Unit 3 Sensors

Student Handouts

3.1 Sensing Light

Getting Started

The AppBoard has 34 pins. It'd be a shame to only use a handful of them for turning LEDs on and off.

In this unit, you will learn how to read signals from sensors. Sensors are the key to all future experiments. Sensors allow you to probe, measure, investigate, explore, and - ultimately – understand the world around you.

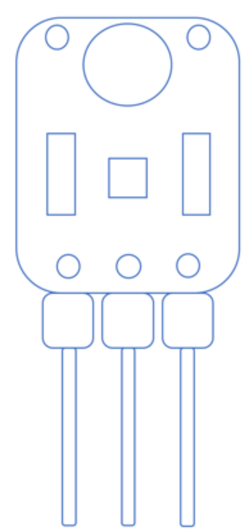
Let's get sensing!

Learning Goals

- ✓ Investigate the structure and function of the OPIC analog light sensor.
- ✓ Learn how the BasicBoard sends output signals from a sensor to the HP Stream.
- ✓ Use experimental observations to explain the structure and function of the OPIC analog light sensor.

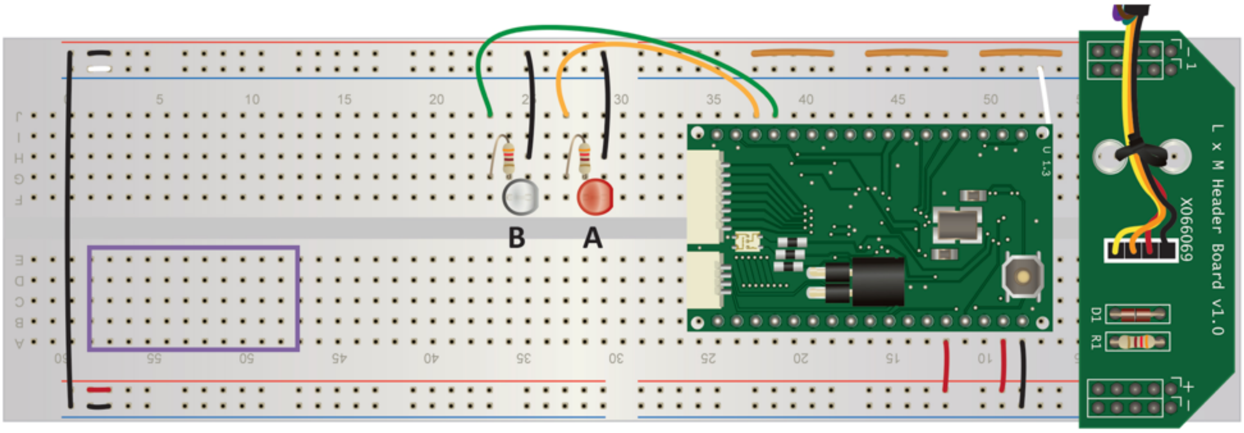
Instructions

1. Gather the following materials
   - ☐ BasicBoard
   - ☐ Wire as needed
   - ☐ Wire stripper
   - ☐ OPIC analog light sensor
2. Draw the following diagram in your notebook. This is the OPIC analog light sensor.



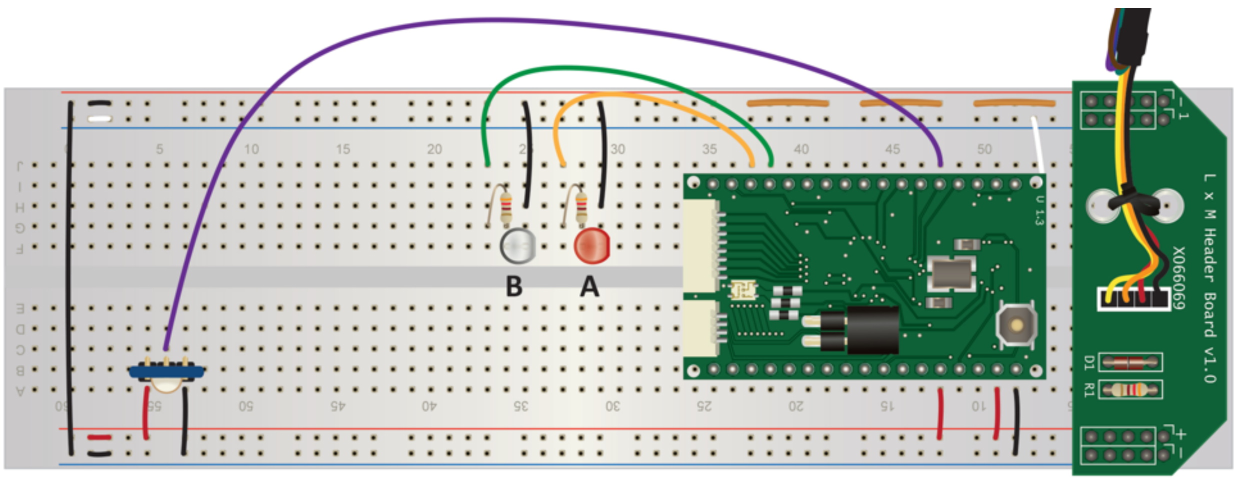
3. Make a prediction. What is the purpose, or function, of the structures on this sensor? Add labels to your diagram with your best guesses.
4. Remove the blue and green LED circuits from your BasicBoard. Remove the wires and resistor as well as the LED.

5. Plug the **light sensor** into the BasicBoard somewhere in the region shown below. The front of the sensor should face towards the power rail (power and ground).



6. Cut and strip a small section of black wire and a small section of red wire. Use this wire to connect VCC to **power** and GND to **ground**.

7. Cut and strip a long section of wire of any color. Use this wire to connect OUT to the pin labeled **ADC 6**.



8. Plug the FTDI cable into your HP Stream.

9. Create your own working copy of the experiment called **`NightLight`** by following these steps. If done correctly, a new black terminal window should pop up that says **`Welcome to Logo`**.

- Answer **`y`** to create a new experiment
- Select **`BasicBoard.tar`**
- Give your experiment a descriptive name. **Record this name in your notebook.**
- Select the version **`NightLight_Start`**

10. <u>Verify</u> that you have the new window that says **`Welcome to Logo`**. If it did not come up, ask the teacher to restart the full screen experiment startup process.

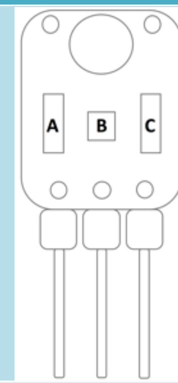
11. **Compile** and **download** the `NightLight` project. **Run** the program with the startup command, `.run-once`.

12. Confirm the program is running properly with your teacher before moving on. You should see a series of numbers print to the screen.

13. Complete the light sensor challenges.

Challenges

<table>
<tr><th>Credit</th><th>Task</th></tr>
<tr><td>◆</td><td>
How does this sensor detect light? Use the Logo output to investigate.



- What are these numbers printed on the screen?
- What part of the OPIC light sensor actually does the sensing? In your notebook, write 2-5 sentences to explain how to identify and isolate this component.
- Which of these three items is the light sensor: A, B, or C? Explain.
</td></tr>
<tr><td>◆◆</td><td>
How does this sensor react to light? Use the Logo output to investigate.

- Change the amount of light shining on the sensor.
- In your notebook, create a table to track light sources and readings.

<table>
<tr><th>Light Source</th><th>Logo Output</th></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>◆◆</td><td>
Describe the structure and function of the OPIC Analog Light Sensor.

- Redraw your sensor diagram with updated labels. What changed? What stayed the same?
- Examine your data table. Describe any patterns or relationships that you can infer from your observations and investigations.
- What can you conclude about how the light sensor communicates with the AppBoard and your computer? Write 2-5 sentences. Include direct evidence from your observations.
</td></tr>
</table>

Helpful Hints

If you need to start over, hold down the **`ctrl`** key and **`c`** at the same time. Next, type the command **`start`** and hit the **`enter`** key.
If you already created the Night Light experiment, answer **`y`** for **`Would you like to load an existing experiment?`**
If you see **`chip not found`**, call the teacher over.
If you see ______ **`undefined`**, you are trying to run a Logo word on the AppBoard that it doesn't understand.
If you see **`I don't know how to`** _______ , you are trying to run a Logo word on the HP Stream that it doesn't understand.
If you get an error message, see if you can figure out what you did wrong by asking a classmate for help. If all else fails, ask your teacher.
Watch the FTDI cable <u>during</u> download. If it blinks fast, the AppBoard is working.
Watch the FTDI cable <u>after</u> download. If it slowly blinks red and green, the AppBoard is working.

Going Further

Extra Credit	Task
◆	*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?
◆◆	How does the light sensor react to different colors? How does the light sensor react to different types of light (infrared, ultraviolet, etc.)? Create your own investigation and write a laboratory report.

3.2 Interpreting Light Sensor Readings

Getting Started

In addition to outputting on/off signals through the digital pins, we can read input voltages from sensors. 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 work by using discrete on/off signals as **inputs**. A sensor, on the other hand, may **output** a wide range of voltages.

What are examples of digital information?
What are examples of analog information?

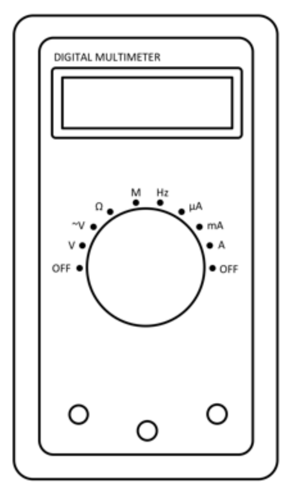
In this lesson, you will investigate how a digital device, the AppBoard, communicates with an analog device, the light sensor.

Learning Goals

- ✓ Use a digital multimeter as a tool to investigate sensor signals.
- ✓ Model the mathematical relationship between analog sensor readings and digital BasicBoard outputs.

Instructions

1. Gather the following materials
   - ☐ BasicBoard
   - ☐ Digital multimeter with probes and extension wires



2. Turn on your HP Stream and plug in the FTDI cable of the BasicBoard.
3. Reload your existing project with the name you chose in Lesson 3.1.
4. Use the digital multimeter to measure the voltage between **ground** and **power** on the BasicBoard. In your notebook, draw a diagram of the multimeter dial setting and probe connection locations.
5. Complete the light sensor challenges.

Challenges

Credit	Task

◆

How does the sensor communicate? Use the digital multimeter to investigate.

- Keep the black probe connected to ground.
- Connect the red probe to the middle pin of the sensor. Which breadboard holes are connected to this pin?
- Change the amount of light shining on the sensor.
- In your notebook, create a table to track light sources and readings.

Light Source	Voltage Reading	Logo Output

◆◆◆

Analog-to-Digital Conversion (ADC) pins convert continuous analog signals into discrete digital signals. The number of available **analog-to-digital unit (ADU) values** is constrained by the processing limits of the AppBoard microchip.

Hardware	Available ADUs	ADU Range
3-bit processor	8	0 to 7
4-bit processor	16	0 to 15
8-bit processor	256	0 to 255
12-bit processor	4096	0 to 4095

If we are using a 3-bit processor to convert voltage signals to ADUs, **`Logo`** can only output 8 possible values – the numbers 0 through 7.

Use this information to complete the **Analog-to-Digital Conversion worksheet**.

◆◆

Create an analog-to-digital conversion graph for the OPIC analog light sensor and AppBoard using your data table from the first challenge.

Graph Requirements

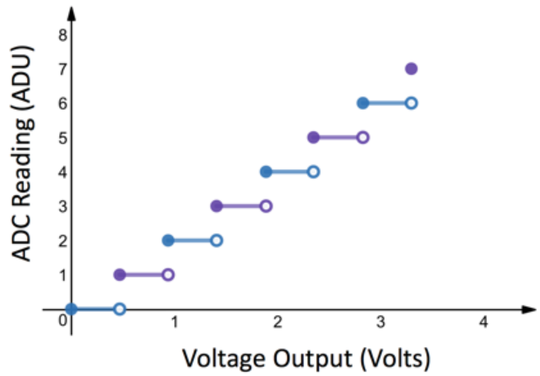
- Descriptive title
- x-axis label with units
- y-axis label with units
- Numbered axes
- Data points

Analog-to-Digital Conversion Worksheet

The following models describe how `Logo` and the AppBoard convert sensor signals to digital readings. Use these models to analyze two ADC/Sensor systems.

Graphical Representation

Example ADC with 8 values reads an analog sensor



1. **Logo** sets a rule for the extremes.

0.0 V corresponds to 0 ADU
3.3 V corresponds to 7 ADU

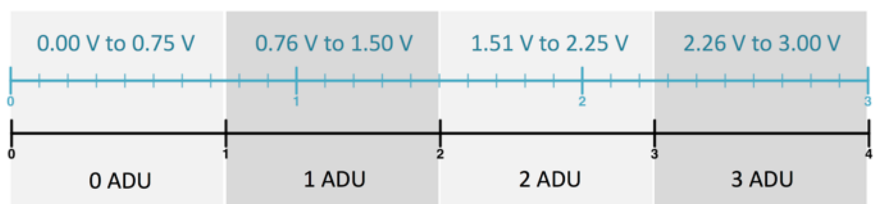
2. **Logo** assigns ranges of voltages to discrete ADU values.

Voltage Range	ADC Reading
0.0 V to 0.47 V	0 ADU
0.48 V to 0.94 V	1 ADU
0.95 V to 1.41 V	2 ADU

Number Line Representation

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

Voltage output
Units: volts



ADC reading
Units: ADU

Written Representation

Cause	ADC pin readings are always integers. Voltage outputs are real numbers.
Effect	A range of voltage outputs will correspond to a single ADU value.

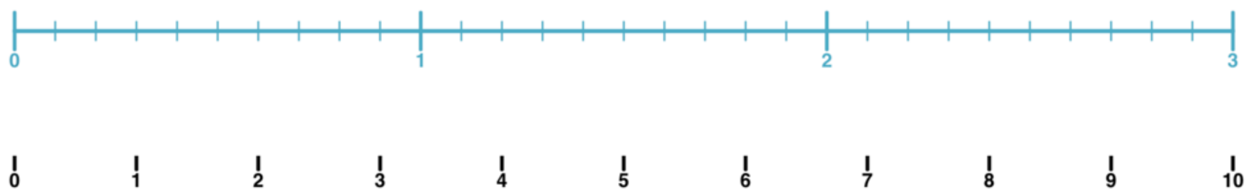
Case A

Sensor 0 V to 3 V
ADC 11 values

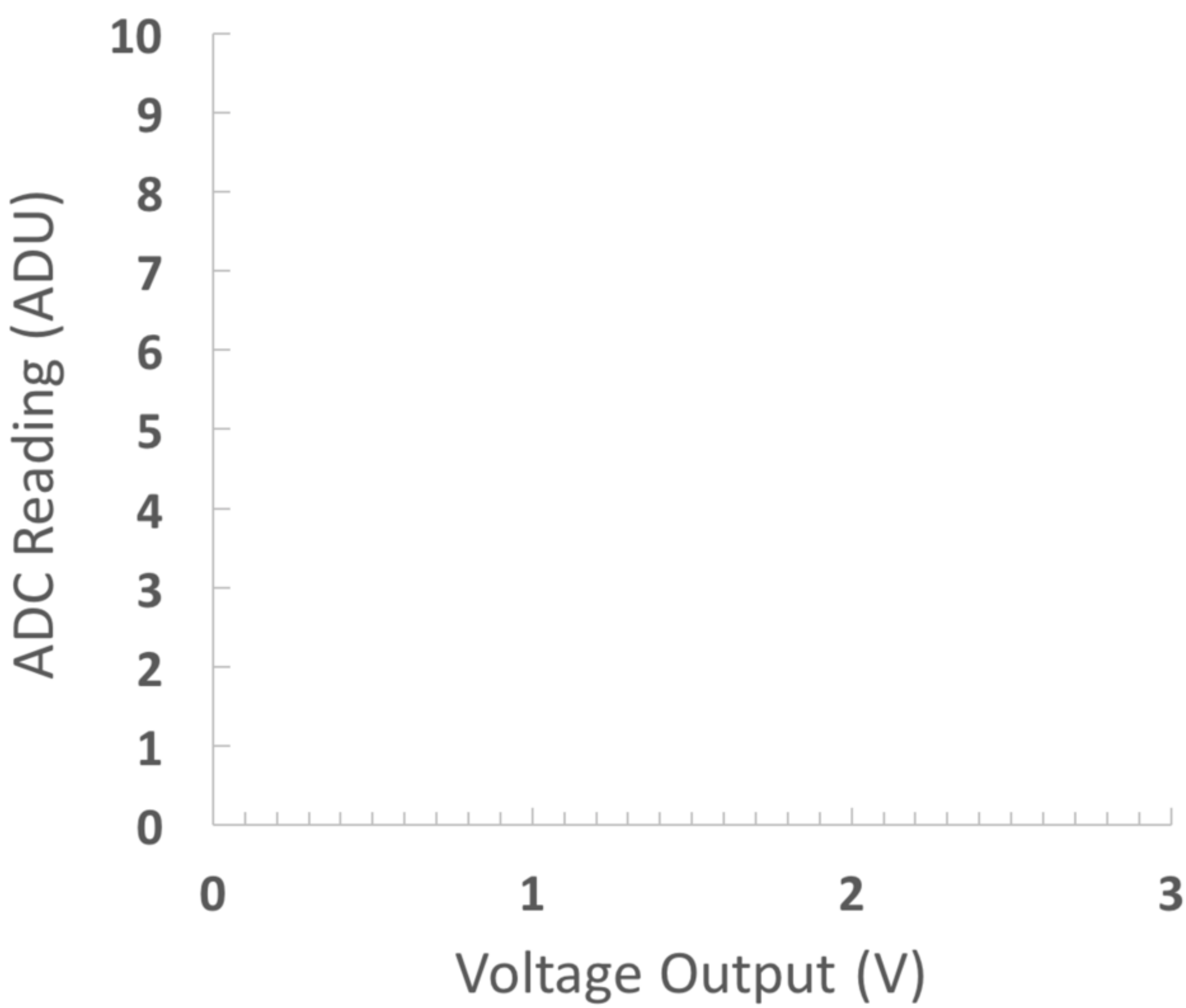
Sensor Signal	Logo Output
1.2 V	4 ADU
2.2 V	
2.3 V	
	8 ADU

1. Mark each reading on the corresponding number line.
2. Fill in the missing entries on the data table.
3. Add data points to the graph.

Voltage output
Units: volts



ADC readings
Units: ADU

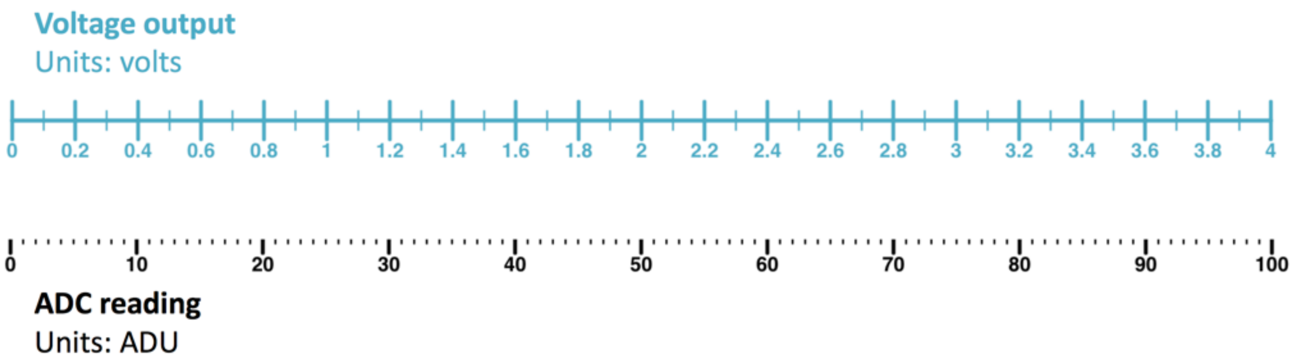


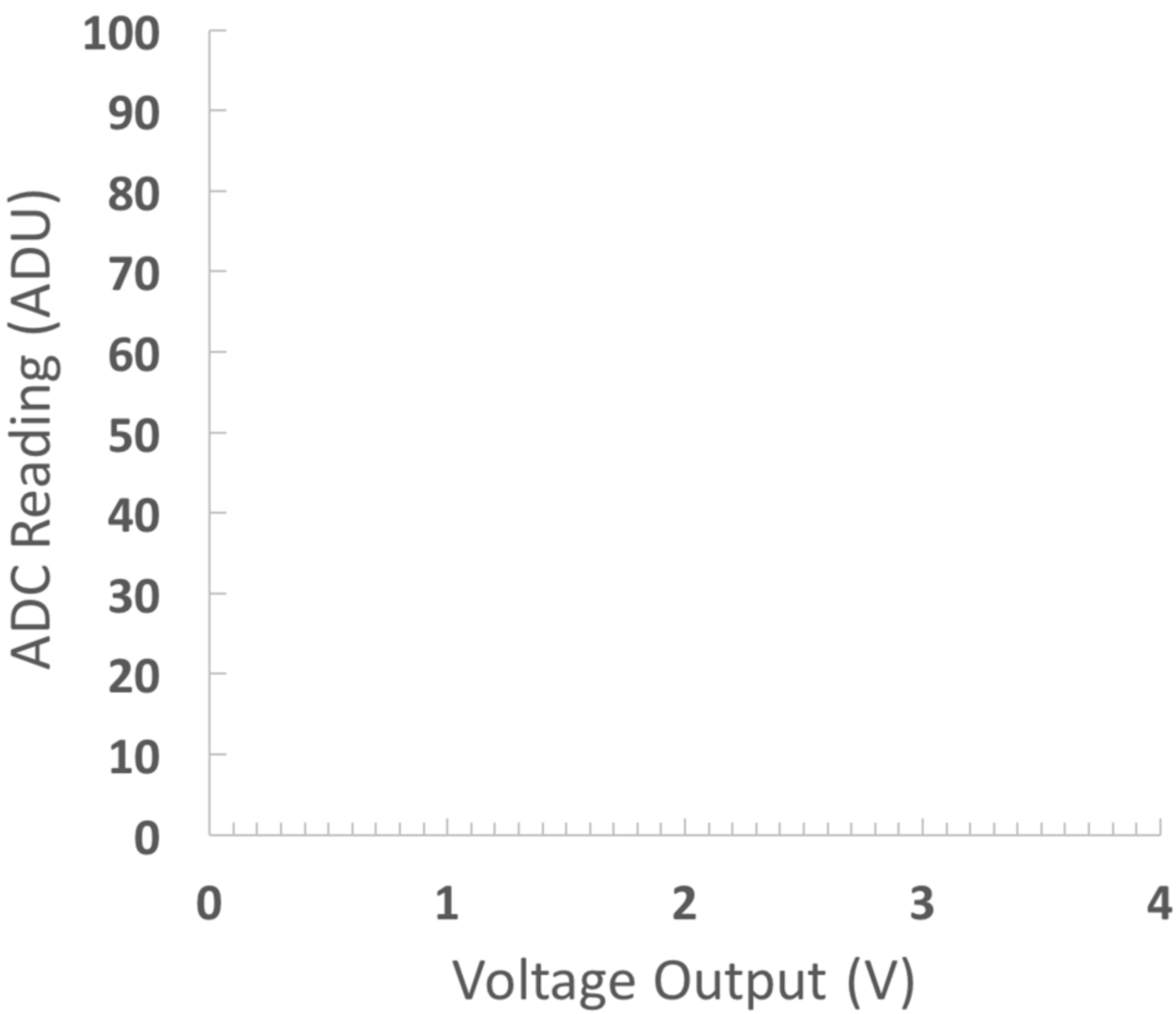
Case B

Sensor 0 V to 4 V
ADC 101 values

Sensor Signal	Logo Output
1.60 V	40 ADU
3.10 V	
	85 ADU
0.35 V	

1. Mark each reading on the corresponding number line.
2. Fill in the missing entries on the data table.
3. Add data points to the graph.





Helpful Hints

If you need to start over, hold down the **`ctrl`** key and **`c`** at the same time. Next, type the command **`start`** and hit the **`enter`** key.
If you already created the Night Light experiment, answer **`y`** for **`Would you like to load an existing experiment?`**
If you see **`chip not found`**, call the teacher over.
If you see ______ **`undefined`**, you are trying to run a Logo word on the AppBoard that it doesn't understand.
If you see **`I don't know how to`** _______ , you are trying to run a Logo word on the HP Stream that it doesn't understand.
If you get an error message, see if you can figure out what you did wrong by asking a classmate for help. If all else fails, ask your teacher.
Watch the FTDI cable <u>during</u> download. If it blinks fast, the AppBoard is working.
Watch the FTDI cable <u>after</u> download. If it slowly blinks red and green, the AppBoard is working.

Going Further

Extra Credit	Task
◆◆	The BasicBoard can only output 4096 possible values for ADC pin readings – the numbers 0 through 4095. The BasicBoard outputs signals ranging from 0 Volts to 3.3 Volts. This constraint limits the sensitivity of our sensors. 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.
◆◆◆	A flashlight shining just a few inches from your face is quite bright. The same flashlight held from the other side of the room appears dimmer. Collect and analyze data about this phenomenon. Plot ADU values versus distance on a graph. In 1-2 paragraphs, describe any patterns or relationships that you can infer from these data.

3.3 Build a Night Light

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**.

Learning Goals

- ✓ Learn how computational tools for making decisions are used in the Logo programming language.
- ✓ Use the comparison Logo words **<**, **>**, and **=** to compare numbers.
- ✓ Use the conditional Logo word, **`if`**, to make decisions.
- ✓ Test and refine a Logo program that turns light sensor input signals into output instructions for blinking an LED.

Instructions

1. Gather the following materials
   - ☐ BasicBoard
   - ☐ Digital multimeter with probes and extension wires
2. Turn on your HP Stream and plug in the FTDI cable of the BasicBoard.
3. Reload your existing project with the name you chose in Lesson 3.1.
4. Do not run the code. Instead, examine to full experiment code with the command:

```
.edit-project
```

5. The `edit-project` word will open several files in the Pluma text editor. Examine each file. Record the following table in your notebook. Include additional notes based on your examination of the files. (note: actual file names depend on your chosen experiment name)

Project File	**jLogo Files**	**uLogo Files**
`MyNightLight.prj`	`MyNightLight.logo`	`MyNightLight_Main.txt` `MyNightLight_Tools.txt`
Tells Logo which files need to be compiled and downloaded to the AppBoard	Code that runs on the HP Stream Logo words defined here can be run from the terminal but must begin with a dot (.)	Code that runs on the AppBoard Logo words defined here can be run from the terminal without a dot (.) `Main` – Experiment specific code that you will edit `Tools` – Common background code you should not edit

6. The file called `[your project name]_Main.txt` is the one you will work with today. close all other tabs in Pluma.

7. Locate the following piece of `Logo` code. In your notebook, explain what this code will do.

```
to ul-go
      greenon
      wait 10
      greenoff

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

8. Compile, download, and run this code to make sure it all works properly. If it doesn't, you may have accidentally changed the code during examination.

9. The following incomplete code will signal the board to turn on an LED if the light sensor reading drops too low.
Fill in the missing pieces. You may have to edit, compile, and download multiple times to get it to work. Use trial and error to set the ADU values for light and dark.

```
to ul-go
      greenon
      wait 10
      greenoff
            loop
            [
                  let [light readLightSensor]
                  if :light < _________ [ dp3on ]
                  if :light > _________ [ _____ ]
                  wait 10
            ]
```

10. What does this code do? How does the Logo word **`if`** work?

11. How does **<**, **>** work in this `uLogo` code? Enter each of the following lines in the terminal window. Use the Logo output as evidence to explain how `jLogo` and `uLogo` use the comparison words, <, >, and =. Are comparison words treated the same in both languages? Use your own numbers to explore further.

```
.print 1 > 30
.print 1 < 30
.print 1 = 30
.print 1 = 1
```

```
print 4 > 2
print 4 < 2
print 4 = 2
print 4 = 4
```

12. Complete the night light challenges.

Challenges

Credit	Task
◆	How reliable is the light sensor? • As a class, gather ADC readings when the classroom lights are turned on. • As a class, gather ADC readings when the classroom lights are turned off. • Reflect on these results. In your group, write down at least 5 reasons why sensor readings may differ.
◆◆	When does the light turn on? • Find a reasonable ADU value to turn on the night light when the classroom lights are turned off. • Find a reasonable ADU value that will turn on the night light ONLY when the sensor is completely covered. • Find a reasonable ADU value that will ONLY turn on the night light for a dim room, but not for a completely dark room.
◆◆	How fast does the night light react? • Your Logo program tells the AppBoard how often it should pause between reading the light sensor. Where is this done? • Change the rate at which your night light reacts to different light levels. Make it faster and make it slower.

Helpful Hints

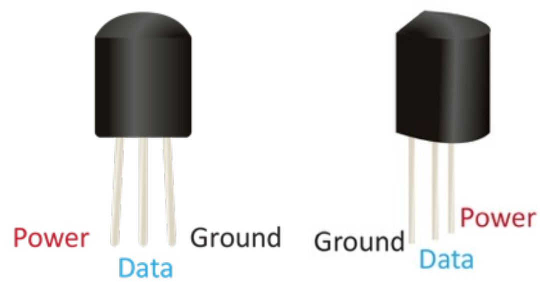
If you need to start over, hold down the **`ctrl`** key and **`c`** at the same time. Next, type the command **`start`** and hit the **`enter`** key.
If you already created the Night Light experiment, answer **`y`** for **`Would you like to load an existing experiment?`**
If you see **`chip not found`**, call the teacher over.
If you see ______ **`undefined`**, you are trying to run a Logo word on the AppBoard that it doesn't understand.
If you see **`I don't know how to`** _______ , you are trying to run a Logo word on the HP Stream that it doesn't understand.
If you get an error message, see if you can figure out what you did wrong by asking a classmate for help. If all else fails, ask your teacher.
Watch the FTDI cable <u>during</u> download. If it blinks fast, the AppBoard is working.
Watch the FTDI cable <u>after</u> download. If it slowly blinks red and green, the AppBoard is working.

Going Further

Extra Credit	Task
◆◆	Turn your night light into a communication device. Use the light sensor to input your message. Use the LED to display your message.
◆	Create an anti-night light. The LED turns on when the room is bright and turns off when the room is dark.
◆◆	How low can you get the ADU values? Construct something to block your sensor from the classroom lights.
◆◆◆	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 report or give a presentation.

3.4 Sensing Temperature

Getting Started



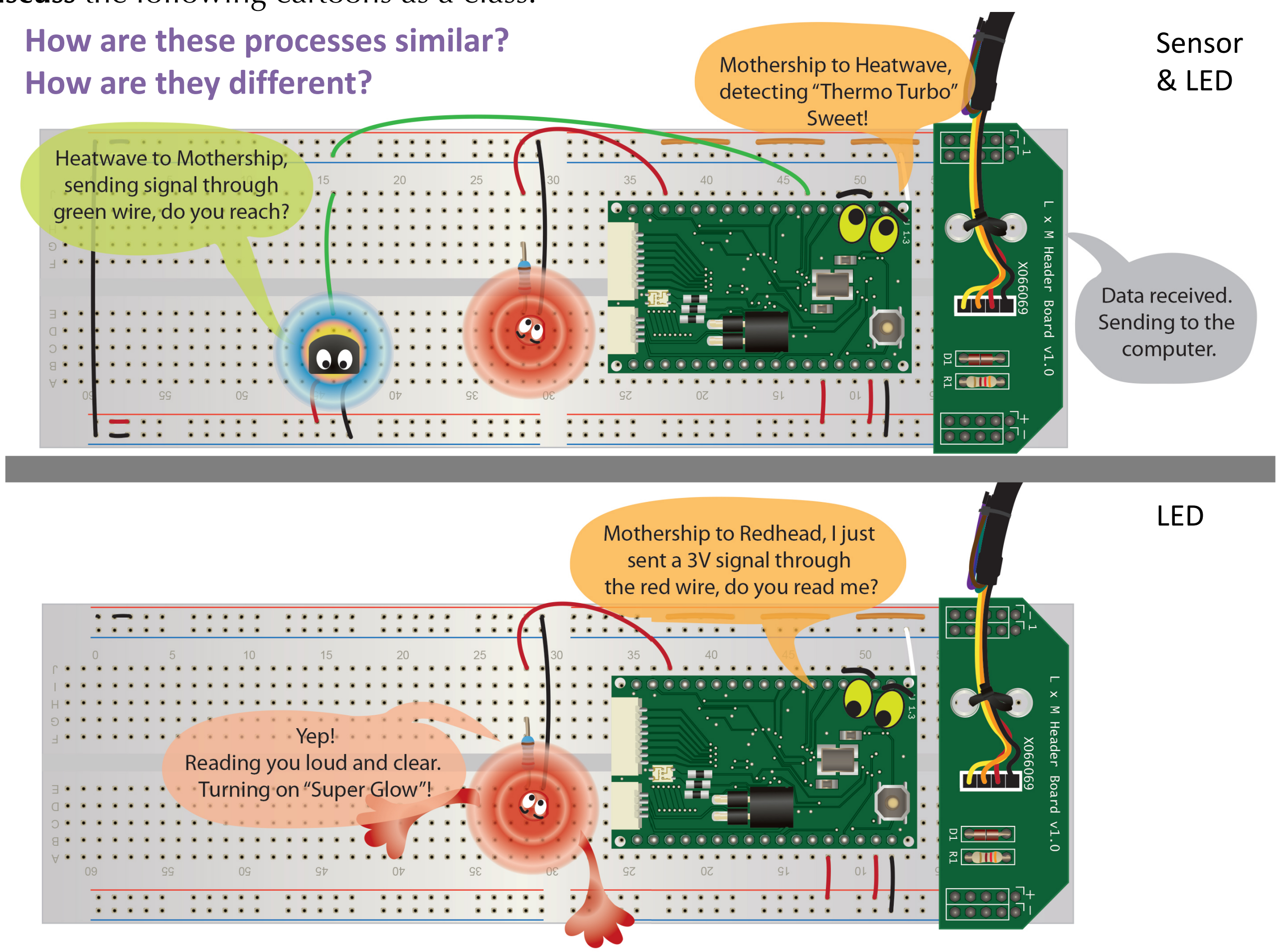
This is the **TMP 36 Analog Temperature Sensor**.

It connects to the AppBoard just like the OPIC Analog Light Sensor.

We will use this sensor to expand our understanding of electronic communications. Let's see just how powerful this `Logo` experiment system can be!

Discuss the following cartoons as a class:

How are these processes similar?
How are they different?



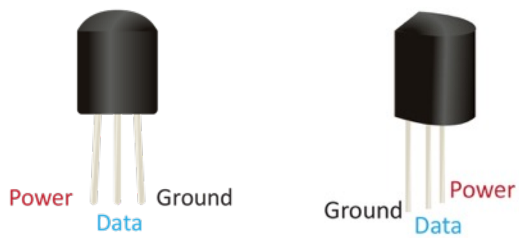
Learning Goals

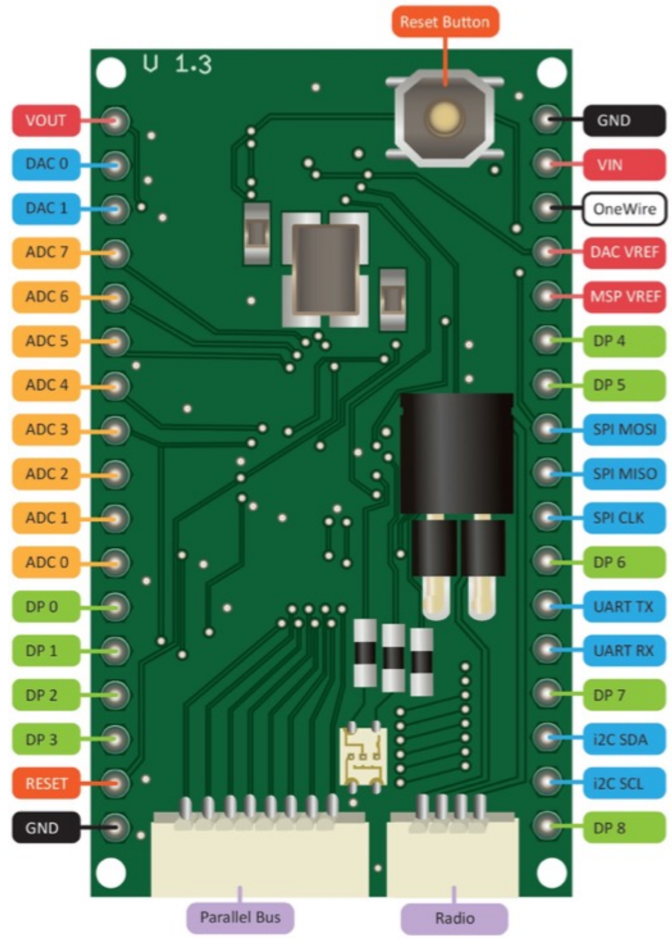
- ✓ Wire additional temperature sensors to the BasicBoard and write `Logo` code to read their signals.
- ✓ Gather evidence to assess the accuracy and precision of the temperature sensors
- ✓ Design an investigation to determine the relationship between sensor readings and temperature in degrees Celsius.

Instructions

1. Gather the following materials
   - ☐ BasicBoard
   - ☐ 2 TMP 36 Temperature Sensors
   - ☐ Wire stripper
   - ☐ Wire (as needed)
   - ☐ Thermometer

2. Use the following diagrams to wire the temperature sensors to **ADC 4** and **ADC 5**. Take care when connecting power and ground. If a sensor is plugged in backwards, it will overheat and fail.





3. Draw a diagram of this sensor and your new circuits in your notebook. <u>Confirm your wiring with your teacher before moving on</u>.

4. Turn on your HP Stream and plug in the FTDI cable of the BasicBoard.

5. Create a <u>new copy</u> of the **`NightLight`** experiment by following these steps. You are going to use this code as a template for working with temperature sensors.
   - Answer **`y`** to create a new experiment
   - Select **`BasicBoard.tar`**
   - Give your experiment the name **`MyTemperature`**
   - Select the version **`NightLight_Start`**

6. <u>Verify</u> that you have the new window that says **`Welcome to Logo`**. If it did not come up, ask the teacher to restart the full screen experiment startup process.

7. Use the command **`.edit-project`** to open and view the `Logo` program files. You will only use **`MyTemperature_Main.txt`**. Close all other tabs.

8. This program reads signals from the light sensor and prints them to the screen. Add two new variables for the temperature sensors and print these to the screen.

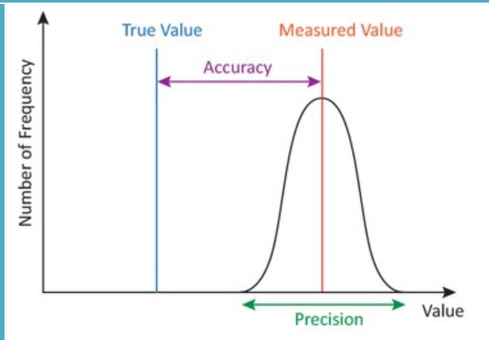
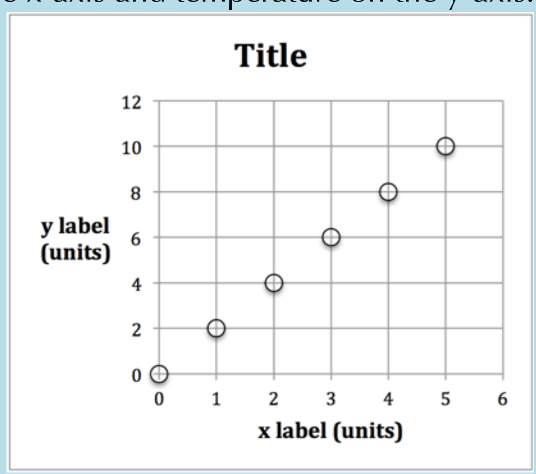
```
to ul-go
     greenon
     wait 10
     greenoff
     loop
          [
               let [light readLightSensor]
               let [temp1 readTempSensor1]
               let [temp2 readTempSensor2]
               print :light
               print :temp1
               print :temp2
               wait 10
          ]
end
```

9. Save **`MyTemperature_Main.txt`**

10. Compile, download, and run your program.

11. Complete the Temperature Sensor challenges.

Challenges

Credit	Task
◆	 If both sensors are **accurate**, they should output the **same signal** when measuring the **same temperatures**. If a sensor is not accurate, its signal may be offset from the true temperature. Each sensor could be offset by a different amount. In your notebook, answer the following questions: • Which numbers on the screen correspond to each sensor? How do you know? • Do the sensor readings agree? Gather evidence by recording sensor readings for at least 3 different temperature conditions. In your notebook, record how you created the conditions for each measurement.
◆◆	If each sensor is **precise**, it should output the **same signal** consistently for a **single temperature**. In your notebook, answer the following questions: • How will you set a temperature that does not change over time? • Is each sensor precise? Gather evidence by recording at least 10 readings for each sensor at a set temperature.
◆◆◆	• Design an investigation to determine the relationship between ADC outputs and temperature in degrees Celsius. • Create a graph with ADC values on the x-axis and temperature on the y-axis. **Your graph must have:** ☐ A descriptive title ☐ Axes labels with units ☐ Numbered axes ☐ Data 
◆◆	• Use your graph to **make predictions**. What ADC output do you expect for ice water at 0 degrees Celsius? What ADC output do you expect for boiling water at 100 degrees Celsius? What ADC output do you expect for room temperature at approximately 23 degrees Celsius? • Do your predictions match those of other groups? Why or why not?

Helpful Hints

If you need to start over, hold down the **`ctrl`** key and **`c`** at the same time. Next, type the command **`start`** and hit the **`enter`** key.
If you already created the experiment, answer **`y`** for **`Would you like to load an existing experiment?`**
If you edit any **`.logo`** files, use the word **`.reload`** before compiling and downloading.
To stop an experiment, type `..` (two dots) in the terminal window and hit the **reset** button on the AppBoard.
If you see **`chip not found`**, call the teacher over.
If you see ______ **`undefined`**, you are trying to run a Logo word on the AppBoard that it doesn't understand.
If you see **`I don't know how to`** _______ , you are trying to run a Logo word on the HP Stream that it doesn't understand.
If you get an error message, see if you can figure out what you did wrong by asking a classmate for help. If all else fails, ask your teacher.
Watch the FTDI cable <u>during</u> download. If it blinks fast, the AppBoard is working.
Watch the FTDI cable <u>after</u> download. If it slowly blinks red and green, the AppBoard is working.

Going Further

Extra Credit	Task
◆	Scientists often look for **proportional relationships** between two or more quantities. Explain what proportional relationships are and why they are important in science. Describe the proportional relationship between ADC readings and temperature in degrees Celsius.
◆◆	Three students are measuring the width of a window that they plan to cover with a poster. Each student writes down his or her result on a piece of paper. The window is actually 125 cm wide. Which of these measurements is most precise? Which of these measurements is most accurate? Which of these measurements is in agreement (overlaps)? 120 cm ± 5 cm 128 cm ± 10 cm 130 cm ± 10 mm

3.5 Sensors and Data Packets

Getting Started

In the **NightLight** experiment, you didn't see the *whole picture*. The AppBoard can send far more information to your computer than single ADC readings.

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

```
.run-once
  T13 1510257102 03338 65535 00008 07243  00001 01234  53713
  T13 1510257109 03338 65535 00009 03244  00001 01234  57711
  T13 1510257114 03338 65535 00009 09245  00001 01234  51710
  T13 1510257121 03338 65535 00010 05246  00001 01234  55708
  T13 1510257126 03338 65535 00011 01247  00001 01234  59706
```

Have you talked on the 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.

By encoding so much information into bite sized chunks, we can pass them around in any order, along any path, and at any rate. These chunks of information are easily decoded and reassembled by modern electronic devices.

Learning Goals

- ✓ Use clues from patterns within Logo program files and patterns displayed on the computer to explain the structure and function of data packets.
- ✓ Write `uLogo` code to assemble data packets with readings from all three sensors on the BasicBoard.

Instructions

1. Gather the following materials
   - ☐ BasicBoard

2. Turn on your HP Stream and plug in the FTDI cable of the BasicBoard.

3. Create your own working copy of the experiment called **`PacketDemo`** by following these steps.
   - Answer **`y`** to create a new experiment
   - Select **`BasicBoard.tar`**
   - Give your experiment a descriptive name. **Record this name in your notebook.**
   - Select the version **`PacketDemo`**

4. <u>Verify</u> that you have the new window that says **`Welcome to Logo`**. If it did not come up, ask the teacher to restart the full screen experiment startup process.

5. **Compile** and **download** the **`PacketDemo`** project.

6. Use **`.edit-project`** to view the Logo files. You will only need **`PacketDemo.logo`** and **`PacketDemo_Main.txt`** to complete this lesson.

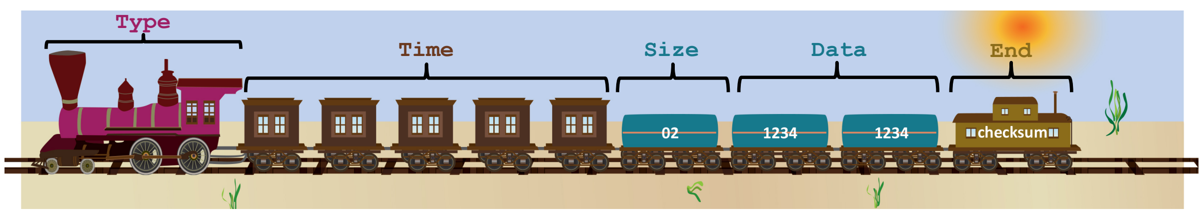
7. We will use a model to describe the structure of data packets in `Logo`. In your notebook, draw a diagram of the train analogy model and record the purpose of each car.

The engine represents the packet **type**.
The first five cars represent **time** it was created.

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.

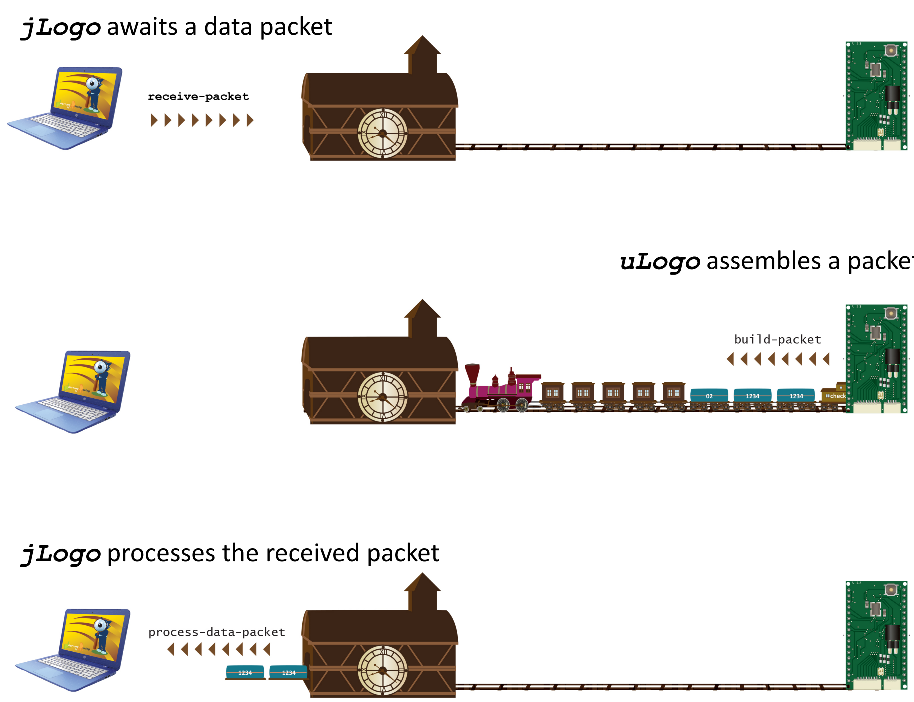
The caboose represents the **checksum**.
The checksum finalizes the packet and allows us to verify that the packet is complete.

```
T13 1510257126 03338 65535 00011 001247 0002 01234 01234 59706
```



8. The packets are assembled on the AppBoard using **`uLogo`** code found in **`PacketDemo_Main.txt`**. The computer follows instructions from the **`jLogo`** code found in **`PacketDemo.logo`** to await these packets and process them.

Locate the words **`receive-packet`**, **`build-packet`** and **`process-data-packet`** in both of these programs. In your notebook, describe where each is located and what each word appears to do.



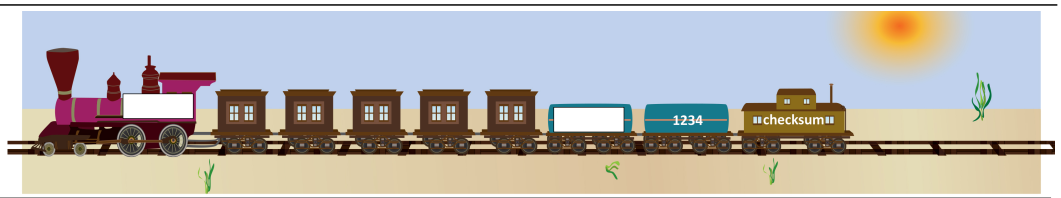
9. From the terminal window, run **`PacketDemo`** using **`.run-once`**. Allow the program to run for approximately 30 seconds. Pause the program by pressing the RESET button on the AppBoard. In your notebook, describe what happened when you ran the program.

10. Complete the Data Packet challenges.

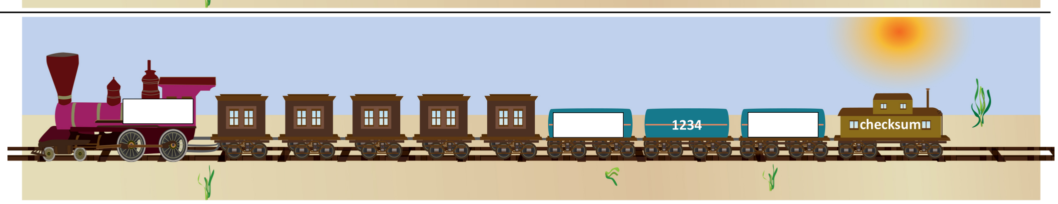
Challenges

Credit	Task
◆	Use clues from sample `uLogo` code to describe the structure of data packets in the image below. • Fill in the missing Packet Train labels based on the code provided. • Draw a train to represent a packet from your three sensors. Label each car and include actual data values.
◆◆	In `uLogo`, build a data packet that includes readings from all three sensors. • Locate build-packet in **`PacketDemo_Main.txt`** • Add new cars to the train with the command packet-word. • For example, to add a light sensor packet, you would add the line: **`packet-word readLightSensor`** • Save your file • Compile, download, and run your new program.

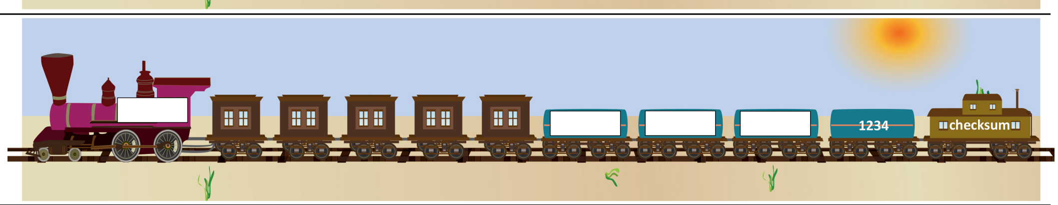
```
to build-packet
      init-packet 13
      packet-word 1234
      add-checksum
end
```




```
to build-packet
      init-packet 26
      packet-word 1234
      packet-word 7865
      add-checksum
end
```




```
to build-packet
      init-packet 42
      packet-word 4095
      packet-word 0001
      packet-word 1234
      add-checksum
end
```



Helpful Hints

If you need to start over, hold down the **`ctrl`** key and **`c`** at the same time. Next, type the command **`start`** and hit the **`enter`** key.
If you already created the experiment, answer **`y`** for **`Would you like to load an existing experiment?`**
If you edit any **`.logo`** files, use the word **`.reload`** before compiling and downloading.
To stop an experiment, type `..` (two dots) in the terminal window and hit the **reset** button on the AppBoard.
If you see **`chip not found`**, call the teacher over.
If you see _____ **`undefined`**, you are trying to run a Logo word on the AppBoard that it doesn't understand.
If you see **`I don't know how to`** _______ , you are trying to run a Logo word on the HP Stream that it doesn't understand.
If you get an error message, see if you can figure out what you did wrong by asking a classmate for help. If all else fails, ask your teacher.
Watch the FTDI cable <u>during</u> download. If it blinks fast, the AppBoard is working.
Watch the FTDI cable <u>after</u> download. If it slowly blinks red and green, the AppBoard is working.

Going Further

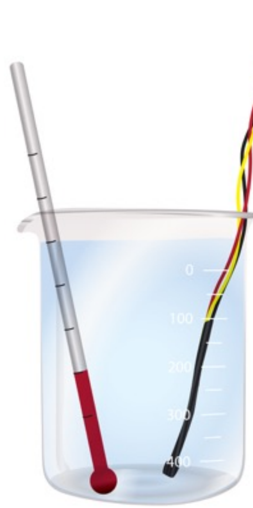
Extra Credit	Task
◆	Research how **data packets** are used to connect you to the internet over Wi-Fi. Create a flyer, poster, or presentation that explains this concept.
◆◆	The 2nd number in `Logo` packets is **Unix Time**. It is the number of seconds that have elapsed since 00:00:00 on January 1st, 1970. How many seconds are in an hour? How many seconds are in a week? How many seconds have elapsed since you were born? Create a table of important events in your life. Calculate or estimate the Unix Time for each event.
◆◆◆	**Checksums** are used to confirm the validity and integrity of data packets. They check to make sure nothing was lost during transfer. The device that assembles a packet performs a mathematical calculation using data within the packet. The answer to this calculation is the checksum. The receiving device reverses the calculation and verifies that it matches the data. Create your own mathematical function to generate checksums with simple datasets of your choosing.

3.6 Temperature Calibrations

Getting Started

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

Thermometer	Logo
2° C	70 ADU
0° C	65 ADU
0° C	43 ADU
0° C	58 ADU
0° C	46 ADU
0° C	52 ADU



In her notebook, 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). What value should she use? How did you decide on this number? Discuss.

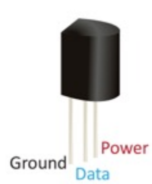
Learning Goals

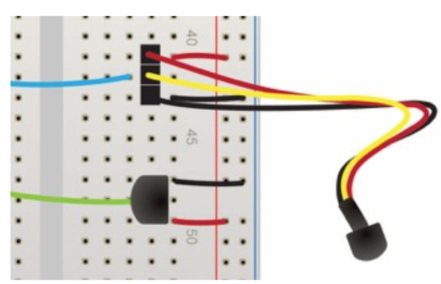
✓ Interpret graphical representations of temperature calibration.

✓ Learn how to use the `Logo` calibration program to automatically convert ADU readings to the physical quantity, temperature in degrees Celsius.

Instructions

1. Gather the following materials
   - ☐ BasicBoard
   - ☐ 2 leashed temperature sensors
   - ☐ 2 insulated cups
   - ☐ Access to hot and cold water





2. Replace each of your temperature sensors with leashed versions. The waterproofing and additional length make these more useful for future experiments.

3. Verify with your teacher that each leashed sensor is plugged in properly.

4. Turn on your HP Stream and plug in the FTDI cable of the BasicBoard.

5. Create a copy of the **`Calibration`** experiment by following these steps.
- Answer **`y`** to create a new experiment
- Select **`BasicBoard.tar`**
- Give your experiment the name **`MyCalibration`**
- Select the version **`Calibration_Start`**

6. Compile, download, and run the program. **You should not see anything print to the screen once you run it.**

7. In the terminal window, enter the command **`print sumTempSensor1`** and wait until a number shows up on the screen. Repeat this process with **`print sumTempSensor2`**. Each of these `uLogo` words adds together 10 individual sensor readings. Why does `sumTempSensor1` take longer to run than `readTempSensor1`? How much longer does it take to run?

8. The sensors are imperfect. Their precision is limited. If we average over 10 separate readings, we can get a better idea of where the true value lies. If the outputs of **`print sumTempSensor1`** and **`print sumTempSensor2`** are each 10 sensor readings added together, what are the average readings? In your notebook, explain how you calculated these averages.

9. Close down the `MyCalibration` project by closing the terminal window. Open a new terminal window and enter the command, **`start`**.

10. Create a copy of the full **`Calibration`** experiment by following these steps.
- Answer **`y`** to create a new experiment
- Select **`BasicBoard.tar`**
- Give your experiment the name **`FinalCalibration`**
- Select the version **`Calibration_Complete`**

11. Compile, download, and run the program.

12. Complete the Temperature Calibration challenges.

Challenges

Credit | **Task**

◆

Use the code within the `jLogo` file and the plots that pop up on the screen to answer these questions:

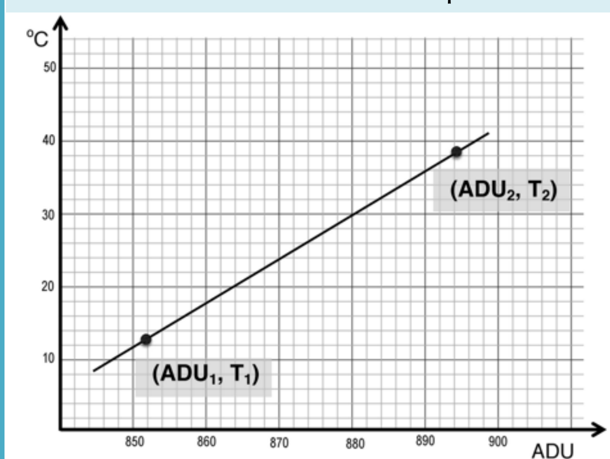
- When `Logo` passes packets from the AppBoard to the HP Stream, it sets a few basic **variables**. Recall, a variable name is set with a `"` symbol and the variable value is used with a `:` symbol. Look for patterns in the `jLogo` code. Look for patterns in the printed packets on the screen. How can you use these patterns to fill out the following table?

	Packet Word Location	Variable Name	Variable Value
Light Sensor			`:p42-word00`
Temperature Sensor #1		`"p42-word01`	
Temperature Sensor #2	4th Blue Number		

- Imagine that we changed the **packet type** to 53. What changes would need to be made to the `jLogo` code?

◆◆

The calibration program will convert all of your ADC readings to degrees Celsius, but first it needs two reference points.

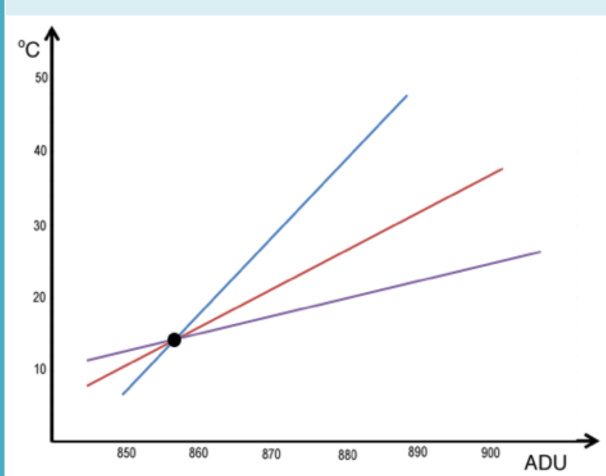


For each sensor, the program needs:

a **cold** pair of measurements (ADU1, Temperature1)

a **hot** pair of measurements (ADU2, Temperature2)

Every point along this graphed line is a valid pair of ADU and Temperature values.



Why wouldn't calibration work with one reference point? Use this diagram to construct your answer.

Determine the calibration reference points for each sensor.

1. Place a thermometer and both temperature sensors in cold water.

2. Wait a few minutes so everything can settle down to thermal equilibrium.

3. Use these `uLogo` words to collect 10 measurements from each sensor. Recall, enter `uLogo` words in the terminal window without a dot.

Logo Command	Logo Output	Thermometer Reading
`sumTempSensor1`		
`sumTempSensor2`		

4. Place a thermometer and both temperature sensors in hot water.

5. Wait a few minutes so everything can settle down to thermal equilibrium.

6. Use these `uLogo` words to collect 10 measurements from each sensor. Recall, enter `uLogo` words in the terminal window without a dot.

Logo Command	Logo Output	Thermometer Reading
`sumTempSensor1`		
`sumTempSensor2`		

7. Open **`FinalCalibration.logo`** and enter your calibration reference points in the word **`init-Calibration`**. The format is as follows:

   [cold ADU sum   cold temperature   hot ADU sum   hot temperature]

8. Save the `jLogo` file.

9. Use **`.reload`** to restart the project.

10. Compile and download the code to the AppBoard.

11. Use **`.run-once`** to run the program.

Helpful Hints

If you need to start over, hold down the **`ctrl`** key and **`c`** at the same time. Next, type the command **`start`** and hit the **`enter`** key.
If you already created the experiment, answer **`y`** for **`Would you like to load an existing experiment?`**
If you edit any **`.logo`** files, use the word **`.reload`** before compiling and downloading.
To stop an experiment, type **`..`** (two dots) in the terminal window and hit the **reset** button on the AppBoard.
If you see **`chip not found`**, call the teacher over.
If you see **`_____ undefined`**, you are trying to run a Logo word on the AppBoard that it doesn't understand.
If you see **`I don't know how to _______`**, you are trying to run a Logo word on the HP Stream that it doesn't understand.
If you get an error message, see if you can figure out what you did wrong by asking a classmate for help. If all else fails, ask your teacher.
Watch the FTDI cable <u>during</u> download. If it blinks fast, the AppBoard is working.
Watch the FTDI cable <u>after</u> download. If it slowly blinks red and green, the AppBoard is working.

Going Further

Extra Credit	Task
◆	Create calibration labels for all of the temperature sensors in the classroom. Use tape or strips of paper to record the calibration reference points and attach them to each sensor.
◆◆	Behind the scenes, Logo turns your two reference points into an equation of the following form: $$\text{Temperature} = m \times \text{ADUs} + b$$ 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.
◆◆	The previous equation solves for Temperature when you know the ADC reading. Determine the equation for predicting ADC readings when you know the temperature. In other words, solve the slope-intercept equation for ADUs rather than for temperature.

3.7 Understanding Temperature

Getting Started

What is the temperature of your school?

On the surface, this seems like a simple enough question. It asks for one number that is quite familiar - temperature. Something is troubling though...

How do you assign one overall value to an entire campus?
Is every room and hallways the same temperature?
Should we measure during the day or at night?
What are the ranges of temperatures throughout the year?
Is there a single number that represents a majority of the school campus?

These are questions that can be approached using **data analysis**. Analyze data by creating and interpreting graphs and by applying statistical tools.

Statistics is the science of learning from data and the science of measuring, controlling, and communicating uncertainty. We will use `jLogo` to determine four statistical quantities and to generate graphs:

Average or **Mean**– numerical average of all quantities in a set
Median – middle value of all sorted quantities
Mode – most frequent quantity in a set
Standard Deviation – characterization of the spread in data

Learning Goals

- ✓ Identify patterns of evidence by organizing, representing, and analyzing data in the `Logo` programming environment.
- ✓ Interpret graphical representations of measured quantities.

Instructions

1. Gather the following materials
   - ☐ BasicBoard
   - ☐ Graph paper (if needed)
2. Turn on your HP Stream and plug in the FTDI cable of the BasicBoard.
3. Reload your temperature calibration project, **`FinalCalibration`**.
4. Use **`.edit-project`** to open the `Logo` files for your project. You will only need the `jLogo` file, **`FinalCalibration.logo`**, to complete this lesson.
5. Complete the Data Analysis challenges.

Challenges

Credit	Task
◆◆	The utility company in San Francisco wants to use the January 2017 temperature history to predict how much gas and electricity it must produce for residents in the future.

- Begin writing a `jLogo` program to predict a typical temperature in January.

☐ At the bottom of **`FinalCalibration.logo`**, create a new word called **`januaryTemperature`**.

Within this word, you will enter these historical temperatures and run a `Logo` analysis word. <u>Type a space between each number.</u>

```
to januaryTemperature
  y-data "temperature [11 11 11 14 …]
  analyze
end
```

☐ Save this file and return to the terminal window.

☐ Enter the command **`.reload`** to restart the project.

☐ Enter your new word in the terminal window to run it.
`.januaryTemperature`

☐ If you get any error messages or something looks wrong with the `Logo` output, return to your program and look for typos or errors.

☐ In your notebook, record the **average (mean)**, **median**, and **standard deviation**.

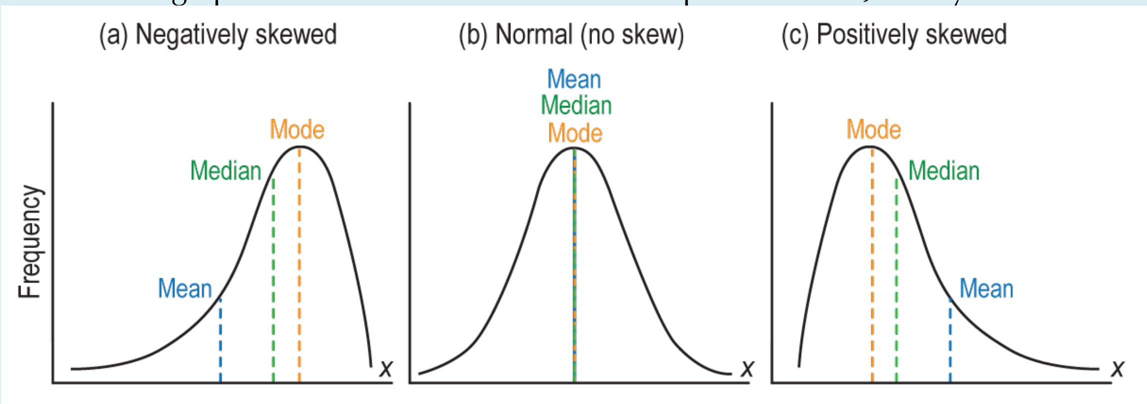
☐ Look at the list printed to the terminal screen for `mode`. Which temperature appears the most number of times? This is the **mode**. Record it in your notebook.

Date	T (°C)
1/1	11
1/2	11
1/3	11
1/4	14
1/5	10
1/6	10
1/7	16
1/8	17
1/9	15
1/10	14
1/11	13
1/12	11
1/13	14
1/14	15
1/15	11
1/16	12
1/17	11
1/18	11
1/19	14
1/20	13
1/21	13
1/22	14
1/23	12
1/24	12
1/25	12
1/26	13
1/27	15
1/28	16
1/29	17
1/30	16
1/31	15

◆

If you gather together a large number of measurements and tally how often certain measurements appear, you can create a **distribution graph**.

- Based on your numbers for **average (mean), median**, and **mode** – which distribution graph best matches the historical temperatures for January 2017?



- Which of these three numbers do you believe should be quoted as "A typical temperature in January"? Explain your choice.

◆◆

There may have been some unusually high or unusually low temperatures in January 2017. We can create a graph to look for outliers and to get a better understanding of the data distribution.

- Continue adding to the `Logo` program for examining temperatures.

☐ Within your word, **`januaryTemperatures`**, add a list to represent the day of the month and two more Logo words that will create and display a graph. Again, enter a space between each number in the list.

```
to januaryTemperature
  y-data "temperature [11 11 11 14 …]
  analyze
  x-data "day [1 2 3 4 …]
  quick-plot
  display-quick
end
```

☐ Save this file and enter **`.reload`** in the terminal window to restart the project.

☐ Enter your new word to run it.

`.januaryTemperature`

☐ If you get any error messages or something looks wrong with the `Logo` output, return to your program and look for typos or errors.

☐ In your notebook, describe any patterns you see in the graph.

◆◆	The **standard deviation** quantifies the amount of spread in a data distribution. The majority of measurements should fall within (average – standard deviation) and (average + standard deviation). Scientists typically use standard deviation to represent the uncertainty. • Examine your graph and your values for average and standard deviation. Do these quantities accurately describe the temperature history of January 2017? • If you were writing a report for the utility company in San Francisco, what would you predict for a typical temperature in January? How much uncertainty would you include? Explain how you determined these two numbers.
◆◆◆	Return to your `Logo` program, **`FinalCalibration.logo`**. This program already generates graphs for temperatures from each sensor. Expand on this project to answer the following question: • What is a typical temperature in your classroom? ☐ Search for these plotting `Logo` words. Add the **`analyze`** word at the bottom of each (after the line that starts with `display-plot` but before `end`). ☐ Save, reload, compile, and download this project. ☐ Run the program for at least 5 minutes. ☐ Gather evidence from the terminal window and graphs to answer the question. Explain how you determined your answer.

Helpful Hints

If you need to start over, hold down the **`ctrl`** key and **`c`** at the same time. Next, type the command **`start`** and hit the **`enter`** key.
If you already created the experiment, answer **`y`** for **`Would you like to load an existing experiment?`**
If you edit any **`.logo`** files, use the word **`.reload`** before compiling and downloading.
To stop an experiment, type `..` (two dots) in the terminal window and hit the **reset** button on the AppBoard.
If you see **`chip not found`**, call the teacher over.
If you see _____ **`undefined`**, you are trying to run a Logo word on the AppBoard that it doesn't understand.
If you see **`I don't know how to`** _______ , you are trying to run a Logo word on the HP Stream that it doesn't understand.
If you get an error message, see if you can figure out what you did wrong by asking a classmate for help. If all else fails, ask your teacher.
Watch the FTDI cable <u>during</u> download. If it blinks fast, the AppBoard is working.
Watch the FTDI cable <u>after</u> download. If it slowly blinks red and green, the AppBoard is working.

Going Further

Extra Credit	Task
◆	Determine the average daily temperature for various locations on your campus.
◆◆	Collect temperature measurements at one location for an entire day. Organize, represent, and analyze these data in at least <u>two</u> different ways. Compare how these representations and analyses help you to identify and interpret patterns in these data.
◆◆◆	Use your `calibration` project as a reference for adding plotting and analysis to your `NightLight` project. Create a flyer, poster, or presentation from your results.

3.8 Design an Experiment

Getting Started

Throughout these past three units of the Learning by Making curriculum, you became something powerful – a **maker**!

A maker learns by doing. A maker tinkers and troubleshoots and constructs. A maker finds new ways to interact with their environment.

In this lesson, you have one goal. Construct something new with the tools that you have acquired.

Make!

Learning Goals

- ✓ Design an experiment that uses the hardware, software, sensors, and data analysis techniques acquired throughout Units 1, 2, and 3.

- ✓ Create a presentation to share your experimental design process and preliminary results. The format is selected by the instructor and may be a laboratory report, oral presentation, poster, video, model, etc.

Instructions

1. Gather the following materials
   - ☐ BasicBoard
   - ☐ Experiment Plan Worksheet
   - ☐ Additional materials as needed

2. As a group, decide on something you want to investigate using the TMP 36 temperature sensor, OPIC analog light sensor, or LEDs.

3. Based on your goals, load one of your previously created `Logo` projects on your HP Stream. You may need to edit the `uLogo` or `jLogo` files to achieve your goals. Remember to save, compile, and download any new changes.

4. In your notebook, write down an investigation plan.
   - Describe which variable(s) you plan to change and which variable(s) you plan to measure.
   - Assemble a materials list.
   - Decide on measurement and data collection methods. Does this require software changes? Does this require hardware changes?
   - Explain how you will process and analyze data.

5. Use the Experiment Plan worksheet to organize the details of your initial plan.
    - In the **Variables** box, list the variables you plan to change (independent or manipulated variables) and list the variables you plan to measure (dependent or responding variables).
    - In the **Materials** box, list all materials you will need.
    - In the **Predicted Result** box, provide an example of the evidence you expect to collect.

6. Perform a test run of your experiment. Record your observations and data in the **Initial Evidence** box.

7. Complete the Experiment Plan Challenges.

Challenges

Credit	Task
◆◆	Peer feedback is an essential part of doing science. Scientists seek outside views to verify the integrity and validity of their investigations. • Organize your initial experimental data using tables and/or graphs. Describe any patterns or relationships that you can infer from these data. • Swap your Experiment Plan worksheet and initial results with another group. • Provide feedback on the following questions: o Is the evidence gathered relevant to the investigation goals? o Is the evidence gathered reliable? o Are there any gaps or weaknesses in the experimental design?
◆	Refine your investigation plan based on peer feedback. In the **Final Investigation Plan** box, describe your revised procedure. • Explain how your plan will improve relevance to your investigation goals. • Explain how your plan will improve the reliability of your data collection.
◆◆◆	Communicating scientific results to peers is a valuable part of doing science. Scientists share results with the media, publish scientific papers, give presentations, and attend conferences. • Communicate your experiment design process and results through a presentation. As you write your presentation, remember to do the following: o State the explanation you are trying to support. o Include genuine evidence (data + analysis + interpretation). o Explain why the evidence is important and relevant. o Organize your argument in a way that enhances readability. o Use a broad range of words including vocabulary that you have learned. o Use proper grammar, punctuation, and spelling.

Experiment Plan

Investigation Plan

Variables

Manipulated:

Responding:

Materials

Predicted Result

Initial Evidence

Perform a test run of your investigation. Record your observations and data.

Peer Feedback

Reviewed by:

Final Investigation Plan

Revise your initial plan. Explain how your plan will improve relevance to your scientific question. Explain how your plan will improve the reliability of your data collection.

Teacher Approval:

Helpful Hints

If you need to start over, hold down the **`ctrl`** key and **`c`** at the same time. Next, type the command **`start`** and hit the **`enter`** key.
If you already created the experiment, answer **`y`** for **`Would you like to load an existing experiment?`**
If you edit any **`.logo`** files, use the word **`.reload`** before compiling and downloading.
To stop an experiment, type **`..`** (two dots) in the terminal window and hit the **reset** button on the AppBoard.
If you see **`chip not found`**, call the teacher over.
If you see _____ **`undefined`**, you are trying to run a Logo word on the AppBoard that it doesn't understand.
If you see **`I don't know how to`** _______ , you are trying to run a Logo word on the HP Stream that it doesn't understand.
If you get an error message, see if you can figure out what you did wrong by asking a classmate for help. If all else fails, ask your teacher.
Watch the FTDI cable <u>during</u> download. If it blinks fast, the AppBoard is working.
Watch the FTDI cable <u>after</u> download. If it slowly blinks red and green, the AppBoard is working.

Going Further

Extra Credit	Task
◆	Do you agree with the following statements? Write a 1-2 paragraph reflection. "In science, there is no difference between data and evidence." "Observations are facts. Inferences are just guesses."
◆◆	Science is an ongoing process. What new questions should be investigated to build on your research? What future data should be collected to answer your questions?