UNIT PLAN

Make a Smart Light Sensor

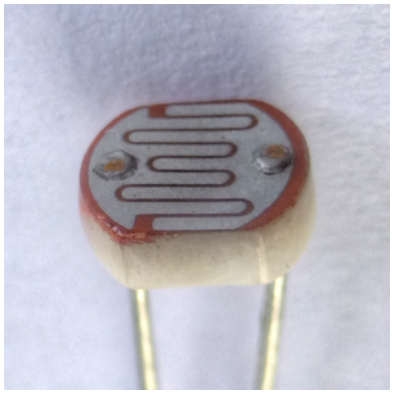
with the Intel® Galileo Gen 2 board

Activity Created by Charles Alba, Bayanihan Labs, UP EEEI

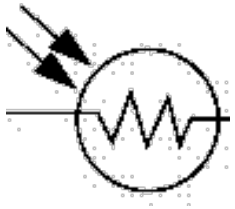
Revised by Tom Seaman, September 15, 2015

What Are Photoresistors?

A ***photoresistor***, or ***light-dependent resistor***, is a variable resistor whose resistance is controlled by light intensity. It works by works by a phenomenon called ***photoconductivity***.



Photoresistor



Symbol for a photoresistor

In line with the trend toward smart devices and systems, we can make efficient use of our lighting systems. In other words, we can build a smart system that ensures that lights only operate when they are needed, ie. when it is dark. Using a photoresistor and some intelligence (the Intel® Galileo Gen 2 board), we can automate this process and maintain constant light brightness in a room, home, or office building.

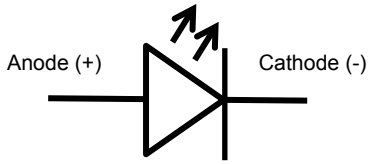
Light-Emitting Diode (LED)

A ***light-emitting diode***, *or* ***LED*** is a type of electronic component that is capable of emitting light.

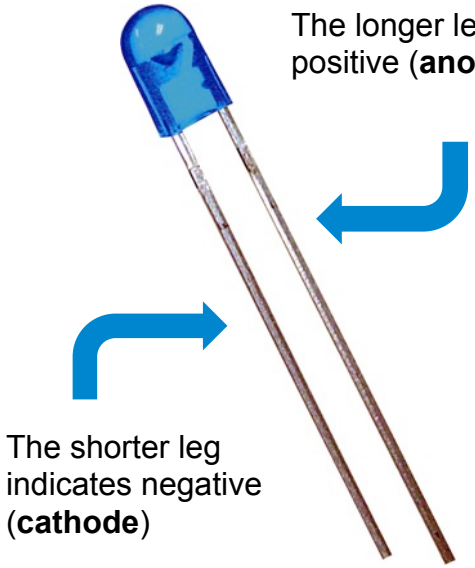
When a suitable voltage is applied across the two leads, energy is released in the form of photons. The material used to manufacture the LED will determine the color of the light emitted.

LEDs are polarized, meaning they will only work when plugged in correctly, since electric current must flow from the **anode** (+) to the **cathode** (-).

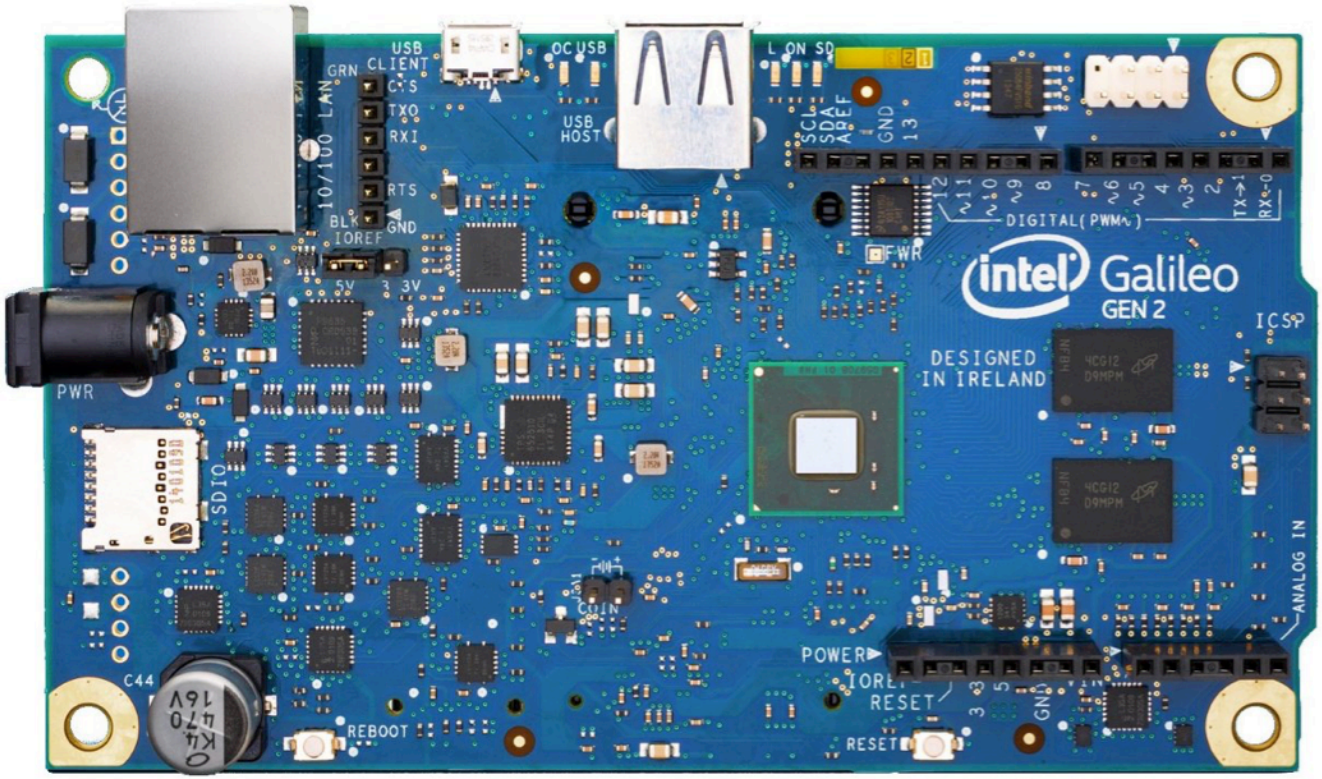
Light Emitting Diodes (LEDs)



Electronic Symbol of LED

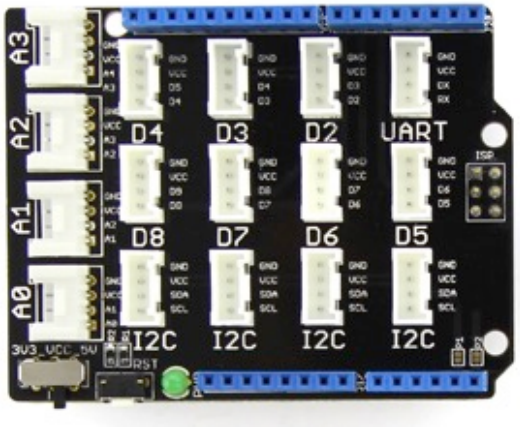


The Intel® Galileo Gen 2 Circuit Board

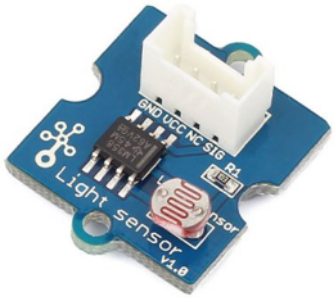


What will you make?

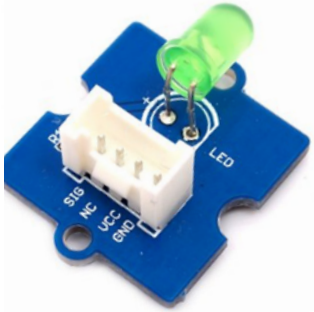
What You Will Need from the Grove Kit



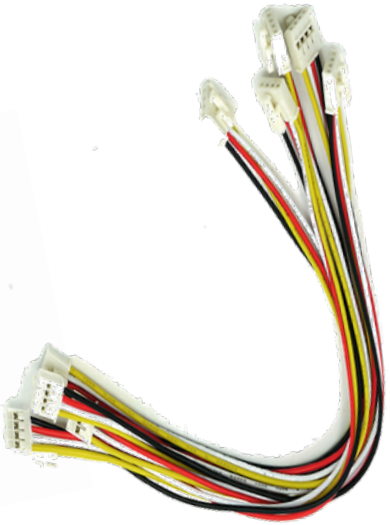
Shield



Light Sensor Module



LED Module



Cables

The shield is hidden under the pink Styrofoam that the RGB Backlight LCD sits on top of.

Figure 1

Coding Tips

```
/*
  Blink
  Turns on an LED on for one second, then

  This example code is in the public doma
 */

// Pin 13 has an LED connected on most Ar
// give it a name:
int led = 13;

// the setup routine runs once when you pr
void setup() {
// initialize the digital pin as an output
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over aga
void loop() {
  digitalWrite(led, HIGH);   // turn the
  delay(1000);               // wait for
  digitalWrite(led, LOW);    // turn the
  delay(1000);               // wait for
}
```

{ Curly Brackets }

Any code you write inside the curly brackets will be executed when the function is called.

// Comments

Comments are notes you leave for yourself that the computer ignores. To write a comment, add two slashes // before you're the text you want ignored.

Case sensitivity

Pay attention to the case sensitivity in your code.

Created by Oliver Chen, Cheryl Dalsin, Shammi Jayaraj

Connect the Galileo Board

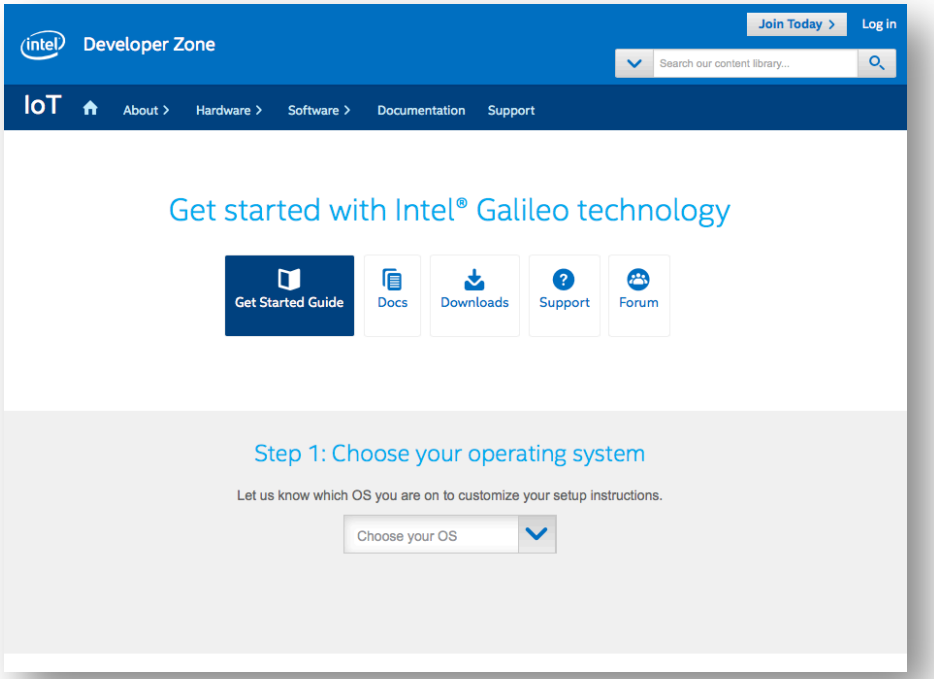
Note: If the set-up is done ahead of time, skip this step and proceed to the next slide, Build the Circuit.

Step-by-step instructions for connecting Galileo are found here:

https://software.intel.com/en-us/iot/library/galileo-getting-started.

Within the step-by-step instructions, when prompted to choose a development environment, choose Arduino.

Proceed all the way through the getting started exercise to the point where you blink the LED on the Galileo board. This affirms the set-up was done correctly.



Caution!

Always make sure the Galileo board is plugged in BEFORE connecting the USB cable to the computer!

Also, always unplug the USB cable BEFORE disconnecting the power from the Galileo.

Doing these steps in the wrong order can permanently damage your board.

Build the Circuit

To simulate the lights of the whole house or room, we will be using a single LED, or more precisely, the LED module in the Grove kit.

Steps:

1. Identify the **shield**, **light sensor module, LED module, green LED,** and **cables** in the Grove Kit.
2. Carefully attach the **shield** to the Galileo Gen 2 board as shown in figure 2.
3. Using the **cables**, connect the **light sensor module** to the shield at A0.
4. 4. Attach the green LED to the LED module, and then using the cables connect the LED module to the shield at D3.
5. Turn on the shield using its switch.
6. Open the Arduino IDE software.
7. Upload the sketch **K11_1.ino** from Github, here: https://github.com/TheCharlesJosh/Bayanihan-Labs-Galileo-Projects/tree/master/K11_1
8. To simulate the changing of brightness, you may opt to cover the sensor with your finger.

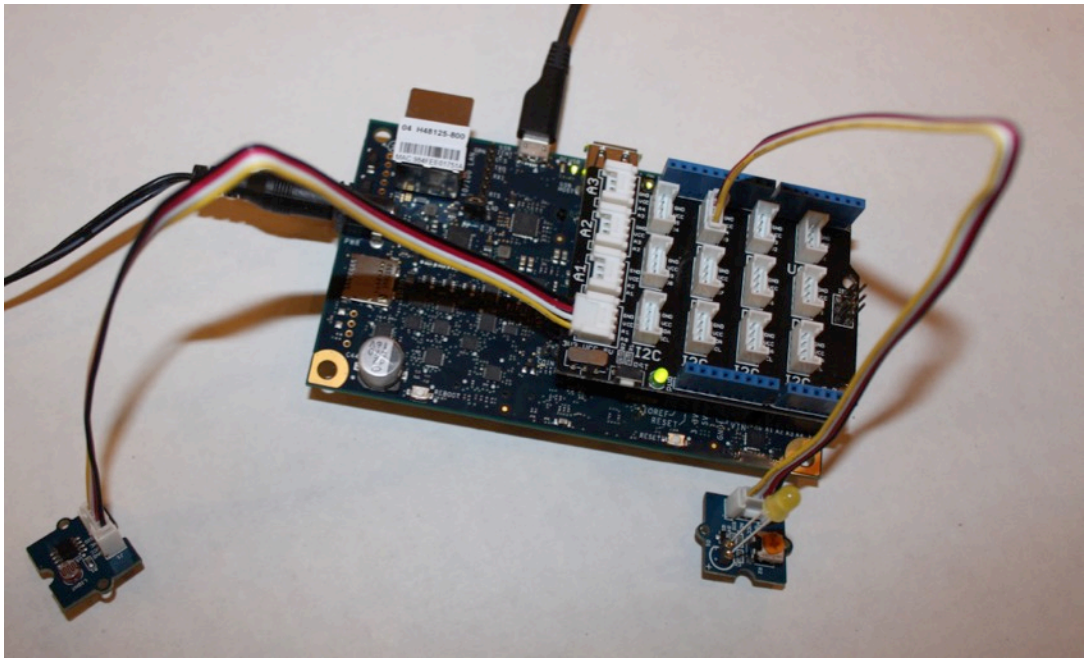


Figure 2

Challenges

Challenge 1

The Intel Galileo board detects the voltage drop being emitted by the module. This voltage changes with respect to the ambient brightness. Galileo then uses this data to change the voltage being applied to the LED via PWM (pulse width modulation), thus making the LED shine brighter or duller.

Using this information, replace the line

"linearlyChange(rawInput)"

with

"triggerByBrightness(rawInput)"

What do you notice is different now in execution?

Try reading the comments in the code as well.

Challenge 2

A computer screen needs to be brighter in bright ambient light, and can be dimmed when the ambient light is lower.

What could you change in the set-up and/or code to adapt the project to make a "smart computer screen" that automatically dims when the ambient light decreases, and brightens when the ambient light is brighter?

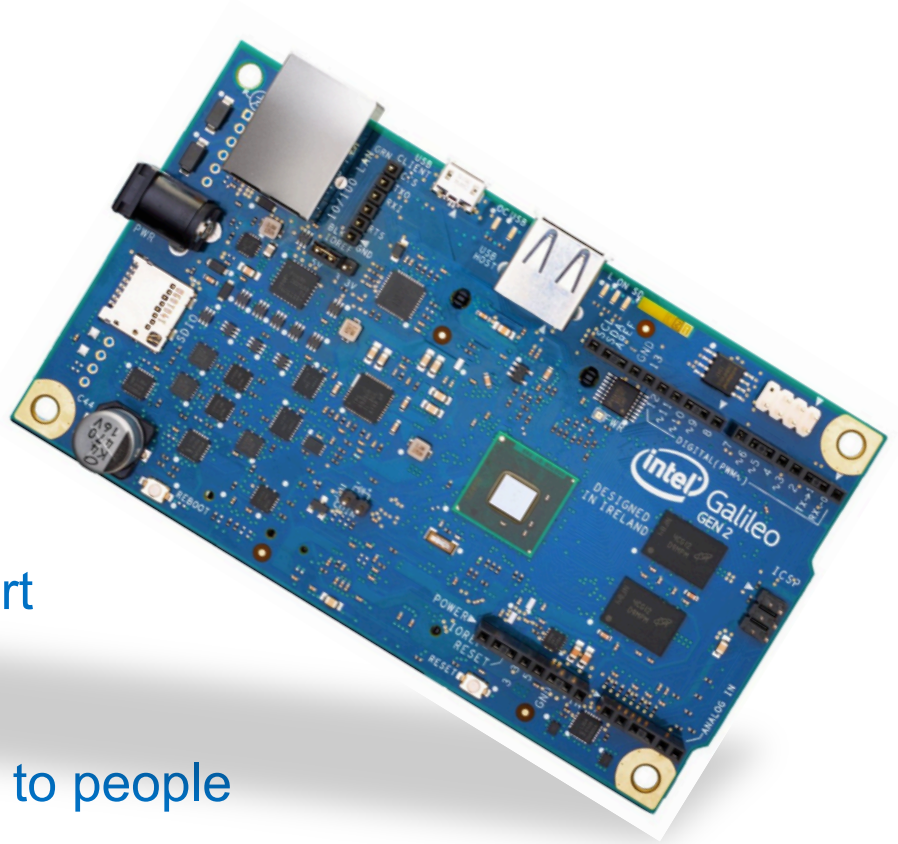
Wrap-Up Discussion

What did we learn?

What are some other applications you can think for marrying the Intel® Galileo board with a photoresistor?

Can you think of any interesting ideas for innovative smart devices for the home?

What kinds of careers are available to people who enjoy this kind of activities?



Troubleshooting Guide

Problem	Resolution

Sketch not working	1. Check Arduino IDE set to correct model, and Port
	2. Check Galileo Port visible in device manager
	3. Try resolution for port not visible
	4. Try upgrading the firmware
	  1. Arduino IDE Help -> Galileo Firmware Update
	  2. Click OK, if you have external power (This step should take ~5 minutes)

Galileo port not visible in device manager	1. Unplug USB 2. Unplug and re-plug power 3. Re-plug USB 4. Wait 5. If not visable after 2.5 minutes, reboot PC and repeat steps 1-4. 6. If still not visible check if *Gadget Serial v2.4* is in device manager
Gadget Serial v2.4 in device manager	1. Install Galileo Driver
Sketch upload complains about missing /dev/tty** Or upload just hangs	Port may be locked, try steps under Galileo port not visible.
Sketch upload complains about missing libraries	1. Close all Arduino IDE instances 2. Install Grove libraries 3. Restart Arduino IDE

If you like this unit plan, you might like one of these other unit plans introducing students to making and coding:

For Teaching Coding and Computer Science

Let's Learn Computer Science 1
Let's Learn Computer Science 2
Let's Learn Computer Science 3
Let's Learn Computer Science 4
Create Your Own Flappy Game

For Teaching Making, Engineering, and Science

Creating with Technology
What Will You Make?
Electrical Engineer for a Day
Unlocking Possibilities
Inspiring Young Engineers
Make a Pong Video Game
Make a Mini Arcade
Make a Smart Toy
Make a Smart Light Sensor
Make a Smart Temperature Sensor

Additional Resources

For an introduction to the benefits of teaching making and coding, and tips for bringing hands-on activities to your classroom, see Gary Stager's paper, "Guide to Creating and Inventing with Technology in the Classroom," found here:

http://innovationtoolbox.intel.com.au/wp-content/uploads/2015/05/18009_IntelEdu_Guide2Making_FA_LR_singles.pdf

To download these and other open-source unit plans, please visit
https://engage.intel.com/community/teachersengage/showcase