

Name: \_\_\_\_\_

## Turning on an LED

Use a breadboard and a power supply without the Arduino microprocessor for this exercise. The purpose of this is to give you hardware trouble shooting skills without using the Arduino. (Before programming the Arduino you should always make sure that the connected electronic hardware components are working using a Multimeter.)

### Goal:

- Hookup two LED's, each at a different brightness, using a 5-volt power supply, resistors, and a breadboard.

### Steps:

- Connect a breadboard to ground and 5V (**use proper wire colors and check with a voltmeter!**)
- Wire the LED's as shown on the white board.
- Adjust the resistor values to achieve the two different brightness levels. You can use resistors in series or parallel to get different resistance value using Ohm's law. Use the manufactures data sheet in the back of this packet to identify the LED's anode (+) and the LED's cathode (-) connects. Hooking up the LED incorrectly will damage it!

**NOTE:** The Arduino can source 40mA or sink 40mA at 5 volts per pin and can source a TOTAL of 200mA's using all it pins. Exceeding these specifications WILL DAMAGE the Arduino!

Instructor/TA Signoff on Hardware: \_\_\_\_\_

**Draw the Circuit Diagram of the two circuits you built here:**

Name: \_\_\_\_\_

### Wire a Contact Switch for Signal Input

Use a breadboard and a power supply without the Arduino for this exercise.

#### Goal: (Using the circuit diagram in the back of this packet)

- Wire the mini contact switch (SPST) so that it will reliably provide a logic level high and a logic level low.
- Understand how to use a “pull down” resistor so there are **NO “Floating Inputs!”**

#### Steps:

- Connect the breadboard to ground and 5V (**use wire proper colors!**)
- Wire the mini contact switch as shown, but instead of the Arduino, connect the output to the multimeter (set the multimeter to read **voltage**).

Instructor/TA Signoff on Hardware: \_\_\_\_\_

**Draw the Circuit Diagram of the circuit you built here:**

**Draw a Circuit Diagram of a (DPST) switch here:**

Name:

---

### **Connect a Potentiometer for Signal Input into the Arduino**

Use a breadboard and a power supply without the Arduino for this exercise.

#### **Goal: (Using the spec sheets in the back of this packet)**

- Wire a Potentiometer so that it will reliably provide a variable analogy signal to the Arduino.

#### **Steps:**

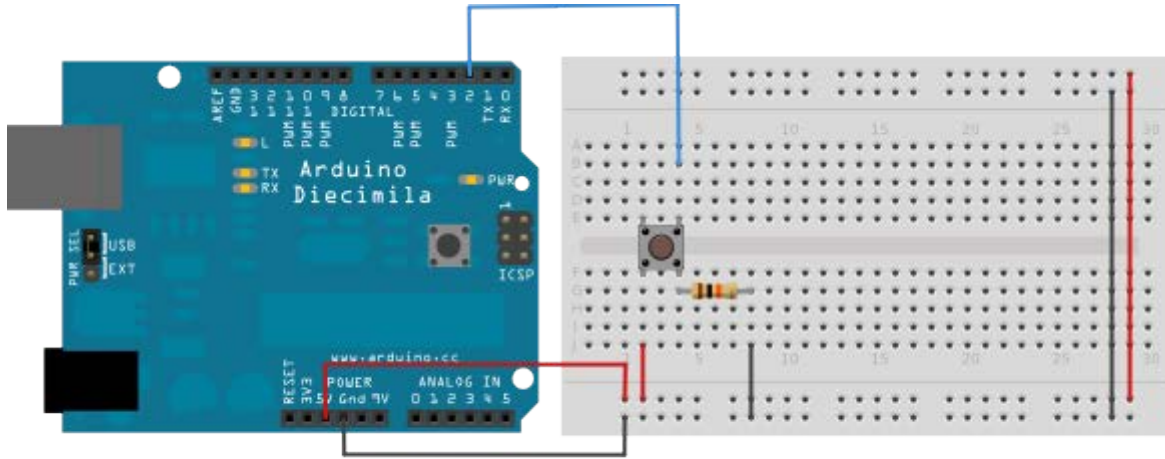
- Identify the fixed and variable resistance pins of the Potentiometer.
- Connect the Potentiometer as directed to create a Voltage Divider circuit. Use a Multi-meter instead of the Arduino to measure the Voltage Signal.

Instructor/TA Signoff on Hardware: \_\_\_\_\_

Name: \_\_\_\_\_

## Arduino Programming

**Goal:** Download and install the Arduino programming environment onto your laptop computer. There are various sample programs from within the Arduino programming environment that you can access from the drop down tab. Using the given electronics kits, build the following circuit. (*What kind of a circuit is this?*)



Program your Arduino with the sample program “Blink.” Please observe and try to understand the code. Then, upload the following code onto your Arduino. Press the button to see what it does.

```
#define buttonPin 2
#define ledPin 13

int buttonState = LOW;
int buttonPressed = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH || buttonPressed == HIGH) {
    digitalWrite(ledPin, HIGH);
    buttonPressed == HIGH;
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

This should have the LED always on once the button is pressed. Then, modify the code so that the LED is only turned on when the button is pressed.

If you don't have a laptop you can use the desktop computers in EBU2, room 311.

The Arduino software can be downloaded directly from the Arduino website (<http://arduino.cc>). If you need help please ask one of your TA's or Chris Cassidy