Learning by Making

UNIT 3

Doing Science with Sensors

Starting a BasicBoard Project

Starting a new BasicBoard project

Answer each question as shown below. If you do not see a black full screen window with questions,
(1) click on the **black icon** at the bottom of your screen,
(2) type `start` and
(3) hit the **enter** key.

Terminal

```
Would you like to load an existing
experiment? (y/n)
> n

Would you like to create a new experiment
(y/n)
> y

Please select an experiment by typing its complete name.
List of experiments
> BasicBoard.tar

Provide a name for the new experiment folder.
> BB-YourName

Please select version by typing its complete name.
PacketDemo Sensors
> Sensors
```

At the end of the questions and answers, you should see a new window pop up in the bottom left corner of your computer screen (and the question screen will disappear).

Starting a previously created BasicBoard project

Answer each question as shown below. If you do not see a black full screen window with questions,
(1) click on the **black icon** at the bottom of your screen,
(2) type `start` and
(3) hit the **enter** key.

Terminal

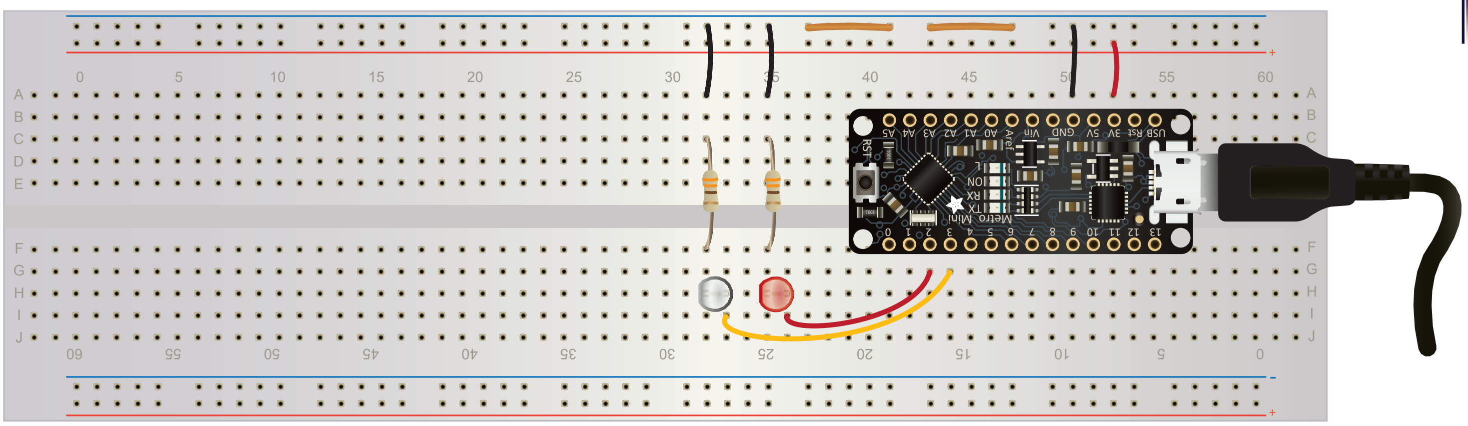
```
Would you like to load an existing experiment? (y/n)
> y

Which experiment would you like to load?
> BB-YourName
```

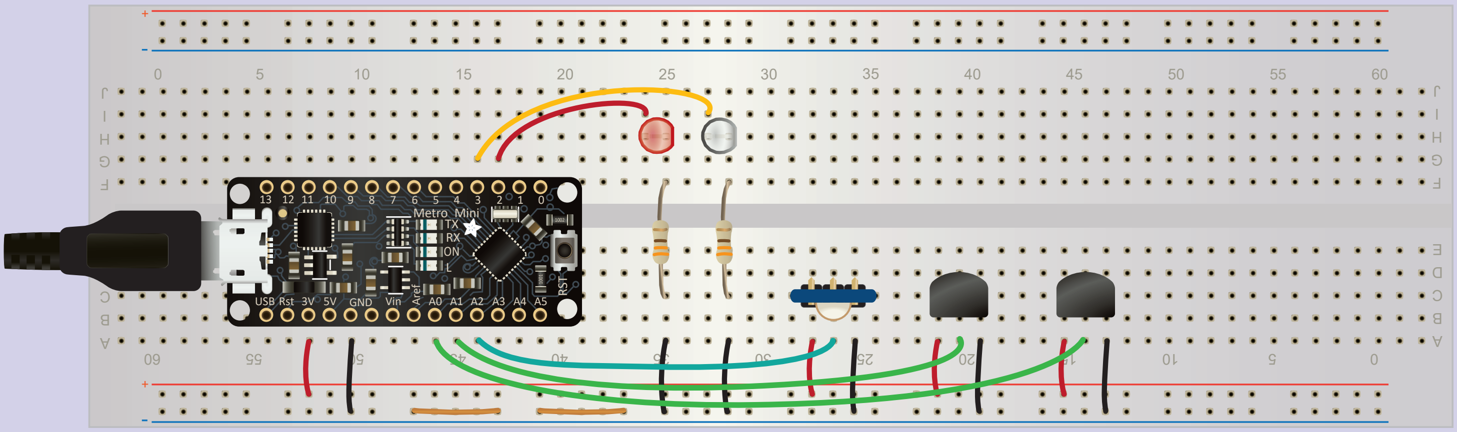
At the end of the questions and answers, you should see a new window pop up in the bottom left corner of your computer screen (and the question screen will disappear).

Components and Material

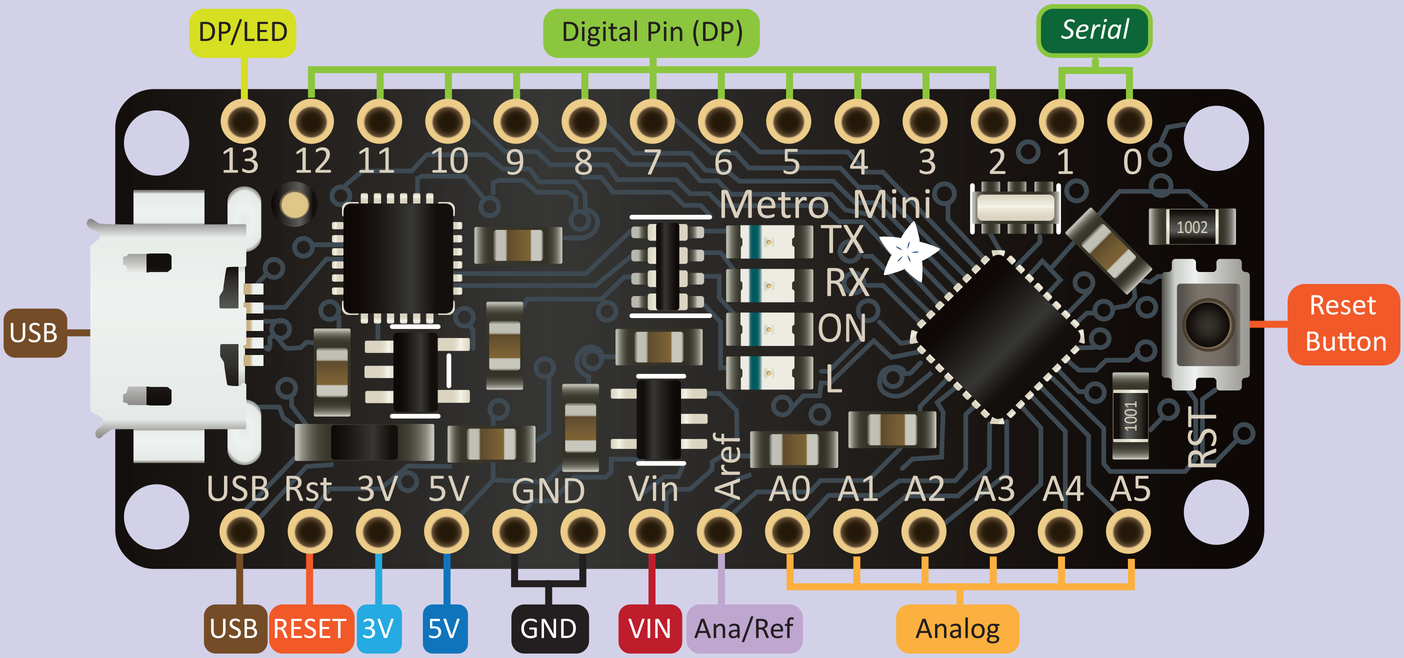
BasicBoard (as delivered)



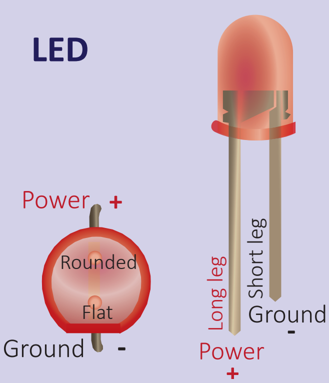
BasicBoard with Light and Temperature sensors



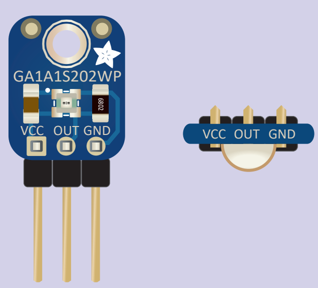
Arduino



LED

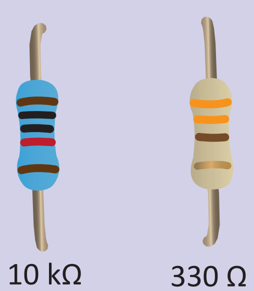


OPIC Analog Light Sensor

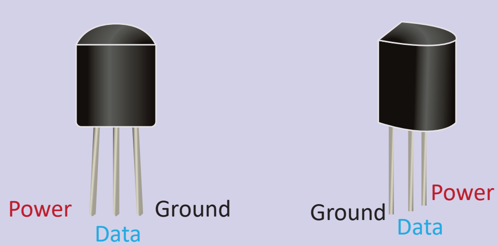


Multimeter

Resistors



Temperature (TMP36) Sensor



Promise to Stay Grounded

Electrostatic discharge can and will damage sensitive electronic compontents like the application board. Simple acts like getting up from a chair, walking around on a carpet, or touching objects can induce a charge on your body.

What can you do to avoid electrostatic discharge?

- **Keep your components in antistatic bags until you are ready to use them**
- **Wear cotton clothing rather than wool or synthetic fibers**
- **Avoid shuffling your feet on carpet**
- **Touch unpainted metal with a clear grounded path**

The threshold for you feeling a small shock is much higher than that of the equipment we will be using in this classroom. It is for this reason that you must shed excess electrons from your body before working with the BasicBoard.

You must **GROUND** yourself.

I, ____________________, hereby promise to ground myself before working with sensitive electronics. I will touch the copper loops on the BasicBoard with both my feet firmly on the floor on a regular basis.

________________________________ ______________

Signature Date

3.1 Sensing Light

Learning Objectives:

- Students will identify the light sensing component of the OPIC light sensor
- Students will learn how to wire up the OPIC Analog Light Sensor

Materials

For each group of 2 students

- Basicboard with Arduino and 2 LEDs wired to the Arduino
- Computer
- Wire stripper and cutter
- Wire as needed
- OPIC analog light sensor

For each student

- Worksheet
- Challenge sheet
- Double Dare Sheet

Vocabulary:

- OPIC Analog Light Sensor

Tasks you need to perform

Answer questions in your Worksheet, Challenge & Double Dare sheets

Getting Started

In unit 2, you learned how to use the Arduino to turn on and off LED lights. The computer sent signals to the Arduino, which changed the voltage output from digital pins connected to the LEDs.

In this lesson you will learn how to read signals output from **sensors** connected to the Arduino. Sensors are the key to many future experiments. Sensors allow you to probe, measure, investigate, explore, and - ultimately – understand the world around you.

Instructions

STEP 1. Set up the computer, BasicBoard, and gather materials

Gather the materials in the materials list. Turn on your computer and plug in the USB cable to the computer and the BasicBoard.

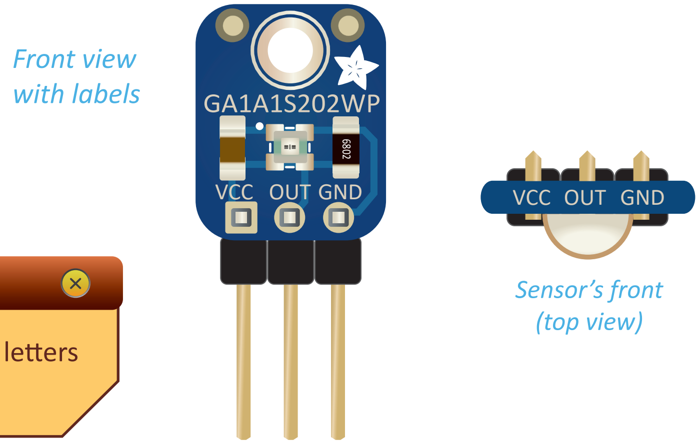
On your worksheet, write down each item on the materials list you have in front of you. If you notice you need to gather more materials, do that now and write them down.

STEP 2. Examine the OPIC analog light sensor

The OPIC analog light sensor converts light into a voltage. The voltage that is generated changes with the intensity of the light. When the light intensity increases, the light sensor will generate a higher voltage. When the light intensity decreases, the light sensor will generate a lower voltage.

H

Hold the OPIC sensor so that the letters GA1A1S202WP are facing you.



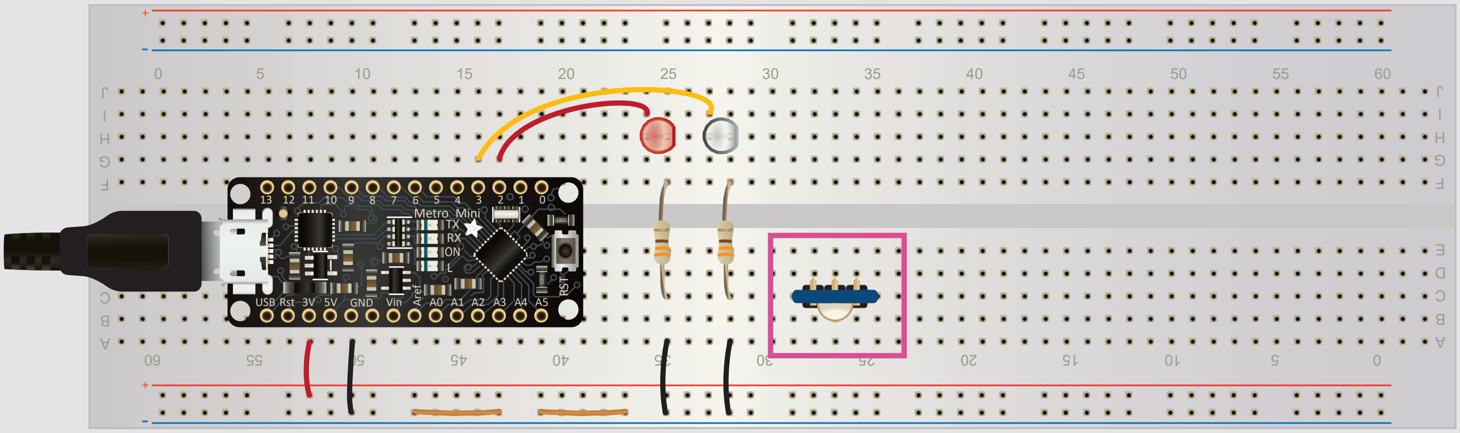
On your worksheet, label the drawing of the light sensor with your best guesses as to the functionality of the three legs of the sensor. Also indicate your prediction for the part of the sensor that you think is sensitive to light.

STEP 3. Wire up the OPIC light sensor

Plug the light sensor into the BasicBoard somewhere in the region shown below. The labeled side of the sensor should be facing the power and ground rails.

Cut and strip short pieces of red and black wires. Use the red wire to connect the pin on the light sensor labeled Vcc to **power**. Use the black wire to connect the pin labeled GND to **ground**.

Cut and strip a longer piece of wire of any other color. Use this wire to connect the pin labeled OUT to the Arduino pin labeled A2.



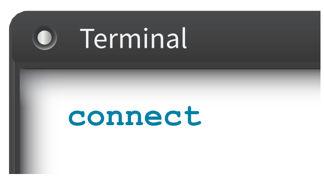
Front Side

On your worksheet, indicate where you have placed the sensor and draw the wire connections. Also, show your wired sensor to a classmate. If they agree that you have wired the sensor correctly, have them initial your worksheet.

STEP 4. Start up your BasicBoard experiment and connect the Arduino to the computer

Turn on your computer and plug in the USB cable from the BasicBoard. Follow the instructions on page 3 "*Starting a New BasicBoard Experiment*".

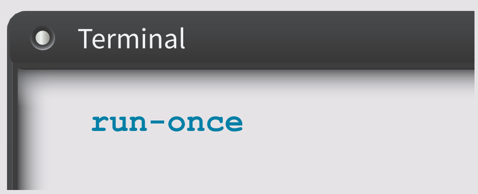
At the end of the questions and answers, you should see a new black terminal window pop up in the bottom left corner of your computer screen (and the question screen will disappear). Type `connect` in this window.



On your worksheet, write down the name you gave your Experiment file for future reference.

STEP 5. Run the light sensor

Type `run-once` in the black terminal window.



On your worksheet, describe any lights that turn on or off, and then record the number that appears on the screen.

STEP 6. Complete the challenges

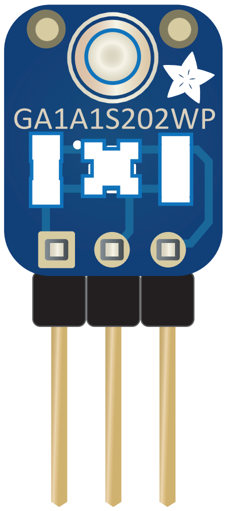
Now that you have learned how to get readings from the light sensor, please continue on to complete the challenges for this lesson.

Name:____

STEP 1 Write down each item on the materials list you have in front of you. If you notice you need to gather more materials, do that now and write them down.

STEP 2 Label the drawing of the OPIC sensor with your best guesses as to the functionality of the three legs of the sensor.

Also indicate your prediction for the part of the sensor that you think is sensitive to light.



____

____

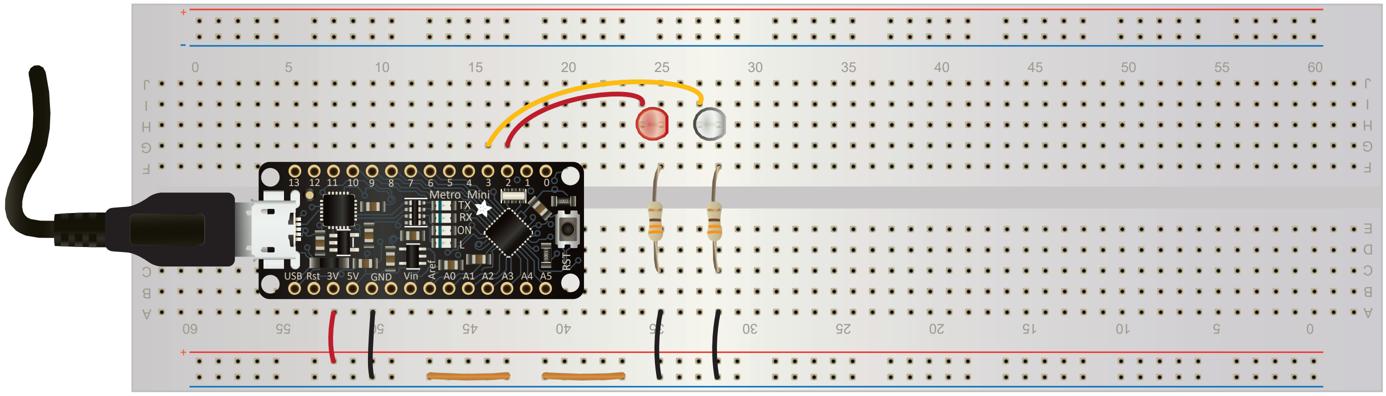
____

____

____

Name:____

STEP 3 Indicate where you have placed the sensor and draw the wire connections. Label the wires with the light sensor pin labels. Also, show your wired sensor to a classmate. If they agree that you have wired the sensor correctly, have them initial your worksheet.



Initials: __________

STEP 4 Write down the name you gave your Experiment file for future reference.

STEP 5 Describe any lights that turn on or off, and then record the number that appears on the screen.

____

____

____

____

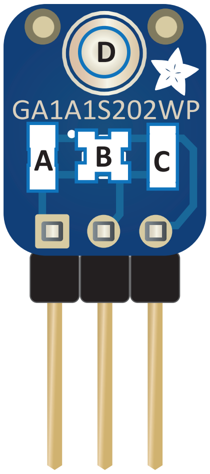
____

____

Name:____

C-1

Investigate the OPIC light sensor to determine what part of the sensor actually does the sensing. Use the drawing below to complete your claim as to which component senses the light. Then finish the sentences describing the evidence you have for this claim and the reasoning why the evidence supports your claim.



Claim: The component that ____________________________ is: A, B, C or D (Circle one)

Evidence: The Average Light Value started as __________ before I ____________________________

After I _______________________________, the Average Light Value was _____________

Reasoning: The Average Light Value _______________________ therefore I conclude ____________ ______________________________

C-2

Create and test a plan to answer the question: what happens to the Average Light Value number when the amount of light shining on the sensor increases or decreases? Describe your plan.

____

____

____

____

In the table, write what you did and the Average Light Value for each action. Include the light source or blocking material you used.

Action Taken	Average Light Value Number

Name:____

D-1

Dynamic range is a term often found in descriptions for cameras, televisions, and images. Dynamic range is the difference between the darkest and the lightest tones or colors.

Investigate the dynamic range of the OPIC Analog Light Sensor. Is there a point at which the output signal saturates? Is there a point at which there is not enough light to generate a signal?

D-2

Design an investigation to answer the scientific question:
How does the light sensor react to different light brightness?
After your investigation, make a claim about how the light sensor works with light sources of different brightness. Write down your evidence and reasoning for the claim that you made, using the evidence you collected in your investigation.

Hint: Gather light bulbs of different known brightness. Keep the distance between the light bulbs and the light sensor. Measure the ADU values. Create a table with the results.

3.2 Light Sensor Readings

Learning Objectives:

- Students will learn how to use a digital multimeter to investigate sensor signals
- Students will model the mathematical relationship between analog sensor readings and digital BasicBoard outputs

Materials

For each group of 2 students

- Basicboard and OPIC light sensor wired to Arduino
- Computer
- Multi-meter

For each student

- Worksheet
- Challenge sheet
- Double Dare Sheet

Vocabulary:

- Analog
- Analog to Digital Converter (ADC)
- Analog to Digital Unit (ADU)
- Bit
- Continuous
- Digital
- Discrete
- Integer numbers
- Real numbers

Tasks you need to perform

Answer questions in your Worksheet, Challenge & Double Dare sheets

Getting Started

In the previous lesson you wired the OPIC light sensor to the Arduino so that it could produce digital output numbers. This introduces a new concept:

Digital Versus Analog Signals:

Digital Signals:

Discrete values - on/off, high/low, **integer numbers**

Analog Signals:

Continuous values - time, temperature, **real numbers**

The LEDs use discrete on/off signals as inputs. A sensor, on the other hand, may output a wide range of voltages. It is the Arduino that converts these voltages into the digital numbers that were printed out on the computer screen. In this lesson, you will investigate how the Arduino (a digital device) communicates with the light sensor (an analog device).

Instructions

STEP 1. Set up computer, BasicBoard and gather materials

Gather the materials in the materials list. Turn on your computer and plug in the USB cable to the computer and the BasicBoard.

On your worksheet, write down each item on the materials list you have in front of you. If you notice you need to gather more materials, do that now and write them down.

STEP 2. Restart your BasicBoard experiment and connect.

Reload your existing project with the name you chose in the previous lesson. Then connect the Arduino to the computer.

H

If you can't remember the name you gave to your project in the previous lesson, refer back to your notes.

STEP 3. Use multimeter to measure voltage from sensor

In order to use the digital multimeter to measure voltages that are output from the light sensor:

a) Use a test wire (a short stripped wire of a different color: blue, green or yellow) with your multimeter to measure the voltage going into the digital pin that is connected to the light sensor. Connect a short black wire to the ground rail.

b) Type `run-forever` in the black project window to start collecting data.

Terminal

```
run-forever
```

On your worksheet, Indicate the dial settings on the diagram of the multimeter. Then sketch where you put the light sensor and where you are placing the red and black probes.

> H
>
> Review how to strip wire from UNIT 2, Lesson 2.2, step 2 and use a multimeter in step 4.
>
> To stop the `run-forever` process, type `..` and hit Enter. It might take more than one try.

STEP 4. Compare the voltage measurements to the digital outputs

Keep the multimeter probes in the locations that you identified in the previous step. The goal of this step is to change the amount of light shining on the sensor and to compare the voltages from the multimeter to the digital numbers output on the computer screen. Note: These digital values are known as Analog to Digital Units (ADUs).

a) Choose a light level that can remain steady. For example, a flashlight at a specific distance from the sensor.
b) Measure the voltage with the multimeter.
c) Use `run-once` to collect the average digital number for this light level.

On your worksheet, fill out the table with the results of your experiments. Then repeat one more time with a different light level. Also take a measurement with the sensor completely covered.

STEP 5. Compare the voltage measurements to the digital outputs

How do the ADU values on A2 vary as we vary the Voltage to that pin? We can explore the answer to this question by gathering evidence through an experiment procedure. Disconnect the wire from the central pin of the light sensor that goes to the A2 pin, but leave the other end of the wire attached to the A2 pin. Note the ADU numbers that result when you connect the free end of the wire to the following locations:
a) 0 V ground rail b) 3.3 V power rail c) 5 V pin on the Arduino

On your worksheet, fill out the table with the results of your experiment procedure.

> H
>
> Consult the Arduino diagram on page 5 to find the 5V pin.

STEP 6. Graph the relationship between voltage and digital outputs

To discover the mathematical relationship between voltage and digital outputs, use the table you created in the previous step to fill in the graph on the worksheet.

On your worksheet, graph voltage (x-axis) vs. digital output (y-axis). Is the relationship proportional or inversely proportional?

> H
>
> Remember, proportional means that x and y either both increase or both decrease. Inversely proportional means that x increases if y decreases or vice versa.

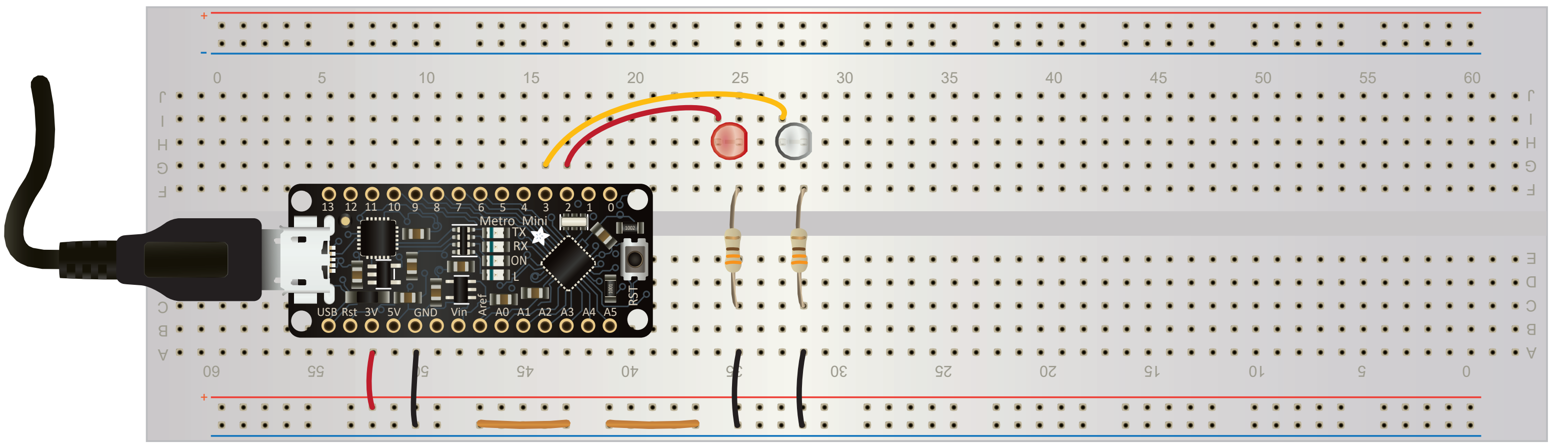
STEP 7. Complete the challenges

Now that you have investigated the relationship between voltage and numerical output, continue on to the challenges for this lesson.

Name:____

STEP 1 Write down each item on the materials list you have in front of you. If you notice you need to gather more materials, do that now and write them down.

STEP 3 Indicate the dial settings on the diagram of the multimeter. Then sketch where you put the light sensor, the power, output, ground wires, the test wire and where you are placing the red and black probes.



Name:____

STEP 4 Fill out the table with the results of your experiments. Then repeat one more time with a different light level. Also take a measurement with the sensor completely covered.

Experiment	Voltage Reading	Digital Output
Sensor covered		

STEP 5 Fill out the table with the results of your experiment procedure.

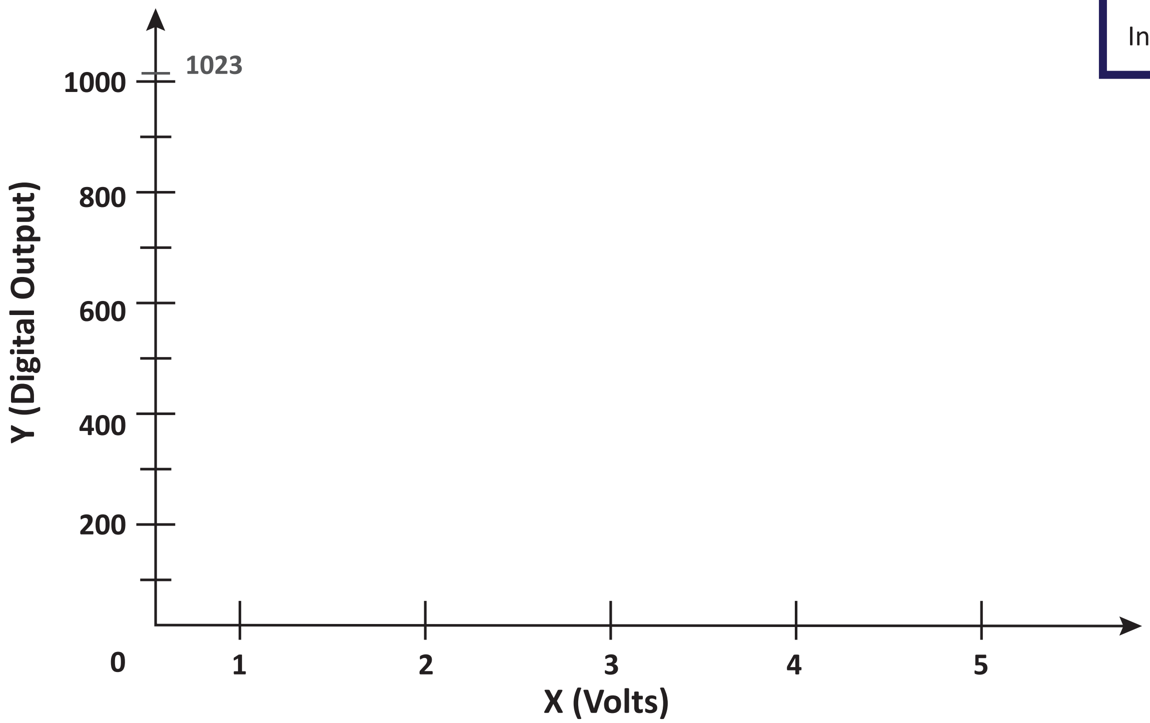
Experiment	Voltage Reading	Digital Output
A2 connected to ground	0 V	
A2 connected to 3.3 V	3.3 V	
A2 connected to 5 V	5 V	

STEP 6 Graph voltage (x-axis) vs. digital output (y-axis). Is the relationship proportional or inversely proportional?

Circle your answer:

Proportional

Inversely proportional



3.2 Challenges

Light Sensor Readings

Name:____

C-1

When you typed `run-forever` in the Terminal window, you saw examples of discrete digital output values, known as **Analog-to-Digital Unit (ADU)** numbers. Type `run-forever` in the Terminal window. Write down the evidence you notice in these numbers that suggest that the ADU numbers are discrete digital values.

Claim: ADU numbers are discrete digital values

Evidence: the ADU numbers are ____

Reasoning: ______________________________ are discrete values. Since ADU numbers are

_________________________, therefore ADU numbers are ______________________________.

C-2

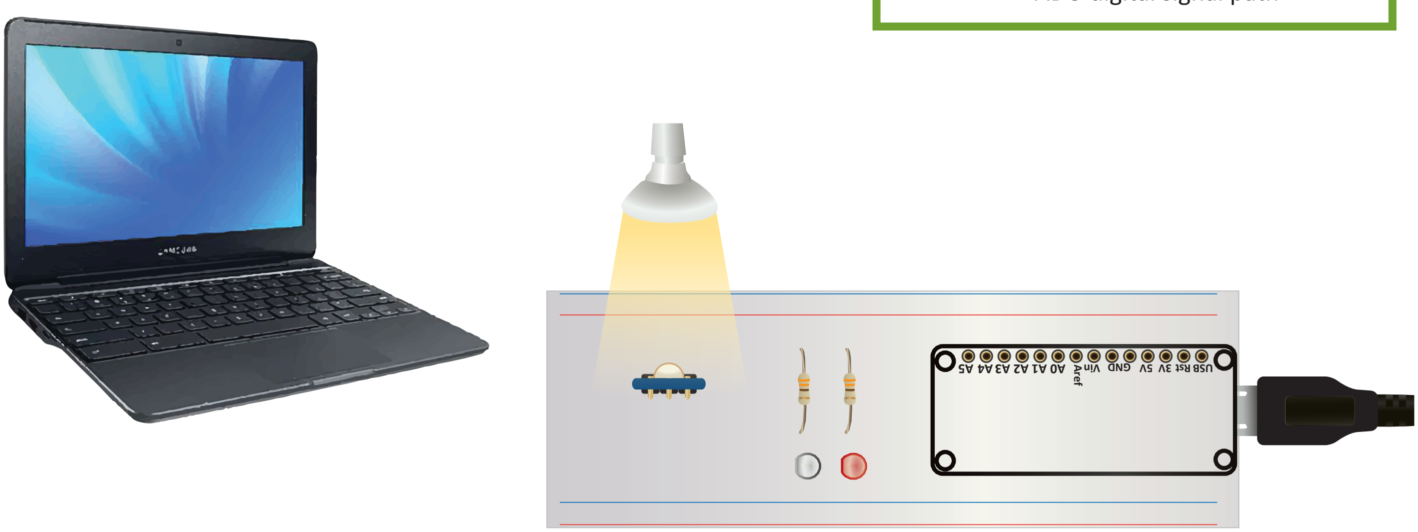
Using the Path Legend, draw the flow of information from the light to the light sensor, through the Arduino, and to the computer. Label where the signal gets converted from analog to digital.

Path Legend

Light path

Voltage signal path

ADU digital signal path



Names:____

C-3

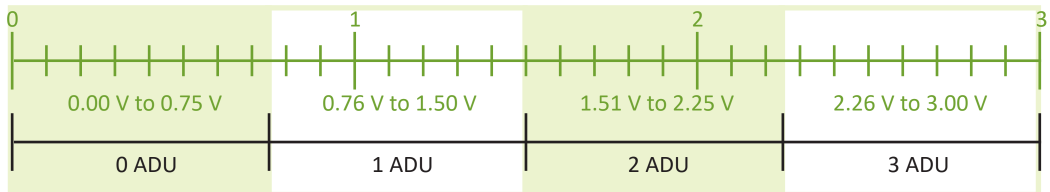
All the Arduino pins labeled “A” are known as analog pins. They are attached to an Analog-to-Digital Converter (ADC). ADCs convert continuous analog voltage signals into discrete ADU values. In the example below, the light sensor converts light into voltages in the range 0 to 3 V. The ADC in this example only has 4 ADU values available: 0 ADU, 1 ADU, 2 ADU, and 3 ADU.

We can use a number line, like the one below, to understand the relationship between a range of voltages and each discrete ADU number. Use the number line to fill in the ADU values for each voltage in the table below.

Voltage	ADU
0.5 V	
1.0 V	
2.0 V	
2.5 V	

Example :
ADC with 4 values reads an analog sensor with 0 V to 3 V output

Voltage output
Units: volts



ADC reading
Units: ADU

C-4

In C-3, the example ADC used 2 bits and had only four ADU values available.
A bit is a number that can only be 0 or 1 (see example to the right).

If we want a larger range of digital numbers, we need more bits! Use the information provided in C-3 & C-4 to fill out the table for a 3-bit ADC and also figure out the voltage range for each ADU where the voltage range is from 0 to 3 V.

Two bits allows for us to make 4 numbers:
0 = 0 0
1 = 0 1
2 = 1 0
3 = 1 1

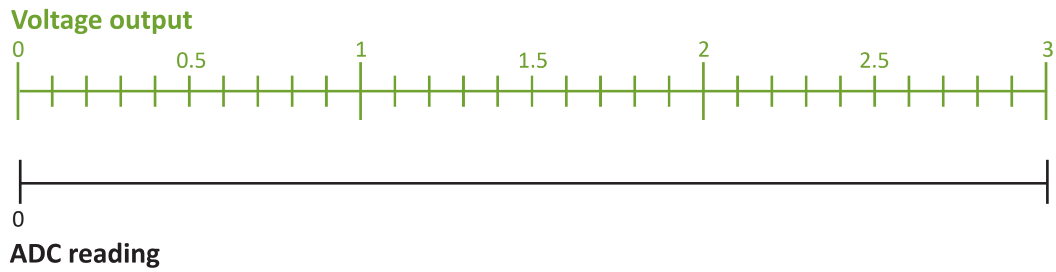
Decimal Number	3-bit binary number	Voltage range
0	000	0 V to 0.375 V
1	001	0.376 V to 0.75 V
2	010	0.751 V to 1.125 V
3	011	
4	100	
5		
6		
7		

Names:____

C-5

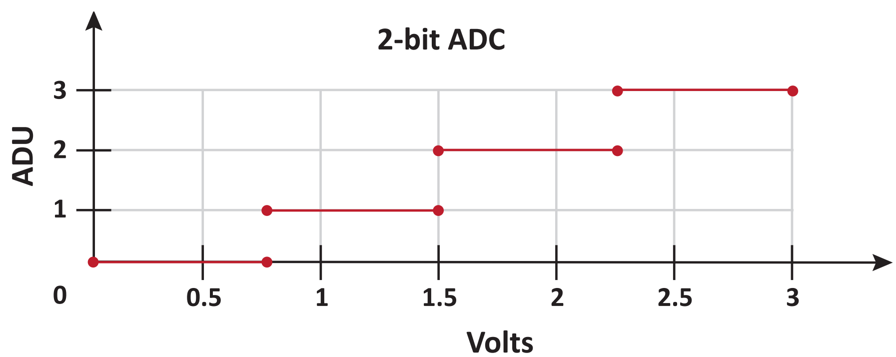
Draw a number line (similar to C-1) that shows the relationship between the voltage ranges and the ADU for that range for the 3-bit ADC table in C-4.

ADU	Voltage range
0	
1	
2	
3	
4	
5	
6	
7	

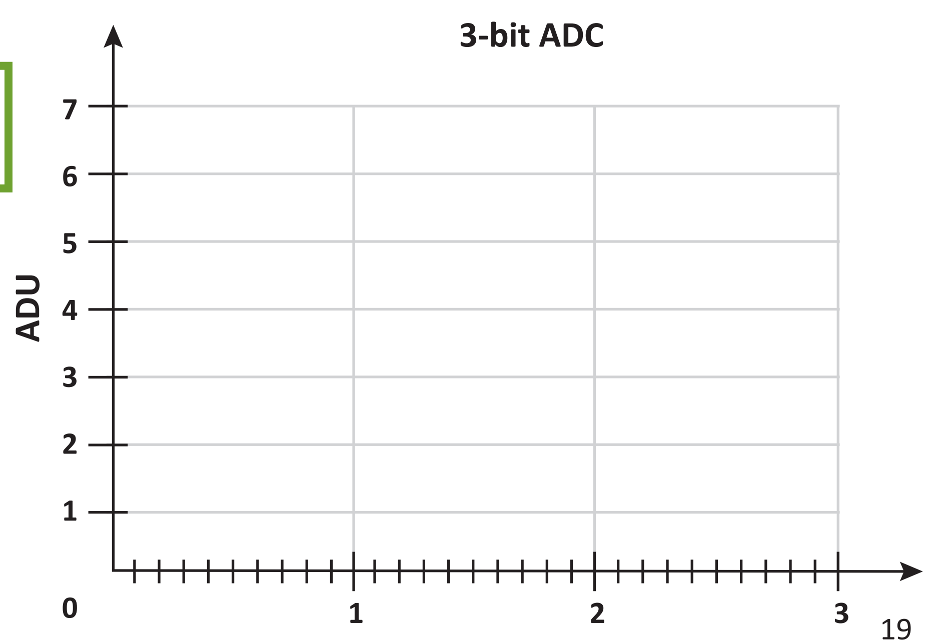


C-6

Another way to visualize the relationship between ADUs and voltage ranges is to use a graph. The illustration shows a 2-bit ADC that covers the range 0 to 3 Volts (as in C-3)



Draw a similar graph for the 3-bit ADC case that you worked out in C-3.



Names:____

D-1

The Arduinos actually used in this class output 1024 possible values for ADC pin readings – the numbers 0 through 1023. The OPIC sensor outputs signals ranging from 0 Volts to 5 Volts.

What is the sensitivity limit of your BasicBoard? Determine the smallest change in voltage needed to change an ADU value. This can be done mathematically. Show your work and explain your calculations.

D-2

Looking at the table, derive a mathematical relationship between the number of bits in the ADC and the number of available ADUs.

What is the relationship between the number of bits in the ADC and the highest number in the ADU range?

Hardware	Available ADUs	ADU range
2-bit ADC	4	0 to 3
4-bit ADC	16	0 to 15
8-bit ADC	256	0 to 255
10-bit ADC	1024	0 to 1023

3.3 Build a Night Light

Learning Objectives:

- Students will learn how to use `Logo` to evaluate arithmetical expressions using `+, -, /` and `*`
- Students will learn how to use the `Logo` words `<, >` and `=` to compare numbers
- Students will learn how to use the `Logo` word `if` to make decisions
- Students will be able to modify code to turn on an LED based on light sensor signals

Materials

For each group of 2 students

- BasicBoard with Arduino, 2 LEDs and OPIC light sensor
- Computer
- Wire, wire strippers and cutters as needed
- OPIC analog light sensor wired to the Arduino

For each student

- Worksheet
- Challenge sheet
- Double Dare Sheet

Vocabulary:

- Conditional
- Operator

Tasks you need to perform

Answer questions in your Worksheet, Challenge & Double Dare sheets

Getting Started

When the Sun goes down Earth and its inhabitants respond. Biological, chemical, and mechanical reactions to light happen all around you.

Some creatures wake up. Some go to sleep. Plants will bloom for nocturnal visitors or fold up protectively. Streetlights and headlights shine brighter to guide your travels. Smartphone screens dim to protect your vision.

In this lesson, you will create your own reactionary device. You will build a tool that responds to light levels in real time – a night light.

Instructions

STEP 1. Set up computer, BasicBoard & restart your experiment

Gather the materials in the materials list. Turn on your computer and plug in the USB cable to the computer and the BasicBoard. Reload your existing project with the name you chose in the previous lesson. Then `connect` the Arduino to the computer.

On your worksheet, write down the UNIX time after `Welcome to Logo.`

If you can't remember the name you gave to your project in the previous lesson, refer back to your notes.

STEP 2. Investigate the arithmetical operators + – / and *

Logo knows how to use arithmetical **operators** such as + (addition), - (subtraction), / (division) and * (multiplication). In order to see the numerical results, you must use the **Logo** word **print**.

Terminal
```
print 1 + 2
```

For each expression in the table on the worksheet, first predict the result that you expect. Then type the expression into the terminal window and hit the enter key. Record the results in the table on your worksheet.

H

If you get an error message, make sure to leave spaces before and after each Logo word, including >, <, and the arithmetic operators.

STEP 3. Investigate the comparison operators < > and =

Logo has several different operators that can be used to compare two numbers or algebraic expressions. These words include < > and =. These are logical operators, so if the comparison is true, **Logo** will respond **true**, and if it is not true, **Logo** will respond **false**. In order to see the numerical results, you must use the Logo word **print**.

Terminal
```
print 1 < 2
```

For each expression in the table on the worksheet, first predict the result that you expect. Then type the expression into the terminal window and hit the enter key. Record the results in the table on your worksheet.

STEP 4. Recall how to use variables

In Lesson 1.5, you were introduced to variables in **TurtleLogo.** Remember that after you assign a descriptive name to the variable, to get the value out of the variable, you put a colon in front of it. In this lesson, we will use the **Logo** word **let** to assign values to a variable.

To see how this works, in the terminal window, type the following:

Terminal
```
let [ testvar 6 ]
print :testvar
```

On your worksheet, write what happens after you type in the code.

STEP 5. Examine the Logo code for `run-forever`

Open the **Logo** code for this lesson by typing `edit-project` in the terminal window. Locate the code for the word `run-forever`. In the previous lesson, you used this code to read values from the OPIC light sensor. The code in between the square brackets [ ] that follows the word `loop` runs continuously until interrupted.

Add comments to the code to explain what happens on each line. Then save the file. Document the comments on your worksheet.

Terminal

```
to run-forever
     OB1on
     wait 10
     OB1off

loop
     [
          let [light readLightSensor]
          print :light
          wait 10
     ]
end
```

Remember that putting a semi-colon in front of a line comments out all the text that follows, and this text will not be executed.

Holding down the keyboard keys **crtl + F** allows you to do a search.

`readLightSensor` returns the ADU value from the OPIC light sensor

STEP 6. Learn how to use the Logo word `if`

Logo has several different **conditional** words. Conditionals are words that are used to test if a condition is true (or not). If the condition is true, the program runs the code inside the square brackets [ ]. If the condition is false, the code inside the brackets is not executed.

Type the following into the terminal window:

Terminal

```
if :testvar < 5 [ print :testvar ]
if :testvar > 5 [ print :testvar ]
```

The value for testvar is set in Step 4

On your worksheet, predict what will happen after you type in each line of code, and then write what actually happened.

STEP 7. Complete the Challenges

Now that you have learned how to use comparison and conditional words in **Logo**, please continue on to complete the challenges for this lesson.

Name:____

STEP 1 Write down the UNIX time after `Welcome to Logo.`

STEP 2 For each expression in the table below, first predict the result that you expect. Then type the expression into the terminal window and hit enter. Record the results in the table

Expression	Prediction	Result
`print 1 + 2`		
`print 5 - 3`		
`print 8 / 2`		
`print 12 / 2 * 3`		
`print 12 / ( 2 * 3 )`		
`print 12 + 2 * 3`		

*H: To multiply two numbers, Logo does not use X as that is a letter. Instead, multiplication is done using *. The * symbol is commonly used in all other computer languages to indicate multiplication.*

*H: In **Logo**, multiplication and division take precedence over addition and subtraction. If all the operators have the same rank, then **Logo** evaluates arithmetical expressions from left to right. Use parentheses if you want a different result.*

STEP 3 For each expression in the table below, first predict the result that you expect. Then type the expression into the terminal window and hit enter. Record the results in the table.

Expression	Prediction	Result
`print 1 < 2`		
`print 2 < 1`		
`print 2 / 3  >  4 - 1`		
`print 3 + 5 = 4 * 2`		

Name:____

STEP 4 Write what happens after you type in the code.

____

____

____

STEP 5 Add comments to the code to explain what happens on each line.

Pluma File Edit View Go Bookmarks Help Fri Jul 26, 14:55

BB-YourName.logo (~/BB-YourName) - Pluma

Open Save Undo

BB-YourName.Logo

```
to run-forever
     OB1on
     wait 10
     OB1off

loop
     [
          let [light readLightSensor]
          print :light
          wait 10
     ]
end
```

STEP 6 Predict what will happen after you type in each line of code, and then write what actually happened.

Expression	Prediction	Result
`if :testvar < 5 [ print :testvar ]`		
`if :testvar > 5 [ print :testvar ]`		

Name:____

C-1

The goal of this challenge is to determine an ADU value that will be used to turn on the red LED when the light in the room is low. Experiment with your light sensor, and determine an ADU value that represents the classroom when the lights are turned off. Then experiment again to find an ADU value that represents the classroom light level when the lights are turned on. Explain what you did below.

____

____

____

____

____

____

C-2

Use the value that you found in C-1 to modify the code in `run-forever` so that the red LED will turn on when the light level is low enough but will turn off when the light level is high enough. After the line of code that reads the light sensor value, you will need to insert two new lines of code. First fill in the blanks with the correct values and numbers.

```
if :light < ____________ [ dp2on ]
if :light > ____________ [ _____ ]
```

Then add these two lines of code with the values and numbers. Write the final lines of code below.

____

____

____

C-3

Now that you have filled in the blanks, save your file, then `reload` and `connect`. Then try running your new version of `run-forever`.

Change the light level and prove that the red LED turns on when the classroom lights are turned off. If it doesn't work, check your ADU values, and repeat the process until you have a working night light.

When it works, have a classmate initial your worksheet, and record the ADU values that you used to turn the light on and off. Congratulations! You have built a working night light!

ADU On value: ______________________

ADU Off value: ______________________

Initials: __________

Name:____

D-1

The `run-forever` word tells the Arduino how often it should pause between reading the light sensor. What is the line of code that sets this rate? Change the rate at which your night light reacts to different light levels. Make it faster and make it slower. Write down your steps and code.

____

____

____

____

____

D-2

Create an anti-night light. Change the code so that the LED turns on when the room is bright and turns off when the room is dark. Write down your steps and code.

____

____

____

____

____

D-3

How does the human eye detect light? How does our brain interpret these signals? What are the similarities and differences between our eyes and the OPIC analog light sensor? Research the human eye and write a short report or give a presentation.

____

____

____

____

____

____

____

____

3.4 Sensing Temperature

Learning Objectives:

- Students will learn how to wire and operate a temperature sensor
- Students will model the relationship between voltage and ADU for a temperature sensor

Materials

For each group of 2 students

- BasicBoard with Arduino, 2 LEDS and the OPIC light sensor
- Computer
- Wire, wire strippers and cutters as needed

For each student

- Worksheet
- Challenge sheet
- Double Dare Sheet

Vocabulary:

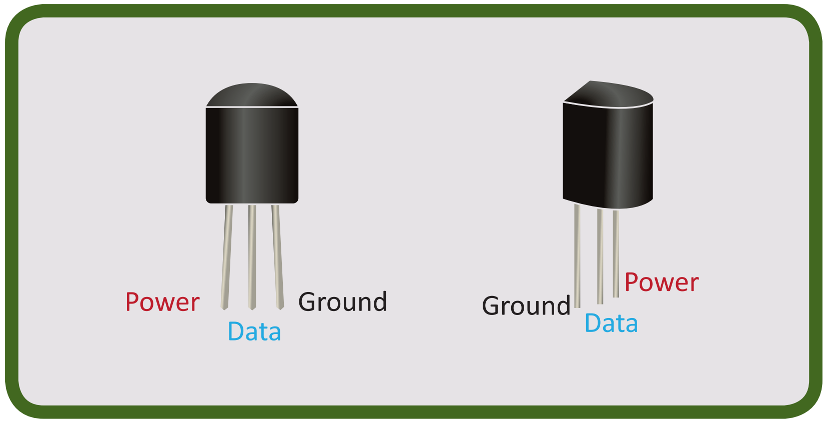
- Block
- Pipe
- String
- Temperature Sensor

Tasks you need to perform

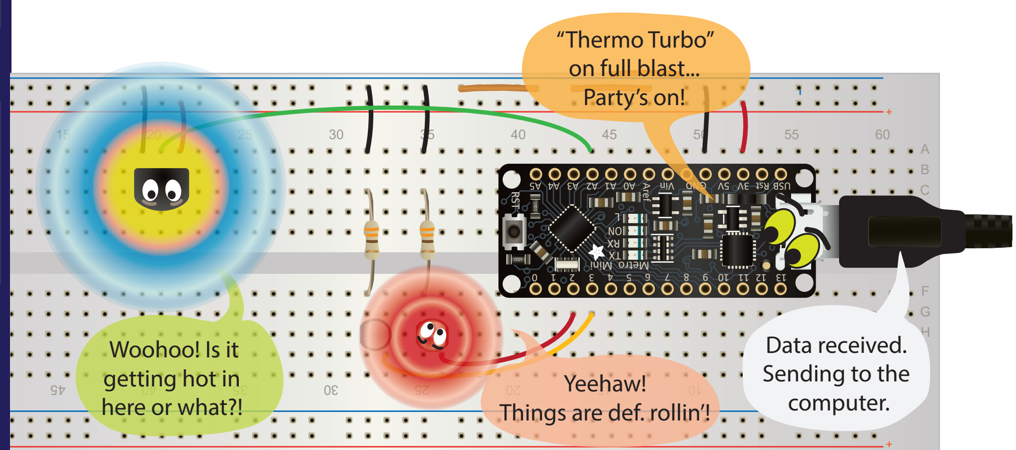
Answer questions in your Worksheet, Challenge & Double Dare sheets

Getting Started

This is the TMP-36 Analog Temperature Sensor. It connects to the Arduino just like the OPIC Analog Light Sensor.



In this lesson, we will wire up two TMP-36 sensors to complete the BasicBoard.



Instructions

STEP 1. Set up the computer and BasicBoard, and gather materials

Gather the materials in the materials list. Turn on your computer but do NOT plug in the USB cable to the computer and the BasicBoard.

On your worksheet, write down each item on the materials list you have in front of you. If you notice you need to gather more materials, do that now and write them down

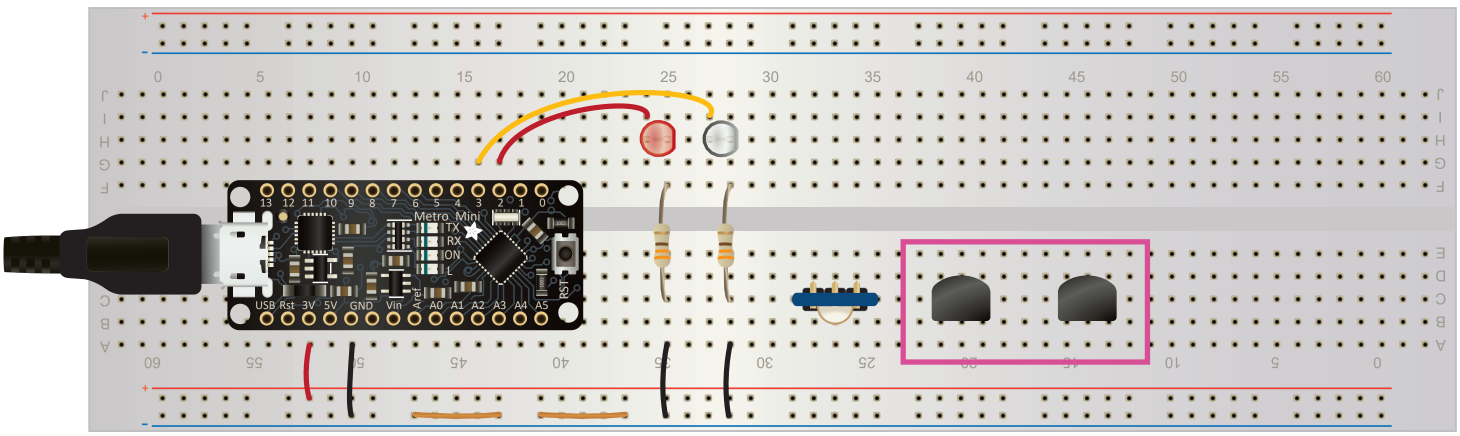
STEP 2. Wire up both TMP-36 sensors

Plug the two TMP-36 sensors into the BasicBoard next to the OPIC light sensor somewhere in the area shown. The flat side of each sensor should be facing the **power** and **ground** rails. Cut and strip short pieces of red and black wires. Use the red wire to connect the power pin to the power rail. Use the black wire to connect the ground pin to ground.

Cut and strip a longer piece of wire of any other color. Connect the data pin of one of the sensors to A0 and the data pin of the other sensor to A1.

Sketch the location and the wires for each sensor on your worksheet. Confirm your wiring with your teacher before proceeding to the next step.

Take care when connecting power and ground. If a sensor is plugged in backwards, it will overheat and fail.



Front Side

STEP 3. Restart your BasicBoard experiment and connect.

Plug in the USB cable. Then reload your existing project with the name you chose in the previous lesson. Then connect the Arduino to the computer.

If you can't remember the name you gave to your project in the previous lesson, refer back to your notes.

STEP 4. Activate code to run the TMP-36 sensors

Use **edit-project** to open the **Logo** code. Examine the code for **run-once**. What steps do you need to take to activate the Sensor readouts so that you can see the outputs for all 3 sensors?

(BB-YourName.logo (~/BB-YourName) - Pluma

File Edit View Search Tools Documents Help

Open Save Undo

BB-YourName.Logo

```
    let[suml 0]
    let[sumt0 0]
    let[sumt1 0]
     repeat 5 [
           let[light readLightSensor]
        let[suml :suml + :light]
     ]
    print (se "|Average Light Value: | :suml / 5)

     ; repeat 5 [
    ;    let[tempsensor0 readTempSensor0]
    ;    let[sumt0 :sumt0 + :tempsensor0]
    ; ]
    ; print (se "|Average Temp0 Value: | :sumt0 / 5)
    ; repeat 5 [
    ;     let[tempsensor1 readTempSensor1]
    ;     let[sumt1 :sumt1 + :tempsensor1]
    ; ]
    ; print (se "|Average Temp1 Value: | :sumt1 / 5)

end
```

> Remember that putting a semi-colon in front of a line comments out all the text that follows, and this text will not be executed.

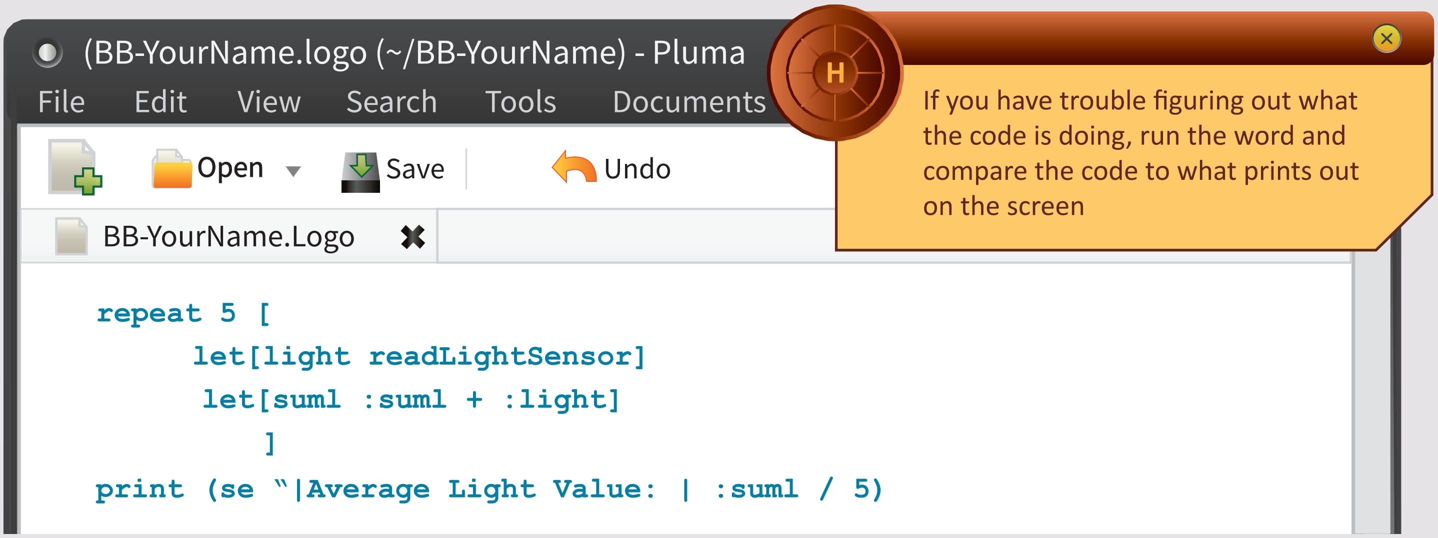
On your worksheet, describe the steps you took that enabled printing the temperature sensors' outputs on the screen when you **run-once**.

> Remember to save your changes and then **reload** and **run-once** to verify your new code.

STEP 5. Examining the `run-once` code

Each of the three sensors on the BasicBoard is read out by some lines of code that are very similar. We call groupings of code "**blocks.**" The goal of this step is to understand what is happening in each of these blocks of code.

Here is the block of code that prints out the light sensor output:



On your worksheet and in the **`BB-YourName.logo`** file, add comments that explain what is happening on each line of code. Be sure to save your code after adding the comments. Then reload and connect to try out the word.

STEP 6. Deconstructing the `print` statement

In the block of code that you examined, there was a print statement that had some new **Logo** words The **Logo** word **`se`** means "sentence". This word combines all the information inside the parentheses into one line (sentence) that will be printed out on the screen. Some of the things that are printed out are numbers, while others are text.

A section of text, including spaces, is called a **string**. Because **Logo** uses spaces to separate words, if you want to print out a string that includes spaces, you need to use vertical bars **`|`** ("**pipes**") to start and end the string. And you use a **`"`** to tell Logo that the items inside the pipes represent a string.

sentence

```
print (se "|Average Light Value: | :suml / 5)
```

Starts a string

Beginning and end of the string

On your worksheet, write down an example of what is printed out and then explain the code that creates the output. Why is **`:suml`** divided by **`5`** in order to represent the Average Light Value? Explain.

STEP 7. Complete the challenges

Now that you have wired up and operated your two temperature sensors, please continue on to complete the challenges for this lesson.

3.4 Worksheet

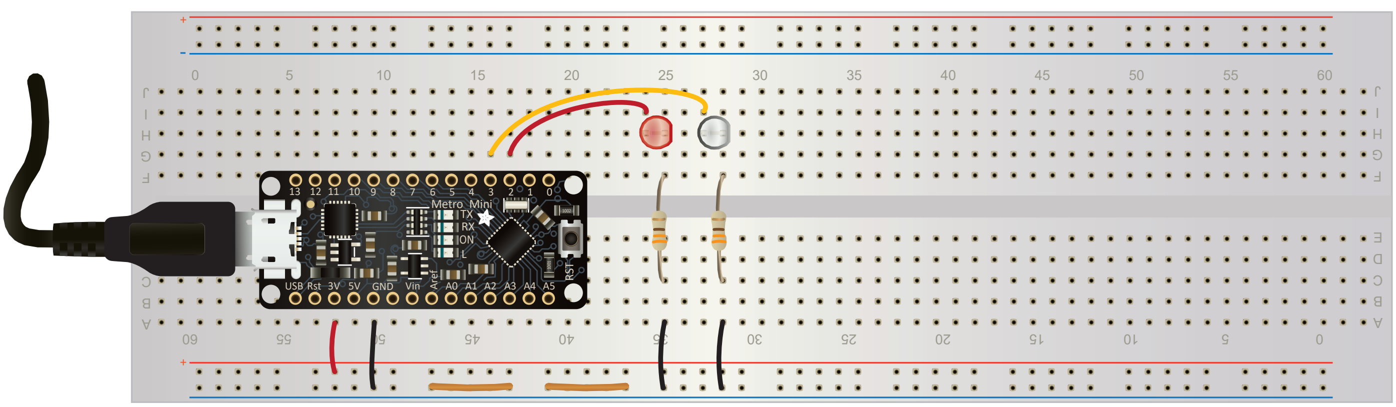
Sensing Temperature

Name:____

STEP 1 Write down each item on the materials list you have in front of you. If you notice you need to gather more materials, do that now and write them down.

STEP 2 Sketch the location and the wires for each sensor. Confirm your wiring with a classmate before proceeding to the next step.

Student's initials: ____________



Name:____

STEP 4 Describe the steps you took that enabled printing the temperature sensors outputs' on the screen when you `run-once`.

STEP 5 Write the added comments that explain what is happening on each line of code.

(BB-YourName.logo (~/BB-YourName) - Pluma

File Edit View Search Tools Documents Help

Open Save Undo

BB-YourName.Logo

```
let[suml 0]
let[sumt0 0]
let[sumt1 0]
  repeat 5 [
        let[light readLightSensor]
    let[suml :suml + :light]
  ]
print (se "|Average Light Value: | :suml / 5)

  ; repeat 5 [
;    let[tempsensor0 readTempSensor0]
;    let[sumt0 :sumt0 + :tempsensor0]
; ]
; print (se "|Average Temp0 Value: | :sumt0 / 5)
; repeat 5 [
;     let[tempsensor1 readTempSensor1]
;     let[sumt1 :sumt1 + :tempsensor1]
; ]
; print (se "|Average Temp1 Value: | :sumt1 / 5)

end
```

Name:____

STEP 6 Write down an example of what is printed out and then explain the code that creates the print outs. Why is `:sum1` divided by `5` before it is printed out? Explain.

Name:____

C-1

Use the digital multimeter to measure voltages that are output from each temperature sensor in the same way that you measured the output from the OPIC light sensor:

a. Use a short stripped red wire with your multimeter to measure the voltage going into the digital pin that is connected to each temperature sensor. Connect a short black wire to the ground rail.

b. Measure the voltage from each temperature sensor. Then type `run-once` in the black project window. Record the results in the table.

c. Repeat a. and b. at least two more times.

How do the values of voltages compare between the two temperature sensors?

How do the ADUs compare?

TempSensor0		TempSensor1	
Voltage	ADUs	Voltage	ADUs

C-2

Now change the temperature of one of the TMP-36 sensors and compare the voltage from the multimeter to the digital number output on the computer screen.

Fill in the table with the results of your experiment. Then repeat at least two more times with different temperatures.

Voltage	ADUs

Hint: one way to change the temperature is to warm it with your fingers.

Warning: You cannot get the sensors wet or they will malfunction.

Name:____

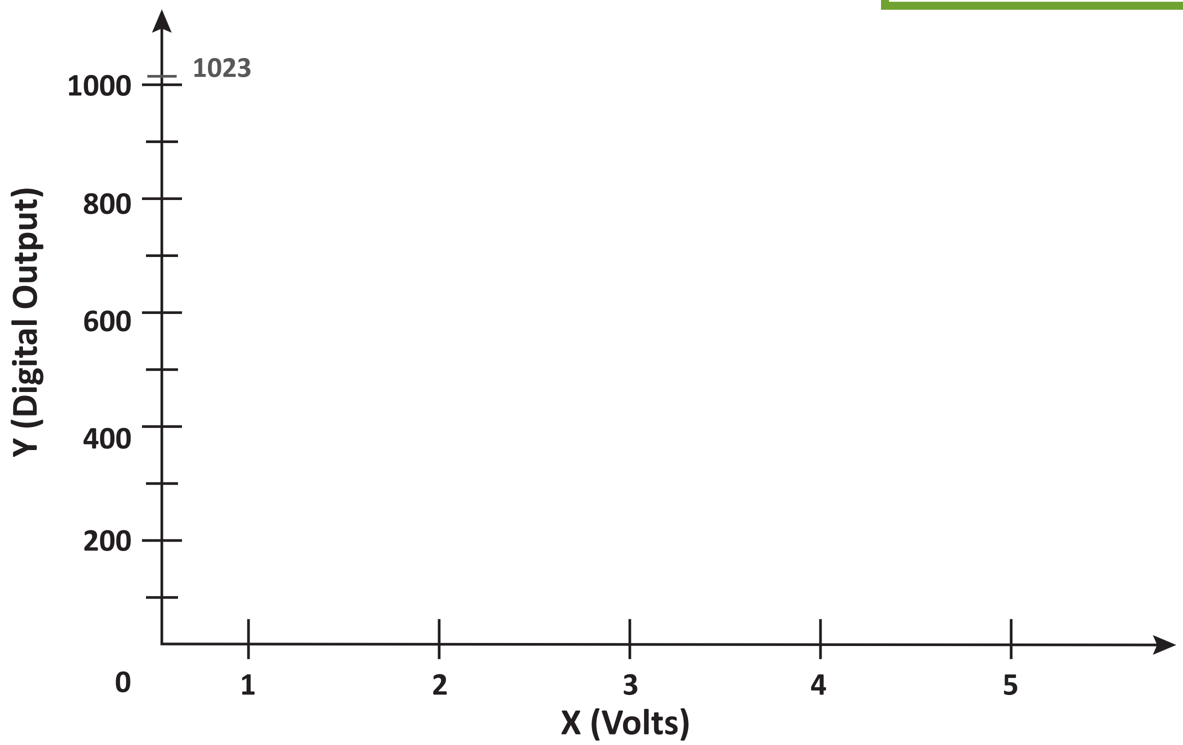
C-3

Use the table you created in the previous challenge to fill in the graph of voltage (x-axis) vs. digital output (y-axis). Is the relationship proportional or inversely proportional?

Circle your answer:

Proportional

Inversely proportional



Name:____

D-1

Scientists quantify the mathematical relationships between two quantities by fitting a straight line to the data. Here is some **Logo** code that will find the best fit line of the form $y = mx + b$ where m is the slope and b is the y-intercept (the value on the y-axis when the x-value is equal to 0).

```
to line-equation  :x1 :y1 :x2 :y2
     let [  slope ( :y2 - :y1 ) / ( :x2 - :x1 ) ]
     let [ yintercept :y1 - :slope * :x1 ]
     print (se "|y = | :slope "|x + | :yintercept )
end
```

Type this code into your `BB-YourName.logo` file, save, reload and try it out using (x,y) pairs of (Volts, ADUs) from your table in C-2.

D-2

Modify the code in D-1 so that the output looks like this:

```
line-equation 2 7 15 40

slope = 2.5384615384615383
y-intercept = 1.9230769230769234
```

Write down the final code and the output using a different set of numbers.

Name:____

D-3

Modify the `run-forever` word in `BB-YourName.logo` to read out the two temperature sensors (in addition to the light sensor). Add code to the print statement for each sensor so that it is clear which ADU value is coming from which sensor. Save, reload, connect and try out your new code.

Debug as needed. Write down the final code and an example output.

3.5 Making Lists of Sensor Values

Learning Objectives:

- Students will use **Logo** to read and write data in **lists**
- Students will write code demonstrating the ability to select elements in lists of data

Materials

For each group of 2 students

- Complete BasicBoard with Arduino, 2 LEDs, 1 OPIC light sensor and two TMP-36 temperature sensors
- Computer

For each student

- Worksheet
- Challenge sheet
- Ohm's Law Worksheet
- Double Dare Sheet

Vocabulary:

- Array
- Concatenate
- List

Tasks you need to perform

Answer questions in your Worksheet, Challenge & Double Dare sheets

Getting Started

In the previous lessons, you learned how to read the data from individual light and temperature sensors. In scientific experiments, you are usually measuring changes in sensor data that occur as time passes and conditions change. You therefore need to record the time at which the data are collected, as well as the data value at that time. One way of representing the data would be in a table such as shown below.

In a table like this, there are different ways to view and organize the data. When you read across one row, you see a list that includes:

Time of the readings *Sensor values at the time*

Time	Temp0 ADU	Temp1 ADU	Light ADU
1569267090	00154	00156	00286
1569267091	00156	00157	00289
1569267092	00155	00156	00284
1569267093	00155	00157	00286
1569267095	00156	00157	00288
1569267096	00155	00156	00285
1569267097	00155	00156	00286
1569267098	00155	00156	00287
1569267099	00155	00156	00285
1569267100	00155	00157	00287

When you view the data in any of the last three columns, you see a list of values from the same sensor, but at different times. For example, a list of all the light sensor readings, or a list of all the temperature sensor readings from `TempSensor0`. The first column is a list of all the times that readings were measured.

In this lesson you will learn how to make lists of sensor values.

Instructions

STEP 1. Set up your computer and BasicBoard

Turn on your computer and plug in the BasicBoard. Use the directions on page 3 to start a new project:

a. experiment: `BasicBoard.tar`

b. experiment folder name: `BB-YourName2`

c. version: `Packet-Cal`

After the black window reappears, `connect` the Arduino to the computer.

On your worksheet, write down the UNIX time after `Welcome to Logo.`

STEP 2. Learn how to make a list

You can also use a variable name to set up a list (or **array**) of numbers or strings or any combination. For example:

Contents of List

```
make "evens [ 0 2 4 6 8 ]
```

Name of the List

Beginning and end of the List

Test your code out to make sure it works before writing it down on your worksheet.

a. On your worksheet, predict what will happen when you type the instructions below into the terminal window:

```
make "evens [ 0 2 4 6 8 ]
print :evens
```

Then type it in and record the result on your worksheet.

b. Write down the code to make a list called **odds** that includes all the odd numbers from 1 through 9. Then type it in and record the result on your worksheet.

STEP 3. Learn how to select and view an element in a list

To see a particular element in the list, use the word **nth**.

Terminal

```
print nth 3 :evens ;prints element 3 in evens
6
```

Note that element 3 is really the fourth number in the list because the list index starts at zero. The **evens** list has 5 numbers in it, which are labeled elements 0 – 4.

To set the value of an element in a list, use the word **setnth**

Terminal

```
setnth 3 :evens 7 ;sets element 3 in evens to 7
```

a. On your worksheet, predict what will happen when you type the instructions below into the terminal window:

```
setnth 3 :evens 7
print :evens
```

Then type it in and record the result on your worksheet.

b. Write down the code that will change the number 7 In the **odds** list to the number 8. Type in the code and debug as necessary.

c. What output do you expect after the changes in step b. if you type `print :odds` ?

STEP 4. Create a list of temperature values from `TempSensor0`

You have previously used the **`Logo`** word **`se`** to print a combination of strings and numbers. In general, the **`Logo`** word **`se`** is used to **concatenate** (group together) types of information. In this step, you will use **`se`** to add values to an existing list. Type **`edit-project`** then find the word **`testTemp0list`** in your **`BB-YourName2.logo`** file.

Using the ten readings from **`TempSensor0`** in the table p.39, on your worksheet predict what you will see printed out for each iteration of the **`repeat`** loop in the **`Logo`** word **`testTemp0list`**.

H

The **`Logo`** word **`make`** is similar to **`let`**. Both assign values to variables, but the syntax is slightly different. We use **`make`** in the word **`testTemp0list`** so that the values will persist outside of the word in which they are defined.

STEP 5. Create a list that includes all three sensors' ADU values

Copy the code for **`testTemp0list`** into a new word that you will name **`testlist`**. Then modify the code so that instead of making a list with 10 values from **`TempSensor0`**, our new code will make one list that has one reading from each of the three sensors (in the order **`TempSensor0, TempSensor1, LightSensor`**). (A total of three values.)

Test and debug as needed. Write the final code and your results on your worksheet.

H

Remember the words **`readTempSensor1`** and **`readLightSensor`** will return the ADU values from these sensors.

STEP 6. Complete the Challenges

Now that you have learned how to use lists, please continue on to complete the challenges for this lesson.

Name:____

STEP 1 Write down the UNIX time after **`Welcome to Logo.`**

STEP 2 a. Predict what will happen when you type the instructions below into the terminal window:

```
make "evens [ 0 2 4 6 8 ]
print :evens
```

Then type it in and record the result below.

b. Write down the code to make a list called **`odds`** that includes all the odd numbers from 1 through 9. Then type it in and record the result below.

a.

b.

STEP 3 a. Predict what will happen when you type the instructions below into the terminal window:

```
setnth 3 :evens 7
print :evens
```

Then type it in and record the result below.

b. Write down the code that will change the number 7 In the **`odds`** list to the number 8. Type in the code and debug as necessary.

c. What output do you expect after the changes in step b. if you type `print :odds` ?

a.

b.

c.

Name:____

STEP 4 Using the ten readings from **`TempSensor0`** in the table p.39, predict what you will see printed out for each line of the **`repeat`** loop in the **`Logo`** word **`testTemp0list`**.

0:____

1:____

2:____

3:____

4:____

5:____

6:____

7:____

8:____

9:____

STEP 5 Test and debug as needed. Write the final code and your results below.

Name:____

C-1 Add a repeat loop to your code for `testlist`. Run the loop 5 times. Record the output.

C-2 Modify the code so that the number of times to repeat the loop is given by a variable that you will type in when you run the `testlist` word. After debugging and testing your code, write the code below.

C-3 There are three different sensor readouts. Modify the code so that the number “3” is printed out before each group of three readings. After debugging and testing your code, write the code below.

Name:____

D-1

Add code to the **`testlist`** word that will store all the values from a given sensor in its own list. Change the print statement so that at the end of the repeat loop, it will print out all the values for each sensor with labels indicating which sensor's values are being printed.

Your output should resemble this (numbers will differ):

Example:

Terminal

```
testlist 3
3 154 155 317
3 154 155 317 3 154 156 317
3 154 155 317 3 154 156 317 3 153 155 316
Temp0 values are 154 154 153
Temp1 values are 155 156 155
Light values are 317 317 316
```

*Your **`testlist`**:*

Terminal

D-2

Add code to the **`testlist`** word to average all the values from each sensor and print out the results. The output should look like that in D-1, but with the average values printed out afterwards, for example:

Example:

Terminal

```
testlist 3
3 154 155 317
3 154 155 317 3 154 156 317
3 154 155 317 3 154 156 317 3 153 155 316
Temp0 values are 154 154 153
Temp1 values are 155 156 155
Light values are 317 317 316
Average Value for Temp0: 154.8
Average Value for Temp1: 155.3
Average Value for Light: 316.7
```

*Your **`testlist`**:*

Terminal

*H: Look at the code for averageTempSensor0 that is in your **`BB-YourName2.logo`** file.*

3.6 Packets of Sensor Data

Learning Objectives:

- Students will devise an experiment to determine the locations in the packet for each sensor readout
- Students will learn how to calculate the approximate UNIX Time
- Students will learn how to calculate the checksum value for a given packet
- Students will extract sensor data from a packet

Materials

For each group of 2 students

- Complete BasicBoard with Arduino, 2 LEDs, 1 OPIC light sensor and two TMP-36 temperature sensors
- Computer

For each student

- Worksheet
- Challenge sheet
- Double Dare Sheet

Vocabulary:

- Checksum
- Packet
- UTC

Tasks you need to perform

Answer questions in your Worksheet, Challenge & Double Dare sheets

Getting Started

In the previous lessons, you learned how to create and read lists of data from individual light and temperature sensors. But the Arduino can send far more information to your computer than single sensor values.

The Arduino can send **packets** of data!

Have you talked on a cell phone today? Sent a text message? Watched a video? Read an email? Browsed the internet? If so:

You use data packets!

Nearly ALL of the digital information that you consume throughout the day is delivered in packets.

In this lesson you will learn how to analyze simple packets that represent the output of a scientific experiment that uses light and temperature sensors.

Instructions

STEP 1. What are Packets?

Discuss your initial ideas about packets with a classmate.

On your worksheet, write a few sentences that summarize your discussion.

Terminal

```
Starting BB-YourName2
T99 1569267090 25358 00001 00000 00000 00003 00154 00156 00286 39579
T99 1569267091 25358 00001 00000 01000 00003 00156 00157 00289 39573
T99 1569267092 25358 00001 00000 02000 00003 00155 00156 00284 39580
T99 1569267093 25358 00001 00000 03000 00003 00155 00157 00286 39577
T99 1569267095 25358 00001 00000 04000 00003 00156 00157 00288 39574
T99 1569267096 25358 00001 00000 05000 00003 00155 00156 00285 39579
T99 1569267097 25358 00001 00000 06000 00003 00155 00156 00286 39578
T99 1569267098 25358 00001 00000 07000 00003 00155 00156 00287 39577
T99 1569267099 25358 00001 00000 08000 00003 00155 00156 00285 39579
T99 1569267100 25358 00001 00000 09000 00003 00155 00157 00287 39576
```

STEP 2. Learning about Packets

The terminal window is showing 10 packets that include some of the information that you saw in the Getting Started table in the last lesson. But now the sensor readings and time values are in a packet structure that repeats for each round of sensor readings. Discuss your ideas about what each number represents with a classmate.

On your worksheet, summarize your ideas.

STEP 3. Set up your computer and BasicBoard

Turn on your computer and plug in the BasicBoard. Reload your **`BB-YourName2`** experiment and **`connect`** the Arduino to the computer.

On your worksheet, write down the UNIX time after **`Welcome to Logo`**.

STEP 4. Understanding UNIX Time

The time given in the packet is calculated as the number of seconds that have elapsed since 00:00:00 UTC on 1 January 1970. This is the definition of UNIX Time.

On your worksheet, calculate and write down how many seconds elapse in a day, year and in 50 years. Which number is closest to the UNIX time you wrote down when you started up your project *(STEP 3)*? Use the **`Logo`** word **`now`** to write down the current UNIX time.

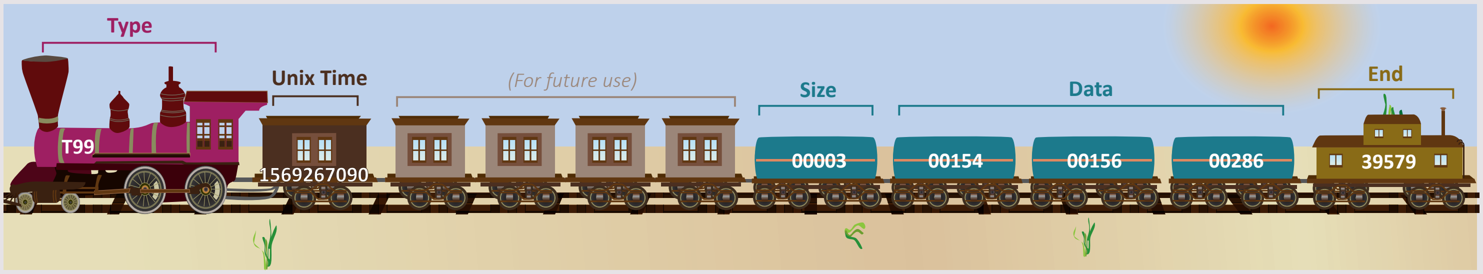
STEP 5. Understanding packets

We will use a train model to describe the structure of data packets used for our experiments.

The engine represents the packet **type**.
The first car represent the **time** it was created; The next four cars will not be explained at this time. The numbers in these entries, however, are used in calculating the checksum. See STEP 7 for more information about checksums.

The first blue car represents the **size** of the packet. This is the number of data points.
The remaining blue cars represent individual **data points** in ADUs.

The caboose represents the **checksum**.
The checksum finalizes the packet and allows us to verify that the packet is complete.



On your worksheet, fill in the missing numbers in the train model diagrams.

STEP 6. Decoding Sensor Information

The blue train cars represent the number of data points and the values of the data. Devise simple experiments to figure out which sensor readout goes with which data value. To see how the packets are changing, use **`run-once`**.

On your worksheet, write a few sentences to describe your experiments. Indicate your claims by filling in the blanks on the worksheet. Then summarize the evidence that supports your claims and explain your reasoning.

STEP 7. Understanding Checksums

When data are transferred, there is always a small chance that some of the bits of the data stream will be lost. In order to ensure that you have received all the bits, a special number is added to the end of a packet that has a specific mathematical relationship to the data in the packet. This special number is called a **checksum**.

Remember that the entry numbers start with 0. The 0^{th} entry is packet type, and the 1st entry is the large number which represents UNIX Time. Neither are used in the checksum calculation.

a) Ignore the packet type and Unix time

b) Begin with a value of 65537

c) Starting with 25358 (which is always the same number for Type 99 packets), subtract all the other packet entries

Repeat this process to calculate the checksum for the packet on your worksheet.

STEP 8. Complete the Challenges

Now that you have learned about packets, please continue on to complete the challenges for this lesson.

Name:____

STEP 1 Write a few sentences that summarize your discussion.

____

____

____

____

____

STEP 2 Summarize your ideas.

`T99 1569267090 25358 00001 00000 00000 00003 00154 00156 00286 39579`

____

____

____

____

____

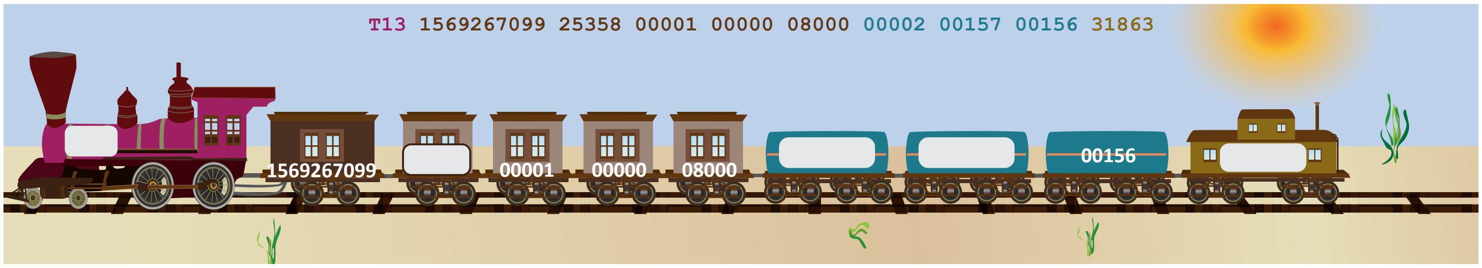
____

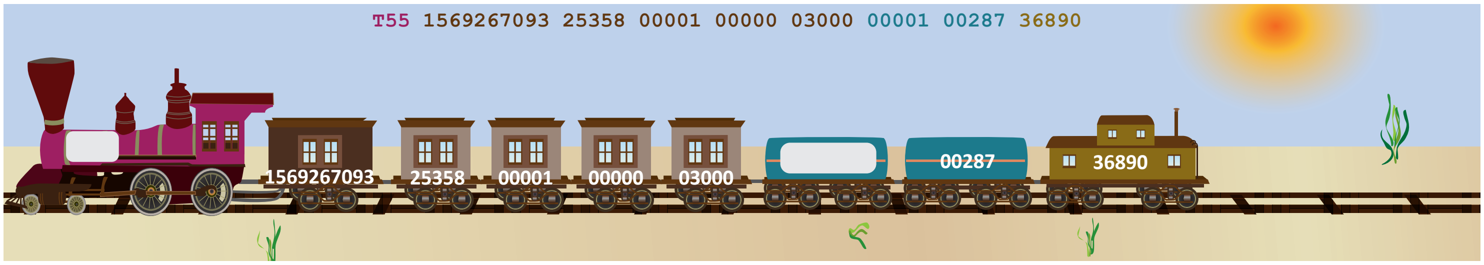
STEP 3 write down the UNIX time after **`Welcome to Logo`**: ____________________

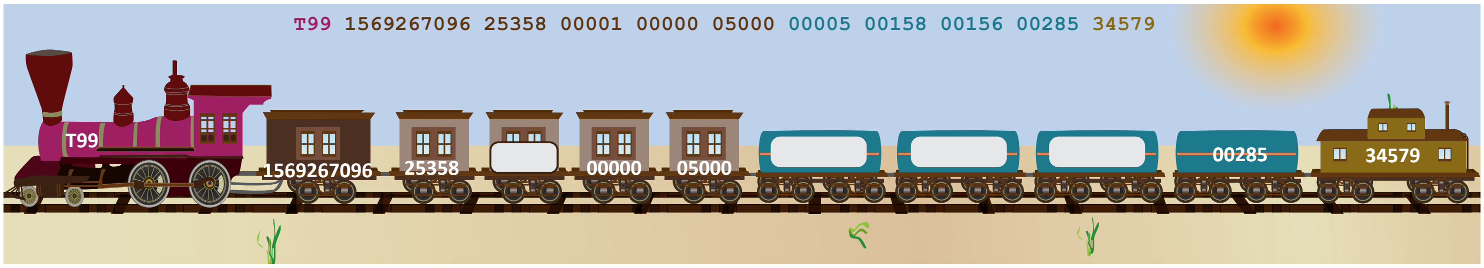
STEP 4 calculate and write down how many seconds elapse in a day, year and in 50 years. Which number is closest to the UNIX time you wrote down when you started up your project *(STEP 3)*? Use the **`Logo`** word **`now`** to write down the current UNIX time.

Name:____

STEP 5 Fill in the missing numbers.in the train model diagrams.







STEP 6 Write a few sentences to describe your experiments. Indicate your claims by filling in the blanks on the worksheet. Then summarize the evidence that supports your claims and explain your reasoning.

____

____

____

____

____

Claims: The first ADU value is from the __________________ Sensor.

The second ADU value is from the __________________ Sensor.

The third ADU value is from the __________________ Sensor.

(continue on the next page)

Name:____

STEP 6 ***continues***

Evidence: When I ___________________________, the first ADU value ___________________________.

When I ___________________________________, the second ADU value _________________________.

When I ___________________________________, the third ADU value ___________________________.

Reasoning: The first ADU value changed from ________________ to ____________ when I____________ _______________________________ because __________________________________.

The second ADU value changed from ________________ to ____________ when I __________________ because ________________________________.

The third ADU value changed from ________________ to ____________ when I _______________because ________________________________.

____

____

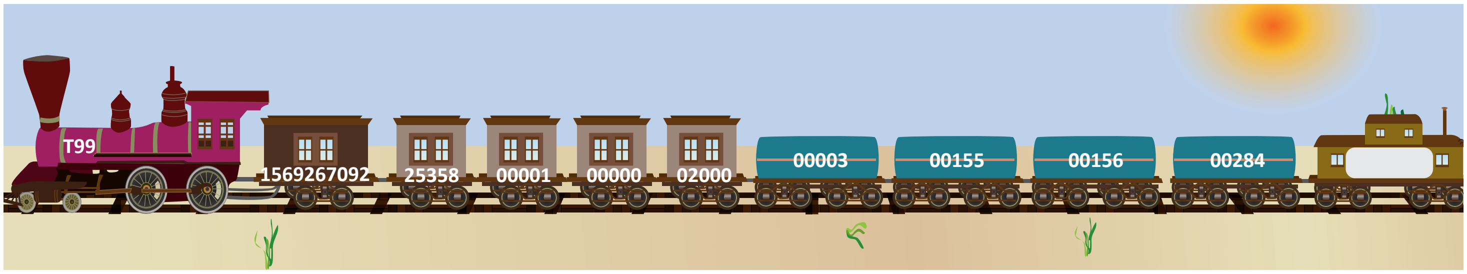
____

____

____

____

STEP 7 Calculate the checksum for the following packet:
T99 1569267092 25358 00001 00000 02000 00003 00155 00156 00284 - - - - -?



Name:____

C-1

Use the file system to locate a file named **`Evaporation_Sample.pac`**. In this file, one of the temperature sensors is showing readouts that indicate cooling. Open this file and analyze it to determine which temperature sensor is being used in the experiment. Explain your reasoning.

C-2

Now open the file called **`heat-diffusion_Sample.pac`**. How many sensors are being read out in this file? How do you know? Can you determine what type of sensors they are? Why or why not?

Name:____

D-1 Write code using `nth` to read out the `TempSensor0` values from type T99 packets.

D-2 Write code using `nth` to read out the `TempSensor1` values from type T99 packets.

3.7 TMP-36 Sensors Calibration

Learning Objectives:

- Students will graph the relationship between ADUs and temperature in degrees Celsius
- Students will plot temperatures in degrees Celsius given ADUs read out from the TMP-36 sensors
- Students will compare the calibration values for two different TMP-36 sensors

Materials

For each group of 2 students

- Complete BasicBoard with Arduino, 2 LEDs, 1 OPIC light sensor and two TMP-36 temperature sensors o
- Computer
- Thermometer
- 2 leashed TMP-36 sensors
- Containers for water and ice water

For each student

- Worksheet
- Challenge sheet
- Double Dare Sheet

Vocabulary:

- Calibration
- Intercept
- Slope

Tasks you need to perform

Answer questions in your Worksheet, Challenge & Double Dare sheets

Getting Started

In the previous lessons, we measured temperature using the TMP-36 sensors, but received results in ADU. It would be far more useful to convert those ADU values into temperatures in degrees Celsius. The process that enables this transformation is known as calibration. Being able to read out temperatures will be very important in future experiments.

Instructions

STEP 1. Accurate measurement

To perform the calibration process, we need to be sure that we have an accurate measurement of both temperature and ADUs at the same time. Consider the following situation:

A student places a temperature sensor and a thermometer in ice water and creates the following data table. Readings were taken at 10 second intervals.

Thermometer	TempSensor0
2˚C	180 ADU
0˚C	167 ADU
0˚C	164 ADU
0˚C	165 ADU
0˚C	164 ADU
0˚C	165 ADU

On her worksheet, the student wrote the following comments:

“I may not have waited long enough for the thermometer and sensor to settle down. The ADC readings bounce around a lot, but I don’t think there is a trend.”

She wants to pick one ADC reading to represent the freezing point of water (0˚ Celsius).

On your worksheet, write a few sentences explaining what ADU value you think she should use. Also describe how you decided on this number.

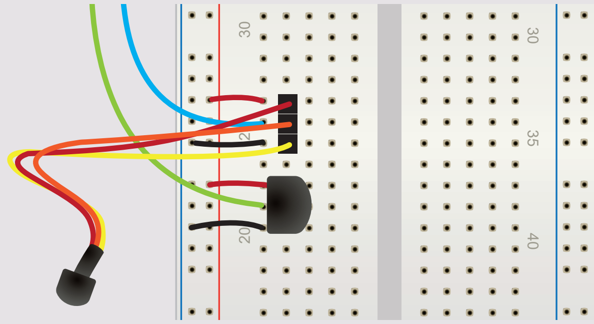
STEP 2. Gather materials

Gather the materials in the materials list.

On your worksheet, write down each item on the materials list you have in front of you. If you notice you need to gather more materials, do that now and write them down.

STEP 3. Replace each TMP-36 with a leashed version

The waterproofing and additional length make these more useful for future experiments. Note that on these sensors, the ground wire is yellow, the data wire is orange, and the power is (still) red. Ensure the data lines are still plugged in to A0 and A1.



Have your teacher verify that the temperature sensors are connected correctly. If so, ask your teacher to initial your worksheet.

> Take care when replacing the sensor with the leashed version. If it is plugged in backwards, it will overheat and fail.

STEP 4. Set up your computer and BasicBoard

Turn on your computer and plug in the BasicBoard. Reload your `BB-YourName2` experiment. And `connect` the Arduino to the computer.

Terminal

```
connect
```

On your worksheet, write down the UNIX time after `Welcome to Logo.`

STEP 5. Prepare the ice water and measure room temperature

Ask your teacher to give you ice and water. Put lots of ice into the water so that there will be enough ice to last for the entire class period without melting away.

On your worksheet, write down the temperature that is being measured by your thermometer which has been measuring room temperature.

STEP 6. Measure the room temperature with the sensors

Both the thermometer and the leashed temperature sensors should both be equilibrated to the room temperature. First, use the sensor that is attached to A0. This is called `averageTempSensor0`. Then use the sensor that is attached to A1, which is called `averageTempSensor1`.

Type `run-once, averageTempSensor0` and `averageTempSensor1` into your terminal window. On your worksheet, write the ADU values printed on the screen. Repeat this experiment two more times. Then average together the three numbers for each sensor to find the final values, and write these on your worksheet.

Terminal

```
run-once
averageTempSensor0
averageTempSensor1
```

STEP 7. Measure the ice water with both sensors and the thermometer

Place the thermometer into the ice water, along with both leashed temperature sensors.

Wait until the thermometer stabilizes near 0 degrees Celsius. (It may not reach 0 degrees but it should be close.) Write down the temperature from the thermometer. Then type `run-once`,`averageTempSensor0` and `averageTempSensor1` and on your worksheet, write the ADU values printed on the screen. Repeat this experiment two more times. Then average together the three numbers for each sensor to find the final values and write these on your worksheet.

Terminal

```
run-once
averageTempSensor0
averageTempSensor1
```

STEP 8. Graph the relationship between ADU and temperature

In order to understand the relationship between ADU and temperature, we can plot pairs of values on a graph.

On your worksheet, fill out the table and then plot the two points on the graph for each sensor.

STEP 9. Modify the code to do the calibration

In your terminal window, type edit-project to open the BB-YourName2.logo file. Locate the word `init-calibration`. Replace the empty brackets with the numbers from your table above in the following format:

```
[   cold ADU     cold temperature     warmer ADU    warmer temperature   ]
```

Pluma File Edit View Go Bookmarks Help Fri Jul 26, 14:55

BB-YourName.logo (~/BB-YourName2) - Pluma

Open Save Undo

BB-YourName2.Logo

```
to init-calibration
      make "temp0_cal []  ;calibration values for TempSensor0
      make "temp1_cal []  ;calibration values for TempSensor1
end
```

On your worksheet, write the two sets of numbers that you have typed into the file. Be sure to save the file, and then `reload` and `connect` your project.

> To see your calibrated values, `run-once` then `print :temp0-list` and `print :temp1-list`.

STEP 10. Complete the challenges

Now that you have calibrated your temperature sensors, please proceed to do the challenges.

Name:____

STEP 1 Write a few sentences explaining what ADU value you think she should use. Also describe how you decided on this number.

____

____

____

____

STEP 2 Write down each item on the materials list you have in front of you. If you notice you need to gather more materials, do that now and write them down.

STEP 3 Have your teacher verify that the temperature sensors are connected correctly. If so, ask your teacher to initial your worksheet.

Teacher's initials: ___________

STEP 4 Write down the UNIX time after `Welcome to Logo`: __________

STEP 5 Write down the temperature that is being measured by your thermometer which has been measuring room temperature: ____________

STEP 6 Type `run-once`, `averageTempSensor0` and `averageTempSensor1` into your terminal window. On your worksheet, write the ADU values printed on the screen. Repeat this experiment two more times. Then average together the three numbers for each sensor to find the final values, and write them down in the table below.

averageTempSensor0	Answer	averageTempSensor1	Answer
Measurement 1		Measurement 1	
Measurement 2		Measurement 2	
Measurement 3		Measurement 3	
Avg. at room temperature		Avg. at room temperature	

3.7 Worksheet *Continued*

TMP-36 Sensor Calibration

Name:____

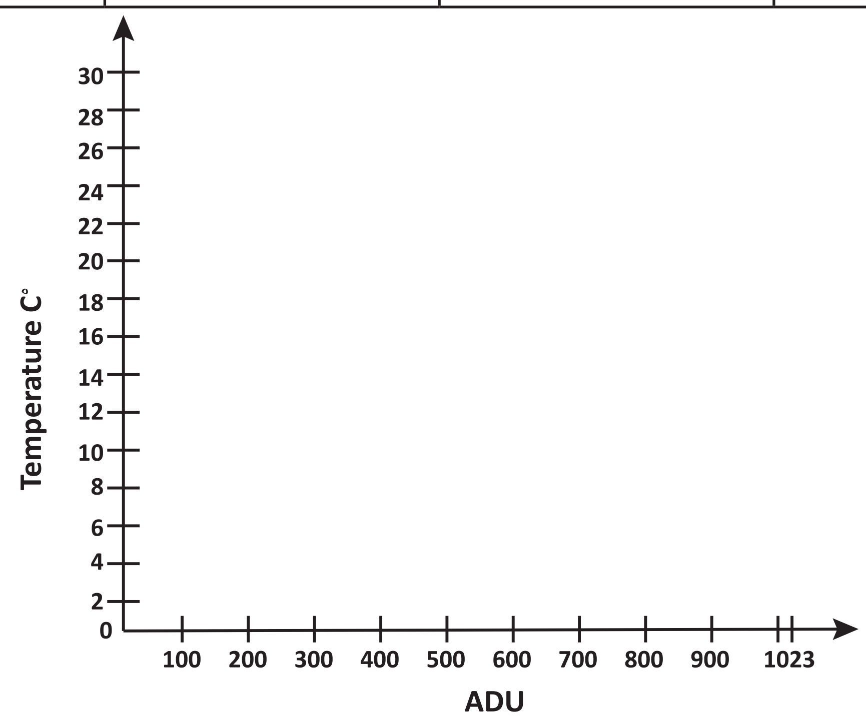
STEP 7 Wait until the thermometer stabilizes near 0 degrees Celsius. (It may not reach 0 degrees but it should be close.) Write down the temperature from the thermometer: _______________________

Then type `run-once`, `averageTempSensor0` and `averageTempSensor1` and write the ADU values printed on the screen. Repeat this experiment two more times. Then average together the three numbers for each sensor to find the final values and write them down in the table below.

averageTempSensor0	Answer	averageTempSensor1	Answer
Measurement 1		Measurement 1	
Measurement 2		Measurement 2	
Measurement 3		Measurement 3	
Avg. near 0° Celsius		Avg. near 0° Celsius	

STEP 8 Fill out the table and then plot the two points on the graph for each sensor.

Which container?	Temperature in °C	TempSensor0 Final ADU value	TempSensor1 Final ADU value
Room Temperature			
Ice water			



STEP 9 Write the two sets of numbers that you have typed into the file. Be sure to save the file, and then `reload` and `connect` your project.

3.7 Challenges

TMP-36 Sensor Calibration

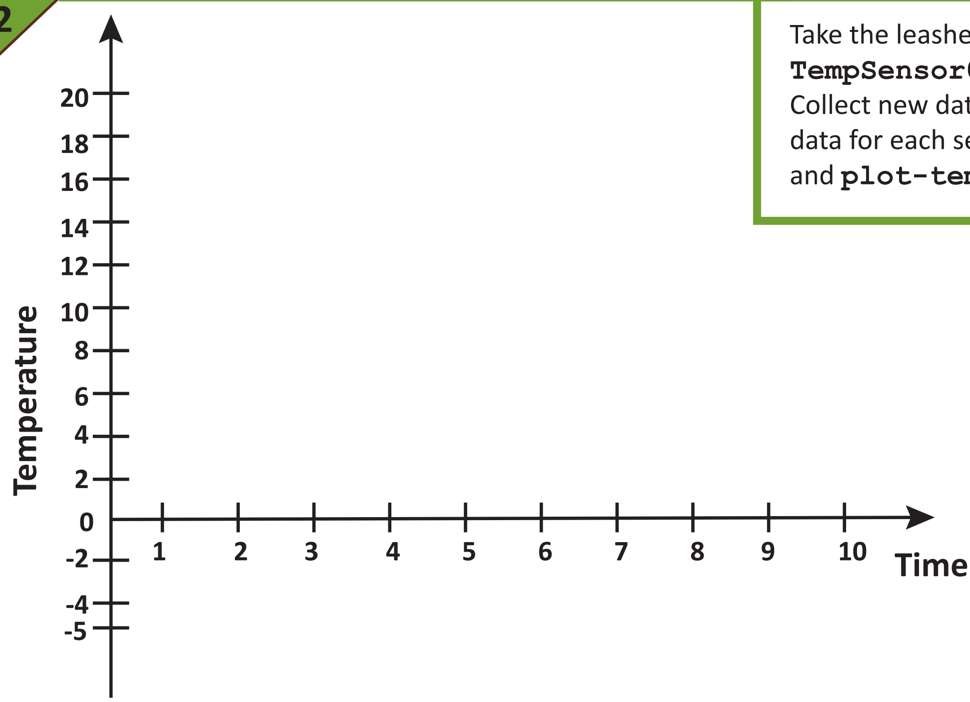
Name:____

C-1

The calibrated values for the temperatures are stored in lists called "**`temp0-list`** and "**`temp1-list`**. The times for both are stored in a list called "**`time-list`**. Print out the values in each of these lists, and write down the first three values in each list.

C-2

Take the leashed sensor attached to **`TempSensor0`** and put it back in the ice water. Collect new data by using **`run-once`**. Plot the data for each sensor by running **`plot-temp0`** and **`plot-temp1`**. Draw sketches of the plots.



C-3

Draw conclusions by looking at the plots from the two sensors. Do the temperature values shown on the graph make sense? If not, you might want to repeat the calibration process.

Name:____

D-1

Behind the scenes, **Logo** turns your two reference points into an equation of the following form:

Temperature = m × ADUs + b

You may recognize that this is the equation of a line in **slope-intercept** form. Determine the slope-intercept form equation for your temperature calibration. Use this equation to calculate temperatures for multiple ADU readings.

D-2

The previous equation solves for Temperature when you know the ADC reading. Determine the equation for predicting ADU readings when you know the temperature. In other words, solve the slope-intercept equation for ADUs rather than for temperature.