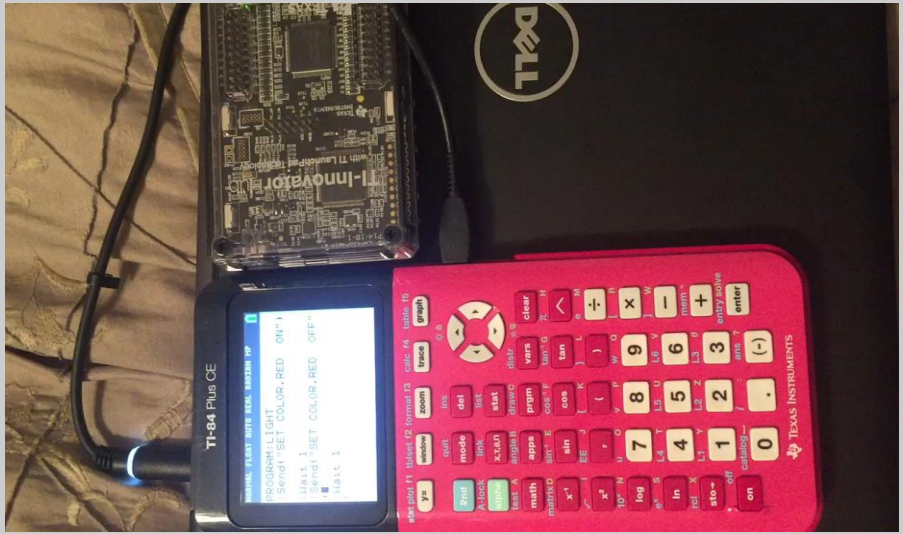
And then

Let’s program the Innovator’s Brightness Sensor on the TI Nspire CX Teacher Software (Read data from Hub)

# Red Light Program using the 84-CE:

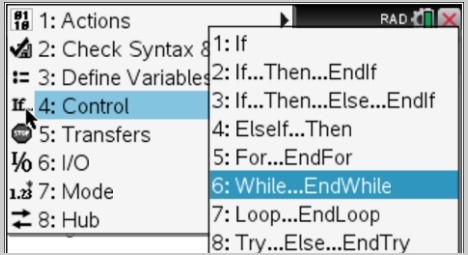
```
NORMAL FLOAT AUTO REAL RADIAN MP

PROGRAM:REDLT
:Send("SET COLOR.RED ON")
:Wait 1
:Send("SET COLOR.RED OFF")
```

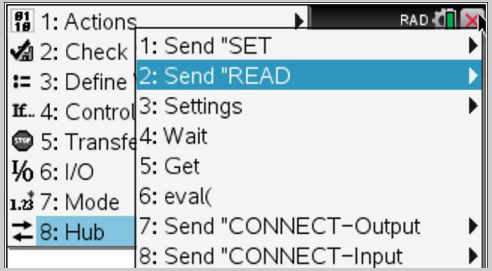


# Brightness Sensor Program:

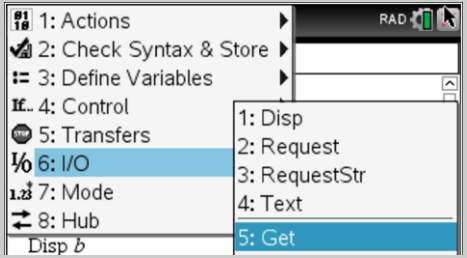
Using a While loop (setting the initial condition)




Reading data from Hub (Send “READ...”




Getting the data



# 10 Minutes of Code with Hub Project 5

Coding Activities on education.ti.com, with Science through Engineering Design and Path to STEM skill builders and projects here.

Developed and refined by TI's team of STEM educators.

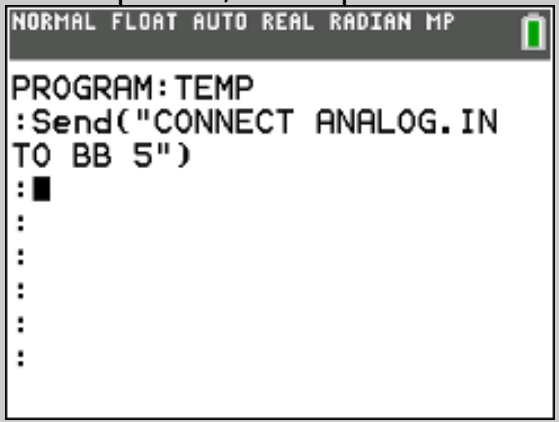
Ready to use

With extensions

Edit as you see fit, these are just a launching point, you are really only limited by your creativity!

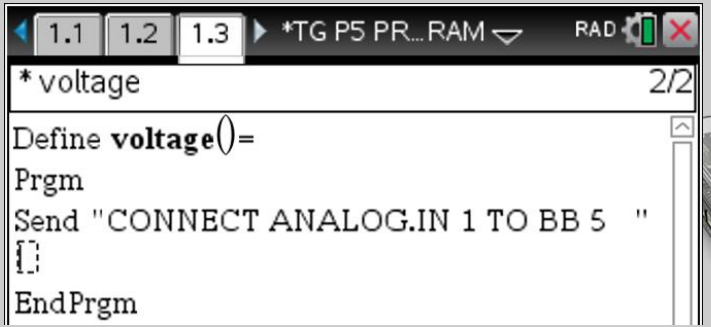
# Program (84-CE on left, NSpire CX on right):

Connect Hub to Breadboard (BB) from analog port
Prgm, new, temp
Prgm, hub, 8 (send connect input),
 C (analog.in)
Prgm, hub, settings, to
Prgrm, hub, ports, BB 5
Close quotes, close parenthesis




Connect Hub to Breadboard (BB) using analog port

Menu->8:Hub->8:Send “CONNECT-Input->D:ANALOG.IN
Type number 1 to name which ANALOG.IN being used
Menu->8:Hub->3:Settings->3:TO
Menu->8:Hub->9:Ports->C:BB 5



# Program (84-CE on left, NSpire CX on right):

Create a For statement -
For range of N - 1-50:
Prgm, CTL, For(
Type in N,1,50)

```
NORMAL FLOAT AUTO REAL RADIAN MP

PROGRAM:TEMP
:Send("CONNECT ANALOG.IN
TO BB 5")
:For(N,1,50)
:
:
:
:
:
```

Create a For statement

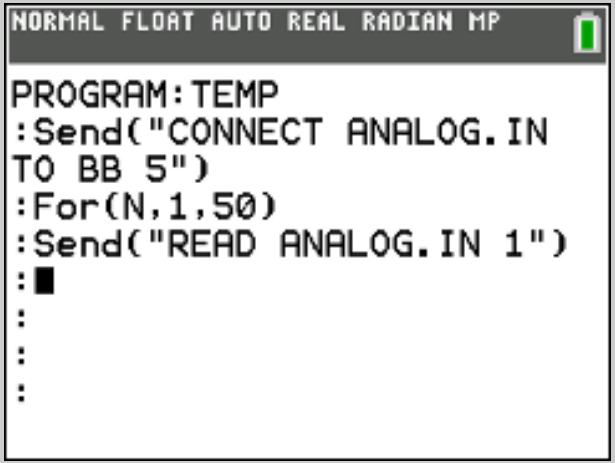
Menu->4:Control->5:For...EndFor
Between commas add n,1,50 for 50 loops

```
1.1 1.2 1.3 *TG P5 PR...RAM RAD
* voltage 3/4
Define voltage()=
Prgm
Send "CONNECT ANALOG.IN 1 TO BB 5 "
For n,1,50

EndFor
EndPrgm
```

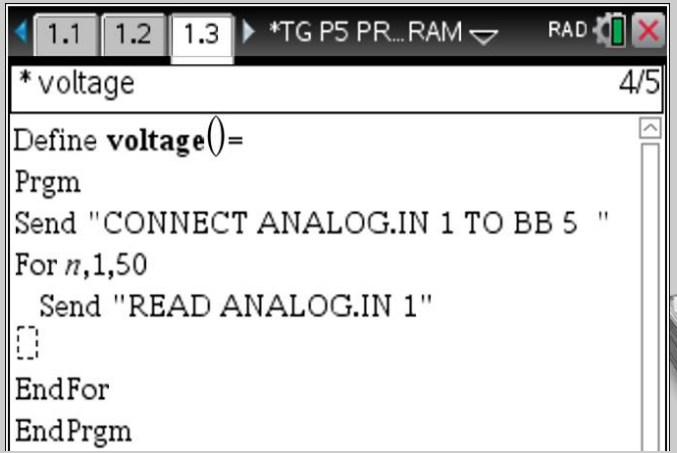
# Program:

Send message to read analog in pin 1:
Program
HUB, 2 Send (“READ...
C, ANALOG.IN
Type 1, close quotes, close parenthesis
Enter




Send message to read analog in pin 1:
Menu->8:Hub->2: Send “READ->D:ANALOG.IN
Type 1 to keep with the name of the ANALOG
we are using

Note: indentation just to make reading program easier, not necessary



# Program:

Get analog in:
Program
I/O - A - Get(
Type V for analog, close parenthesis
Enter

```
NORMAL FLOAT AUTO REAL RADIAN MP

PROGRAM:TEMP
:Send("CONNECT ANALOG.IN
TO BB 5")
:For(N,1,50)
:Send("READ ANALOG.IN 1")
:Get(V)
:
:
:
```

Get analog in values
Menu->6:I/O->5:Get
Type V for value name of analog reading

```
1.1  1.2  1.3  *TG P5 PR...RAM   RAD
* voltage                        4/5
Define voltage()=
Prgm
Send "CONNECT ANALOG.IN 1 TO BB 5  "
For n,1,50
  Send "READ ANALOG.IN 1"
  Get V
EndFor
EndPrgm
```

# Program:

Set display and value for analog in=A:
Program, I/O
3: Disp
Type - “Analog space
2nd 0 (catalog) up until you get to the = sign
Then type ”,A
Enter
Prgm, HUB, 4: Wait, type 1, enter
Prgm, CTL, 7: End

```
NORMAL FLOAT AUTO REAL RADIAN MP

PROGRAM:TEMP
:Send("CONNECT ANALOG.IN
TO BB 5")
:For(N,1,50)
:Send("READ ANALOG.IN 1")
:Get(V)
:Disp "ANALOG IN=",V
:Wait 1
:
```

Display values with an appropriate string and add a wait command to take a one second pause in the readings.

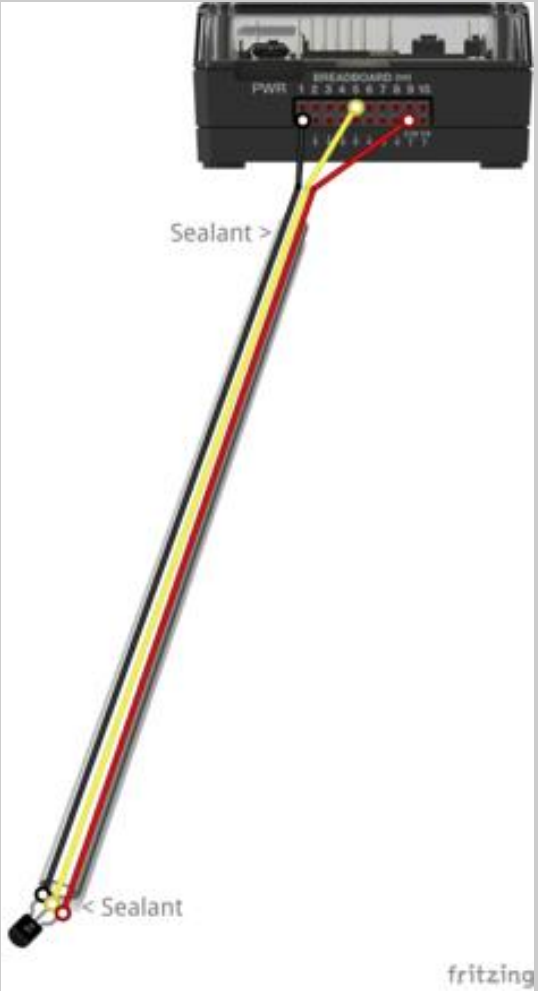
Menu->6:I/O->Disp “ANALOG IN=”,V
Menu->8:Hub->4:Wait 1

```
1.1  1.2  1.3  *TG P5 PR...RAM   RAD
* voltage                        6/7
Define voltage()=
Prgm
Send "CONNECT ANALOG.IN 1 TO BB 5  "
For n,1,50
  Send "READ ANALOG.IN 1"
  Get V
  Disp "ANALOG IN=",V
  Wait 1
EndFor
EndPrgm
```

# Innovator Design:

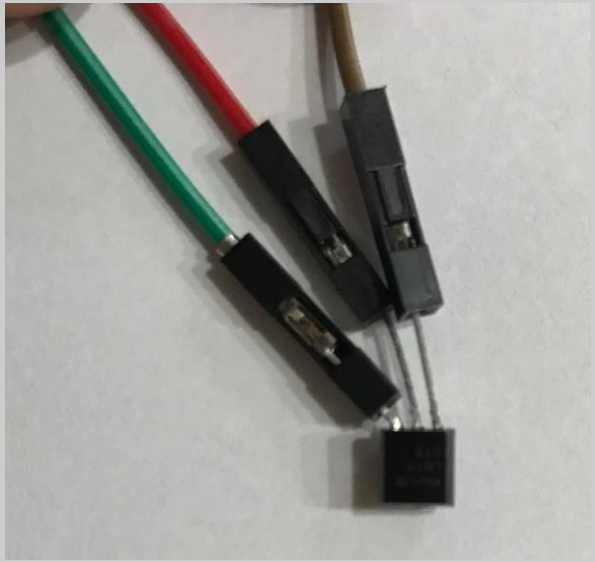
Materials:

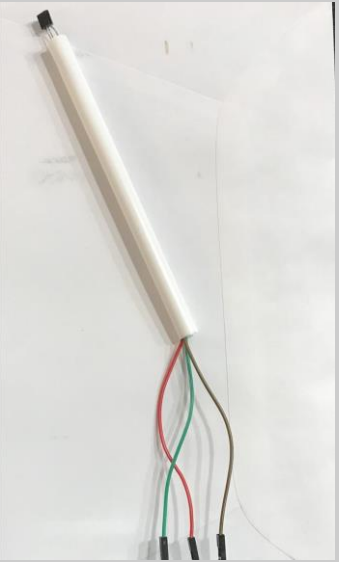
- TI -84 Plus CE / TI NSpire CX
- TI- Innovator Hub with USB Cable
- TI Innovator Breadboard
- Male to Female jumper wires (3 different colors)
- One analog temperature sensor



# Innovator Design:

Connect the three leads from the temperature sensor into the female ends of the jumper wires. Then run them through a protective sleeve, this is a pen case, a drinking straw is fine as well. Glue the top closed making it waterproof.



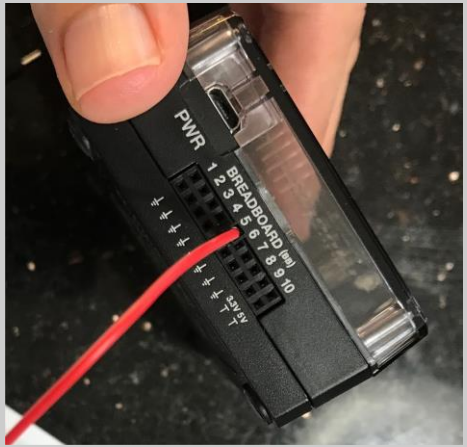


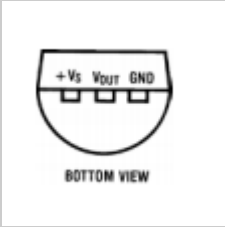
# Innovator Design:

The middle wire (red in this set up), then goes into the BB5 port we coded to connect to.

Connect the GND in to the a ground port (brown lead into the ground port below BB 1), and the +Vs lead into the 3.3V port (green lead below BB 9).

The B side of the mini cable then goes into the DATA port on the opposite side of the Hub, and the A in to your device with the program to be run.



# Run the program:

- Prepare an ice bath, place the analog temperature sensor in the water, and collect the Analog.in value at $0^{o}$ C, then in boiling water and collect at $100^{o}$ C.

- Use the two ordered pairs of Temperature vs. ANALOG.IN values to determine the slope and y-intercept, calibrating the sensor to the Voltage value.

- Check for accuracy with a new program set to use the linear equation you have created to determine the temperature in the room and the degree of accuracy.

# Program 2 (84-CE on left, NSpire CX on right):

Connect Hub to Breadboard (BB) from analog port  and prompt (request) values from the program user.
Prgm, I/O, 2: Prompt.
You are asking for input for the slope, M and y intercept B from program 1

```
NORMAL FLOAT AUTO REAL RADIAN MP

PROGRAM:TEMP2
:Send("CONNECT ANALOG.IN
TO BB 5")
:Prompt "M=?",M
:Prompt "B=?",B
:
```

Connect Hub to Breadboard (BB) using analog port  (same command as first program) and requesting values from the program user.

Menu->6: I/O-> 2: Request

You are asking for input for the slope and y-intercept from program 1.

```
1.1   *Doc   RAD
* voltage2                    4/4
Define voltage2()=
Prgm
Send "CONNECT ANALOG.IN 1 TO BB 5"
Request "M = ?",m
Request "B = ?",b
EndPrgm
```

# Program 2 (84-CE on left, NSpire CX on right):

Connect Hub to Breadboard (BB) from analog port

Create a for loop with 200 readings to measure the voltage at room temperature. Using the slope and y-intercept from part one, the program will then calculate the temperature and display this on the screen.

```
NORMAL FLOAT AUTO REAL RADIAN MP

PROGRAM:TEMP2
:Send("CONNECT ANALOG.IN
TO BB 5")
:Prompt "M=?",M
:Prompt "B=?",B
:For(N,1,200)
:Send("READ ANALOG.IN 1")
:Get(X)
:Disp "ANALOG IN=",X
:M*X+B→C
```

```
NORMAL FLOAT AUTO REAL RADIAN MP

PROGRAM:TEMP2

:Get(X)
:Disp "ANALOG IN=",X
:M*X+B→C
:Disp "TEMPERATURE IN C=",
C
:Wait 1
```

```
1.1   *Doc   RAD
* voltage2                5/12

For n,1,200
  Send "READ ANALOG.IN 1"
  Get x
  Disp "ANALOG IN = ",x
  m·x+b→c
  Disp "Temperature in C = ",c
  Wait 1
EndFor
EndPrgm
```

# Percent Error and Extras

Use a standard thermometer and the temperature from the program to determine a percent error.

Extensions - determine the error range for the class using a list from Navigator

Add lines to the program to request the temperature unit desired, and to then display the converted temperature.

Work backwards from a known temperature to determine what the expected voltage would be, run the program to test.

Create a heating or cooling curve from the data.

# Resources:

TI Codes: https://education.ti.com/en/activities/ti-codes

TI Innovator and I/O Packs: https://education.ti.com/en/products/micro-controller/ti-innovator

Path to STEM Projects:

https://education.ti.com/en/activities/stem/path-to-stem