

Exploring simple circuits using an Arduino board

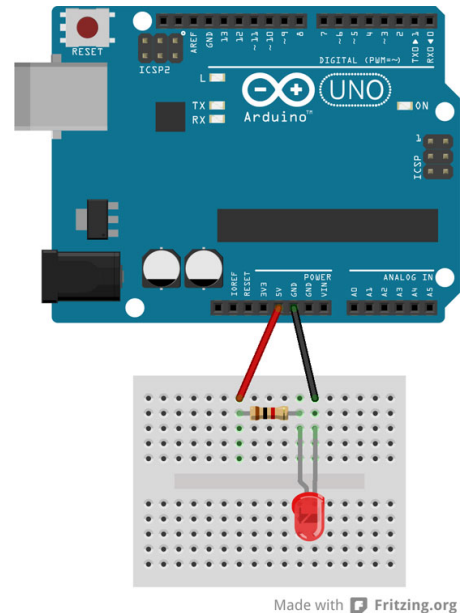
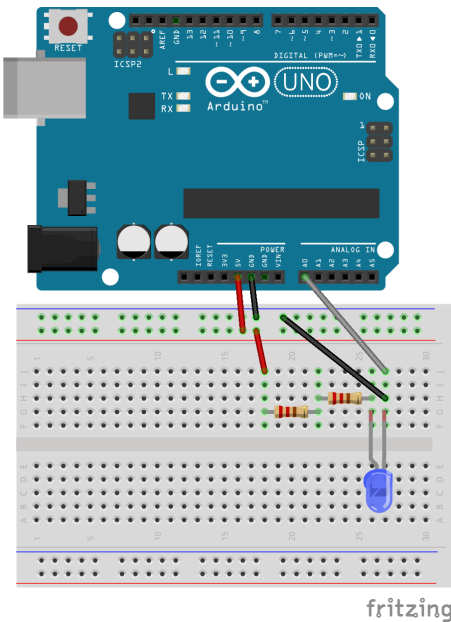
Install Arduino IDE: <http://arduino.cc/en/Main/Software>

Load *Blink* sketch on the board:

<http://arduino.cc/en/Tutorial/Blink>

Create circuit:

You must create a simple series circuit and simple parallel circuit using the Arduino kit. There must be at least one LED and two resistors in both circuits. The LED needs one by itself - this can be included in the count of 2. You must draw the circuits to include the lab write-up.



What to include in the write-up: the circuit diagram the number and resistance values for the resistors; the color of the LED; the voltage you chose (5V or 3.3V - **be sure to read the voltage directly using a multimeter from the leads attached to the board**); the voltage and current of the circuit in all branches; a discussion about the mathematical relationships between voltage, current, and resistance. You should incorporate the [loop and junction rules](#) in these discussions. It's best to independently confirm the value of each resistor using a multimeter. This should be done for each resistor and before the element is placed in the rest of the circuit. Using

a spare spot on a breadboard would work. Be sure to calculate the current in each branch. Don't include the LED in these calculations. While LEDs do have a voltage drop, they should not be treated as Ohmic.

Turn in a write-up using a word processor and don't forget the calculations and diagrams! Here is the [DigiKey Resistor Color Code Calculator](#) if that helps.

① Build series & parallel circuit w/ 1 LED in series w/ resistor pair.

② include wiring schematic for both circuits



② Measure: ① ΔV from source

② ΔV across LED

③ ΔV across resistors

④ Use Ohmmeter to measure resistance

③ Calculate current in each loop

In some simple circumstances, one can read the voltage via one of the analog pins. There are limitations though.

To read the voltage using the Arduino directly, load the [ReadAnalogVoltage](#) sketch from the Arduino software and install it onto the Arduino board. Now when you put a jumper wire into pin A0 of the board you can read the voltage from 0-5V from the serial monitor window. Once the program is loaded and running and the circuit is wired and powered, from the Arduino IDE choose Tools>Serial Monitor. Wherever you touch the wire connected to A0, you'll see the voltage in that branch of the circuit. The 10-bits of data read from pin A0 is mapped to a range of voltage by the program running on the Arduino board. With this setup we can use the Arduino board to both drive a 5V circuit and also to read the voltage.

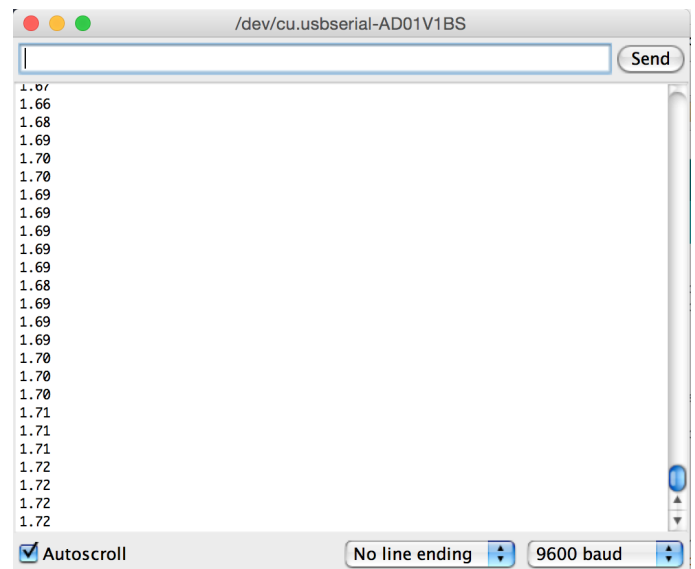
```
ReadAnalogVoltage

/*
  ReadAnalogVoltage
  Reads an analog input on pin 0, converts it to voltage, and prints the result to
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V

  This example code is in the public domain.
  */

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage);
}
```



The screenshot shows the Serial Monitor window for the device /dev/cu.usbserial-AD01V1BS. The window displays a list of voltage readings: 1.67, 1.66, 1.68, 1.69, 1.70, 1.70, 1.69, 1.69, 1.69, 1.69, 1.68, 1.69, 1.69, 1.69, 1.70, 1.70, 1.70, 1.71, 1.71, 1.71, 1.72, 1.72, 1.72, 1.72, 1.72. The window has a 'Send' button at the top right and a 'No line ending' button at the bottom right. The 'Autoscroll' checkbox is checked.

```
/*
  ReadAnalogVoltage
  Reads an analog input on pin 0, converts it to voltage, and prints the result to the serial monitor.
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.
  This example code is in the public domain.
  */

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage);
}
```