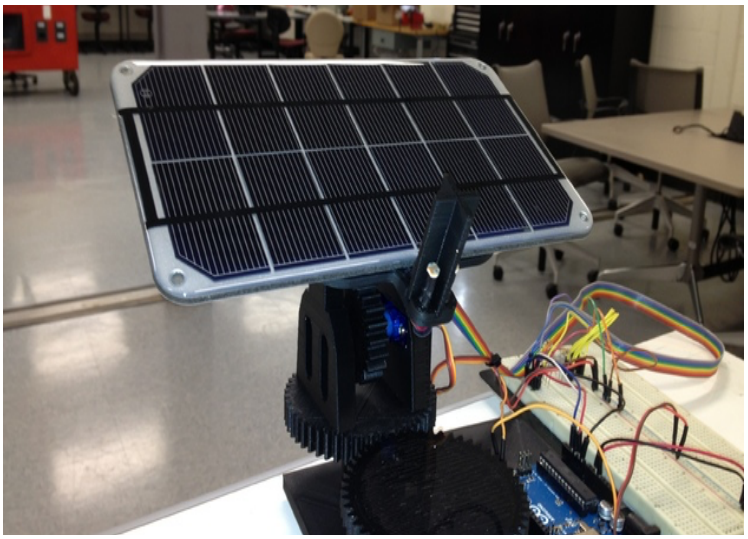
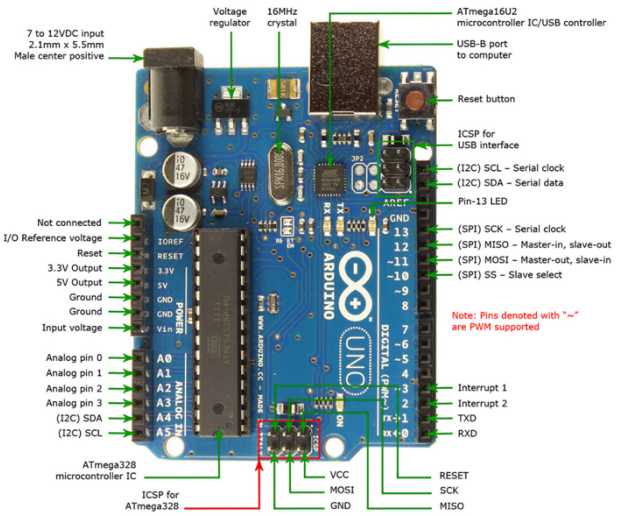
Intro to Arduino: Solar Tracker





"Dual Axis Solar Tracker"

originally developed by **OpenSourceClassroom**

Modified by Project-in-a-Box developers:

Eric Ho, Khai Tran, Phuong Truong

Dear Students,

The ECE 196 course and Project in a Box team would like to credit the original work of Solar Tracker to OpenSourceClassroom. We were inspired by their work and have selected their project to be the inspiration of choice for one of the class projects. As always we would also like to thank the Makers and Inventors behind Arduino, Sparkfun, Instructables, and Sparkfun for their simple tutorials that have been selected in our beginner section of the project. In this sequence, students will learn how to build two different types of solar trackers! We wish our engineers luck on optimizing the design and improving the presentation of the project via documentation and development. As always, have fun!

Sincerely,
P.I.B. Team

Required Downloads:

1) Latest version of Arduino: https://www.arduino.cc/en/Main/Software
2) Latest version of Sublime Text Editor: https://www.sublimetext.com/

Quick References:

1. Please visit Sparkfun to learn more about how breadboards work!
   https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard
2. The Arduino website is a very good resource for grasping circuits and Arduino in general. Use the following link and click on the "Learning" tab, where you will find how to get started, tutorials, and other useful references!
   https://www.arduino.cc/en/Main/Documentation#
3. SparkFun is a vendor that sells a variety of platforms, one of them Arduino. Luckily, they have great resources on the site to help you learn Arduino.
   https://learn.sparkfun.com/tutorials/what-is-an-arduino
4. Adafruit is a DIY (Do-It-Yourself) vendor that sells and utilizes a variety of platforms. Luckily, they have a section dedicated to Arduino! Simply click on the "Learn" tab and follow the Arduino links. It will take you to a large amount of projects for you to build on your own!
   https://www.adafruit.com
5. Instructables is a good project-based resource with a variety of platforms just like Adafruit. Check the website out for more interesting projects!
   http://www.instructables.com

** You must have EVERY challenge signed by a TA or tutor in order receive full credit on the project! Incomplete projects will result in a zero grade. **

Challenge #1: Blinking L.E.D.

Objective: You are going to create a simple circuit using an LED light. The goal is to have the LED light turn on for 2 seconds, and then turn off for 1 second.

Components:

1. 1 Arduino MEGA or UNO
2. 1 Bread-board
3. 2 wires (choose one black/brown for GROUND, and the other is any color)
4. 1 resistor (330 ohms)
5. LED light

Wiring: Using the circuit schematic provided, wire the Arduino.
Note: For the LED light, shorter leg goes to GND (ground), and longer leg goes to a pin. Ground is the reference point in an electrical circuit which voltages are measured (usually zero).

Coding/Programming: Open the Arduino IDE software and write the following code:

```
// assign a value of your choice to the integer variable led
int led = ________;

// setup() function runs once when the Arduino turns on or is reset
void setup() {
  // initialize the digital pin as output.
  // assign variable led as this output
  pinMode(led, OUTPUT);
}

// loop() function runs the code inside repeatedly while Arduino is still on
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(2000);               // wait for two seconds
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

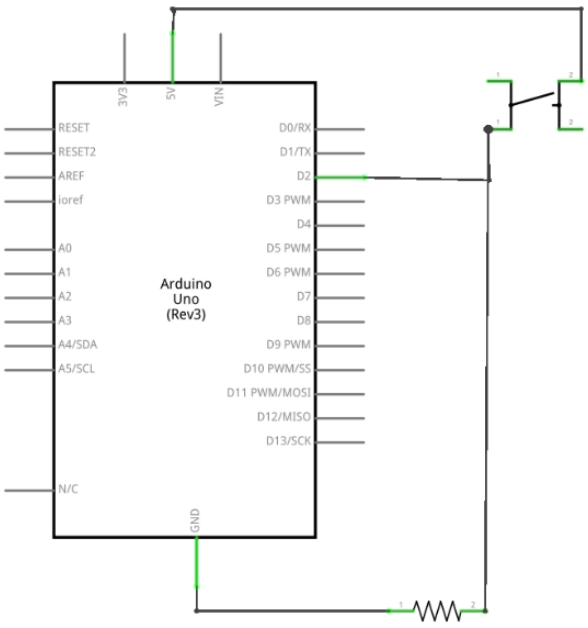
Note: Everything behind these slashes // are just comments, not actual code. The // indicates that everything after it will be seen as a comment and ignored by the compiler and ARDUINO.

Note: WAIT! DON'T TAKE DOWN THE CIRCUIT! You'll need it for the next challenge. ☺

Terminology:
LED (Light Emitting Diode): A diode is a device that only allows for the flow of electricity to pass in one direction. A LED is a diode that generates a specific wavelength of light when a voltage is applied across its leads.
digitalWrite(): Is a built in function that outputs to a specific pin to turn something ON (HIGH) OR OFF (LOW). In Arduino Coding, HIGH translates to ON which is 1 and LOW translates to OFF which is 0 (to the devices).
pinMode(): Is a built in function that initializes a pin as an input or output (default is output). Meaning, it tells the ARDUINO, "Hey! This pin will be your target for input or output!"
delay(): Is a built in function that does what the name says: delays. It stops everything in the Arduino and holds for however many milliseconds specified. In the code above, the LED is turned on with digitalWrite, then, delay of two seconds occurs, then digitalWrite turns the LED off, and the delay of one second occurs before looping back up. Normally, depending on the circuit, a delay will be problematic for sensors because once a delay occurs, this stops everything in the Arduino, *including* the reading of data.

Challenge #2: Blinking L.E.D. with Button

Objective: Building from the last circuit you've created, you are going to take what you learned to construct a more complicated circuit involving both a button and LED (simply integrating a button circuit into the existing LED circuit). You will keep the previous circuit and continue building it. The goal is to turn the LED on and off at the press of a button. Press the button, and the LED turns on (button/LED should STAY on until pressed again). Press it again, and it turns off (STAYS off). The following circuit diagrams show the button circuit ONLY, make sure to consider the LED circuit (from the last challenge) when you are coding:



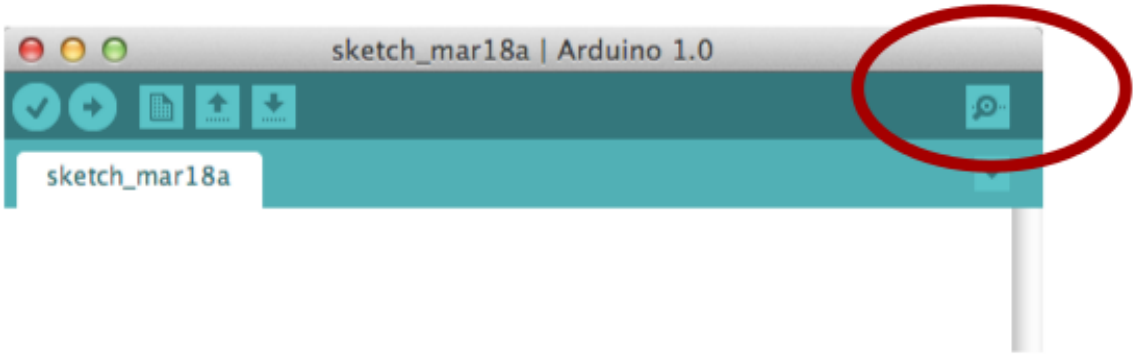
Components (for the button circuit):

1. 1 Arduino MEGA
2. 1 bread board
3. 5 wires (choose 2 black or brown for GROUND, 2 red for 5V and the other any color)
4. 1 push button
5. 1 resistor (330 ohms)

Wiring: The beauty of the Arduino is the fact that it will allow you to integrate both circuits without needing to disconnect the LED circuit. Using the circuit from the previous challenge, and the circuit in the figure, wire the ARDUINO (essentially add the circuit on the right to the circuit from the previous example). You will not be connecting the button and LED in series! **They are two separate circuits communicating with each other via the Arduino!**

Note: Remember to keep the LED circuit intact! You'll need it for a later challenge. ☺ You can take down the button circuit.

Challenge #3: Printing to the Serial Monitor



The serial monitor is a pop-up window that acts as a separate terminal that communicates by receiving and sending Serial Data. Its job is to allow you to both send messages from your computer to an Arduino board (over USB) and also to receive messages from the Arduino. This challenge will make use of the serial monitor and allow you to print text. As indicated in the figure, to access the text printed, click on the icon in the top right corner of the Arduino IDE (software interface). The serial monitor will open as a new window if the Arduino is connected to the computer and there is text to display. Make use of the serial monitor to view data or debug your program!

Objective: Using the following code, print the line "HELLO, WORLD!" into the serial port once, print an empty line, and lastly, print the names of the members in your group repeatedly. **No wiring of electrical components is necessary in this challenge**. You are simply going to connect the Arduino to the computer.

Begin by placing the following line in the void setup() function:

```
Serial.begin( 9600 );
```

This opens the serial line for data transmission via printing. The 9600 is the baud rate, which is the number of bits per second being transferred to the serial line for printing (data transmission rate).

```
Serial.print("_____insert text_____");
//Prints the text between the quotation to the serial monitor

Serial.print("\n");
```

Note: The \n means "to the next line". This is equivalent to hitting the enter/return key on Microsoft Word. How many of these do you need to print an empty line?

To see the text in the serial monitor, simply click on the button on the top right hand corner of the Arduino IDE window.

Challenge #4: Building Your Own Library

Follow the Writing a Library for Arduino tutorial provided by the Arduino website:

https://www.arduino.cc/en/Hacking/LibraryTutorial

Please notify a TA that you have reached this part of the project. You will need to set up an LED circuit on the breadboard. Begin by setting up a sketch and use pin 9 for the LED (instead of pin 13) for the code below:

```
int pin = 13;

void setup()
{
  pinMode(pin, OUTPUT);
}

void loop()
{
  dot(); dot(); dot();
  dash(); dash(); dash();
  dot(); dot(); dot();
  delay(3000);
}

void dot()
{
  digitalWrite(pin, HIGH);
  delay(250);
  digitalWrite(pin, LOW);
  delay(250);
}

void dash()
{
  digitalWrite(pin, HIGH);
  delay(1000);
  digitalWrite(pin, LOW);
  delay(250);
}
```

Have a tutor or TA check your circuit and sketch.

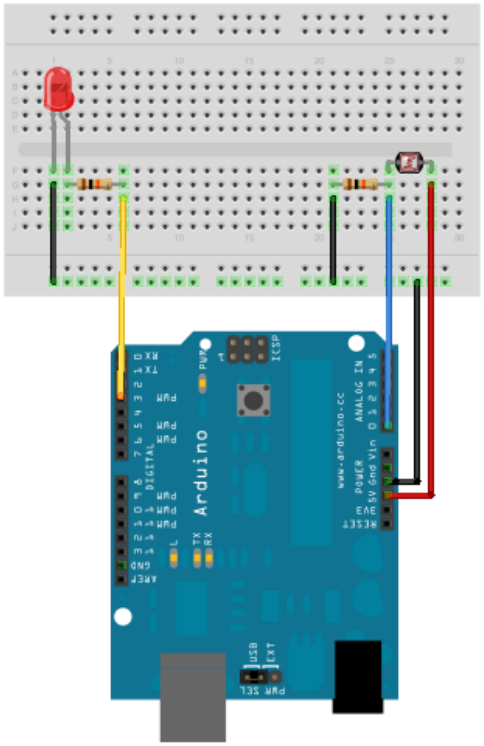
Next, follow the tutorial to develop your own libraries written in C++. In the end, you will have one .cpp file and one .h file. Have a tutor or TA check your circuit and sketch once again.

Answer these questions briefly:

What is #include?

What is an object?

Challenge #5: Photo-resistors with PWM's



Task: You are going to create a photo-resistor and LED light based circuit. A photo-resistor changes its resistance based on how much light it detects from the environment. The goal is to use the photo-resistor to turn on the LED (as you begin to cover the photo-resistor, the LED should turn brighter and brighter until the photo-resistor is completely covered). You will need:

1. Arduino Uno
2. Bread-board
3. 5 wires (2 red for voltage/pin, 2 black (ground), 1 green wire)
4. A resistor (330 ohms)
5. Photo-resistor
6. LED light (the long leg goes to a PIN, and the short end goes to GROUND)

WIRING:
Using the circuit in the two figures, wire the RED BOARD:

TYPE IN THE CODE:

**NOTE: Everything behind these slashes // are just comments, not actual code. When you put // everything after it will be seen as a comment and ignored by the compiler and Arduino.

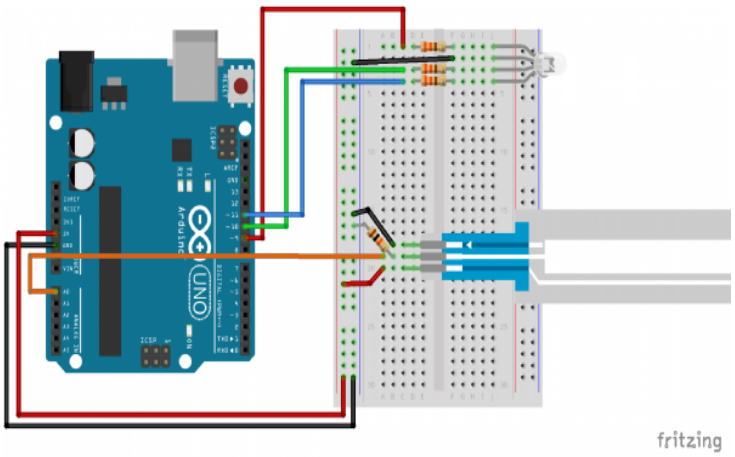
```
const int sensorPin = _____;
const int ledPin = _____;
// Fill in the number of LED pin.

int lightLevel;
void setup()
{
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  lightLevel = analogRead(sensorPin);
//Use linear equation to map light level to PWM minimum and maximums
  lightLevel = _____;
  analogWrite(ledPin, lightLevel);
```

Terminology:
Pulse-width modulation (PWM): https://www.arduino.cc/en/Tutorial/PWM analo
analogWrite(): Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. https://www.arduino.cc/en/Reference/AnalogWrite
analogRead(): Reads the value from a specified analog pin. Arduino contains a 6 channel, 10-bit analog to digital converter. This means that it will map input voltages between zero to five volts into integer values between 0 and 1023. https://www.arduino.cc/en/Reference/AnalogRead

Challenge #6: Soft-potentiometer with RGB LED

Objective: You are going to create a soft potentiometer based circuit. The goal is to turn on the LED based on the position of where you put your hand on the soft potentiometer. You will need:



Components:

7. Arduino Uno
8. Bread-board
9. 7 wires
10. 3 Resistors (330 Ohms)
11. 4 Legged LED (the LONGEST leg goes to ground! Three other legs could go to any pin)
12. Soft potentiometer

WIRING:

Using the circuit in the figure, wire the RED BOARD.

TYPE IN THE CODE: **NOTE: Everything behind these slashes // are just comments, not actual code. When you put // everything after it will be seen as a comment and ignored by the compiler and Red Board.

Fill in the highlighted blanks based on how you wire the Arduino. Touch the soft potentiometer to change the resistance values! The higher up you touch, the lower the resistance! The lower down, the higher the resistance! Readings can be seen if you click on the top right hand button on the ARDUINO coding interface.

```
int potPin = __________;    // select the input pin for the potentiometer
int ledPin = _________;   // select the pin for the LED
int led2Pin = _________;
int led3Pin = _________;
int val = 0;       // variable to store the value coming from the sensor

void setup() {
  Serial.begin(9600);
  pinMode( _________ , _________));  // declare the ledPin as an OUTPUT
  pinMode( _________, _________));
  pinMode( _________, _________));
}
void loop() {
  val = analogRead(potPin);    // read the value from the sensor FROM ANALOG PIN
  Serial.print("Potentiometer: ");
  Serial.println(val);

  if (val >= _________ && val < _________ ) {
    digitalWrite(__________, _________);  // turn the ledPin on
  }
  else if(val >= _________ && val < _________) {
    digitalWrite( _________ , _________));
  }
  else {
    digitalWrite( __________, _________));
  }
    delay(2000);
    digitalWrite(_________, LOW);   // turn the ledPin off
    digitalWrite(_________, LOW);
    digitalWrite(_________, LOW);
}
```

CAN YOU MIX COLORS? Try turning on more than one LED pin at once.

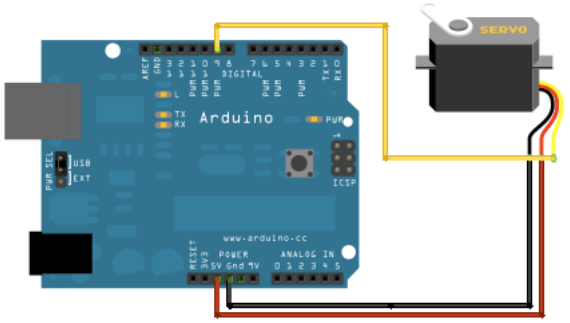
Challenge #7: Control a Servo with a TrimPot

Objective: Recreate the exact same circuit from the Soft Potentiometer challenge, and replace the Soft Potentiometer with a knob potentiometer. What values do you read from the analog pin? What is the minimum? _____. What is the maximum? _____. Record the values. Where do these numbers come from (what generates these numbers), and what voltages do they correlate to? Use the tutorial on the potentiometer to help you: https://www.arduino.cc/en/Tutorial/Knob

Restrict the knob potentiometer values from zero to 90, and adjust the range values (in the if statements) accordingly. Write the linear equation below.

Equation: _____

Objective: Now, let's control a single servo with our Arduino. In this circuit, you will utilize the servo library. Your goal is to turn the servo to the right and then to the left repeatedly.



Coding:

```
#include <Servo.h>

Servo __________; // Create object

void setup() {
       servoMain.attach(_______); // Attach to a specific pin
void loop() {
       servoMain.write(________); // Highest angle
       delay(1000);
       servoMain.write(_________); // Lowest angle
       delay(1000);
}
```

Objective: Once you have understood how a servo and knob potentiometer works and how to restrict values from zero to 90, you are ready to control a servo with the knob potentiometer. Find the micro servo in your kit. Create a circuit in which turning the potentiometer would also turn the small servo (in sync). Restrict the range values of the potentiometer from zero to 60 (to represent the angle degree of the servo). Print the values onto your serial monitor. Note: DO NOT use the map function to restrict the values. You must use a linear equation based on what you know about analog inputs.

Challenge #8: Servo Sweep!

Objective: You will combine what you know about LED's, photo-resistors, and servos to complete a servo sweep circuit. A photoresistor will be attached on the arm of a servo motor. It will sweep back and forth to look for the brightest LED light in front of it as shown in the picture. (Wires are not hooked up, image is only used for visualization purposes)

Components:

1. 1 Arduino MEGA or UNO
2. 1 Bread-board
3. 1 Photoresistor
4. 1 Microservo
5. 5 LED lights of the same color
6. Jumper wires