

Project #2 – Fading

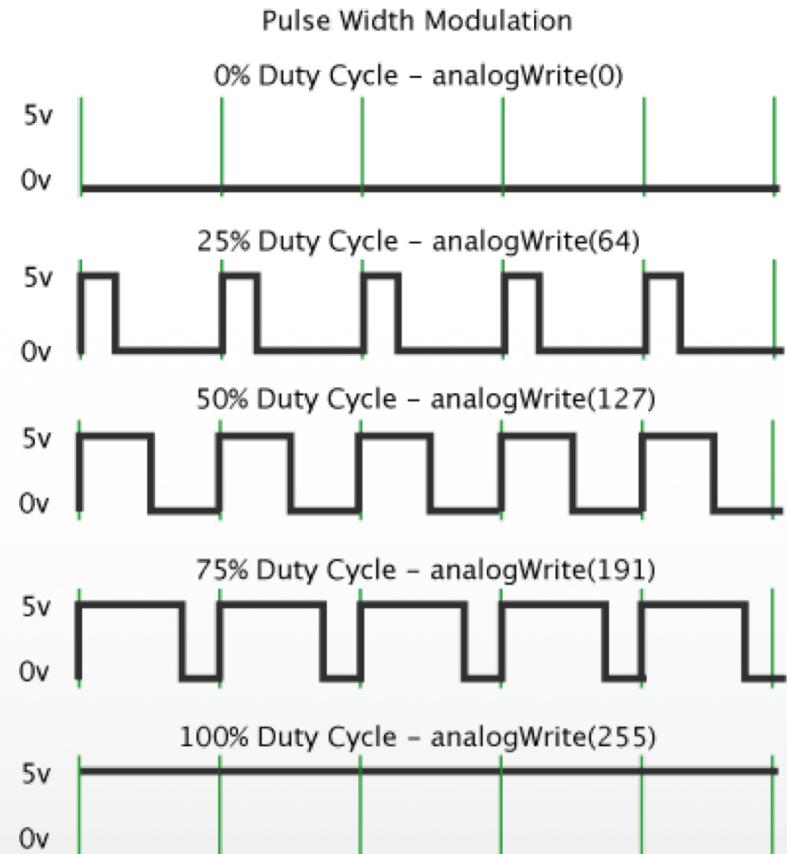
Introducing a new command...

```
analogWrite(pin, val);
```

pin – refers to the OUTPUT pin
(limited to pins 3, 5, 6, 9, 10, 11.)
– denoted by a \sim symbol

val – 8 bit value (0 – 255).

0 => 0V | 255 => 5V

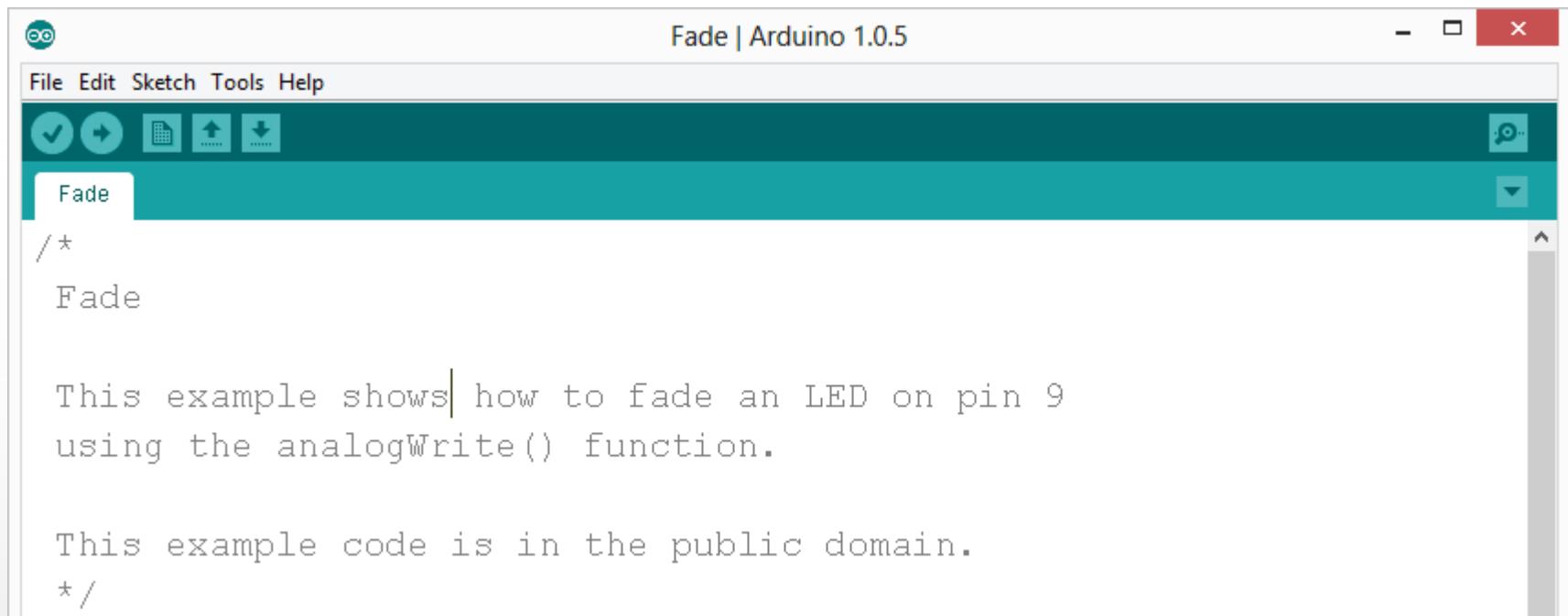


Move one of your LED pins over to Pin

9

In Arduino, open up:

File → Examples → 01.Basics → Fade



The screenshot shows the Arduino IDE interface with the title bar "Fade | Arduino 1.0.5". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, download, and other functions. A status bar at the bottom right shows the number 39. The main code editor window displays the following code:

```
/*
Fade

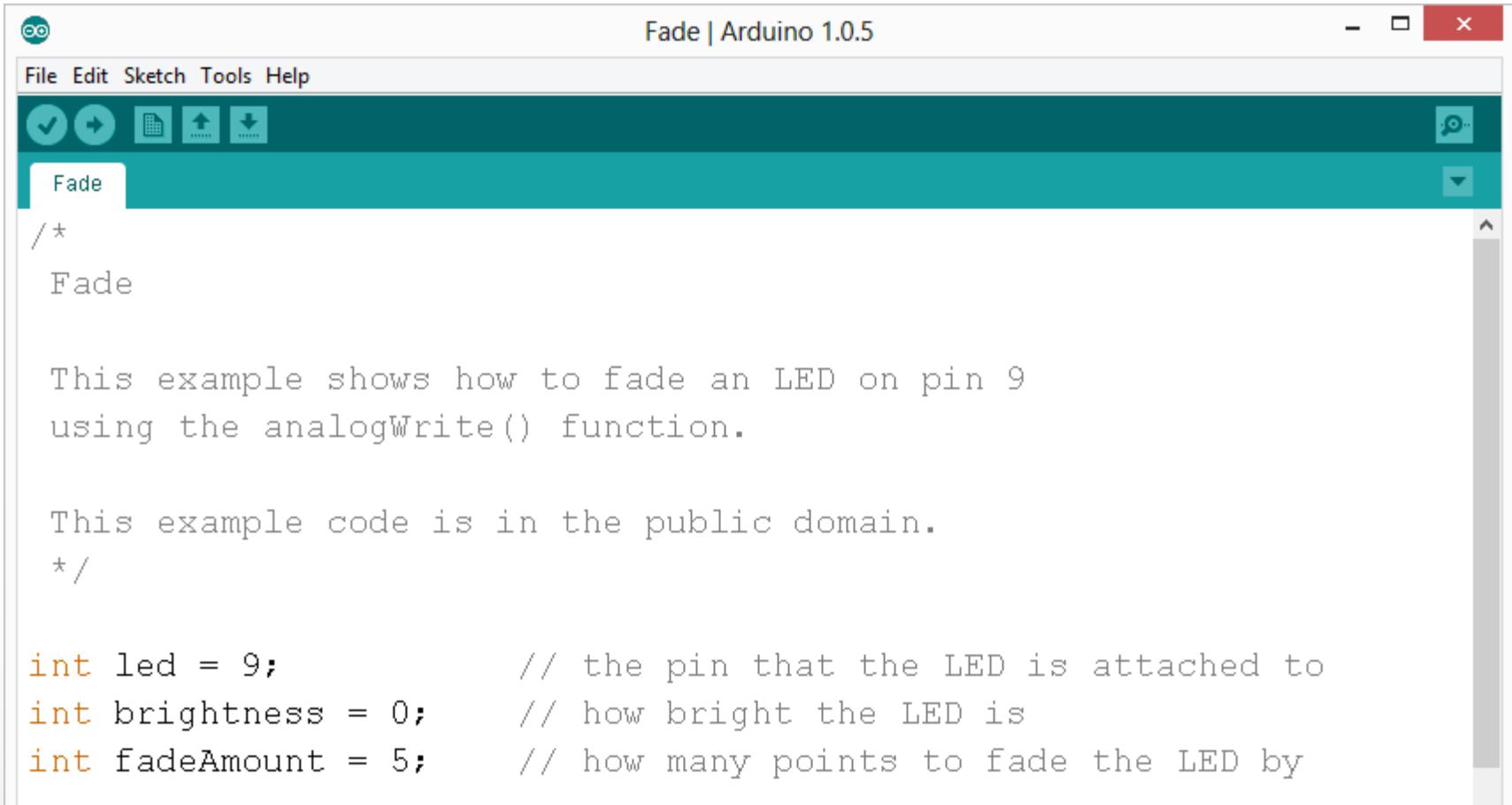
This example shows how to fade an LED on pin 9
using the analogWrite() function.

This example code is in the public domain.
*/
```

The code is a basic example for fading an LED connected to pin 9 using the `analogWrite()` function.



Fade - Code Review



The screenshot shows the Arduino IDE interface with the title bar "Fade | Arduino 1.0.5". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for upload, download, and other functions. A sidebar on the left shows the sketch structure with "Fade" selected. The code editor contains the following code:

```
File Edit Sketch Tools Help
Fade
Fade

/*
Fade

This example shows how to fade an LED on pin 9
using the analogWrite() function.

This example code is in the public domain.

*/
int led = 9;          // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by
```

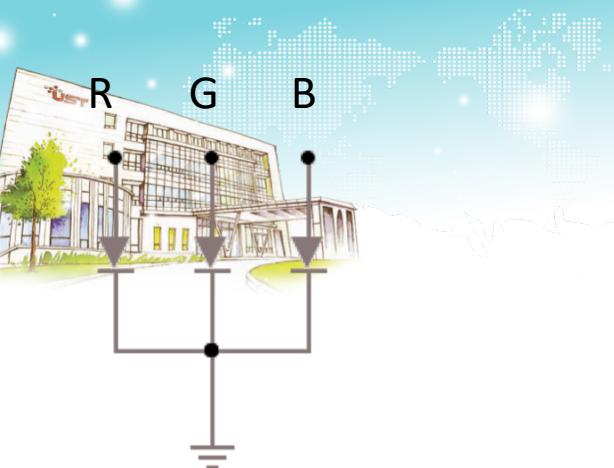
Fade - Code Review

```
void setup() {  
  // declare pin 9 to be an output:  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  // set the brightness of pin 9:  
  analogWrite(led, brightness);  
  
  // change the brightness for next time through the loop:  
  brightness = brightness + fadeAmount;  
  
  // reverse the direction of the fading at the ends of the fade:  
  if (brightness == 0 || brightness == 255) {  
    fadeAmount = -fadeAmount ;  
  }  
  // wait for 30 milliseconds to see the dimming effect  
  delay(30);  
}
```



A few simple challenges# 2 -- Fading

- **Challenge 2a –**
Change the rate of the fading in and out.
- **Challenge 2b –** Use 2 (or more) LEDs – so that one fades in as the other one fades out.

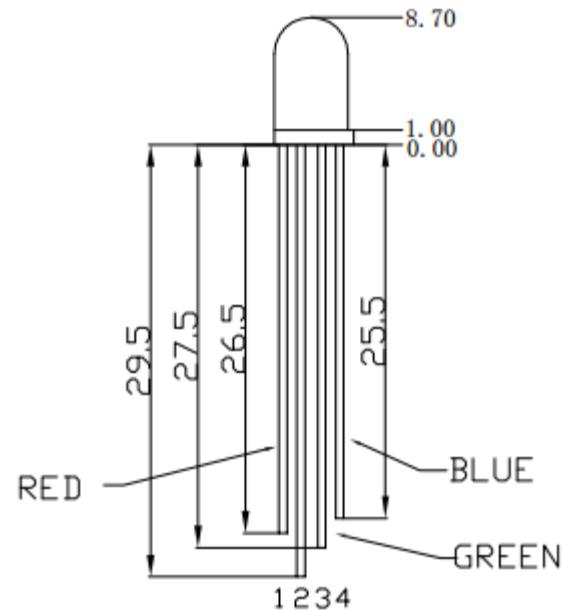


Color Mixing Tri-color LED



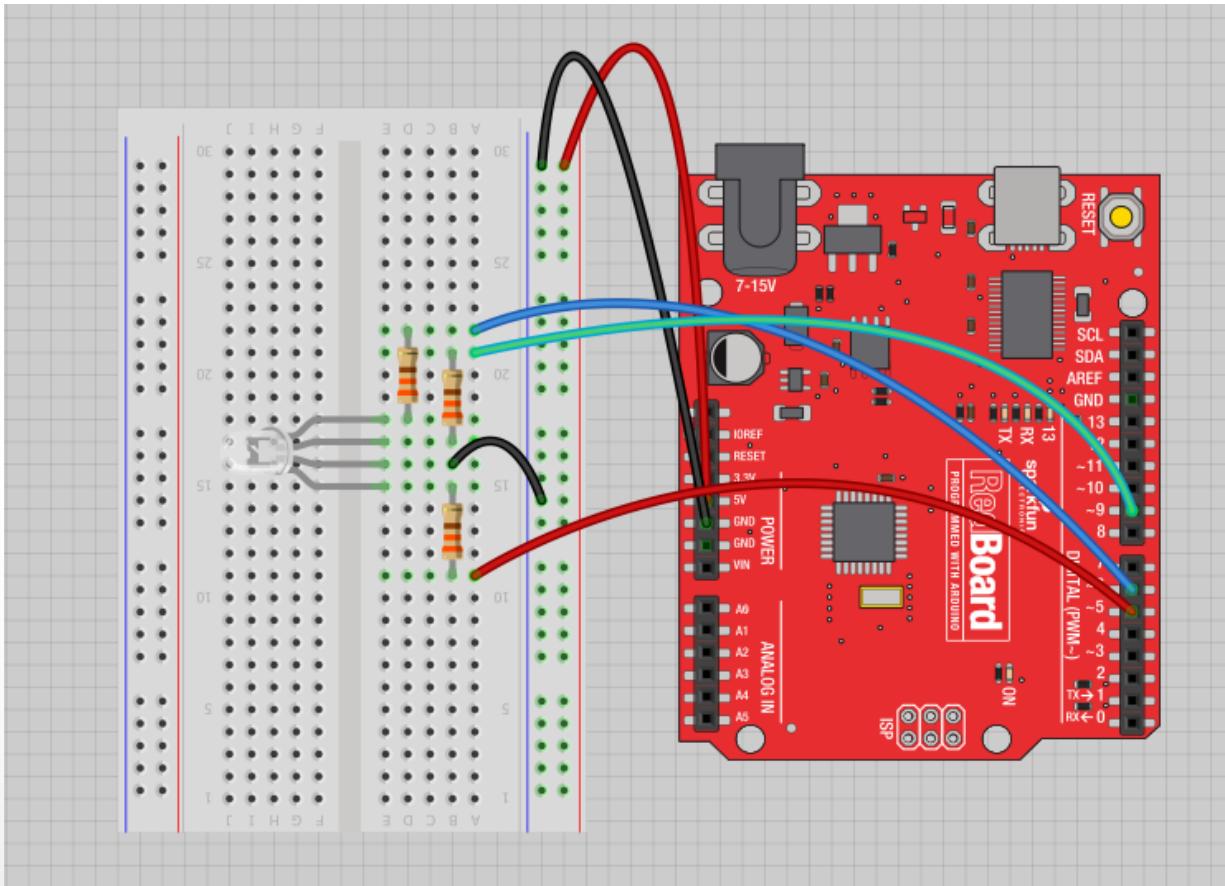
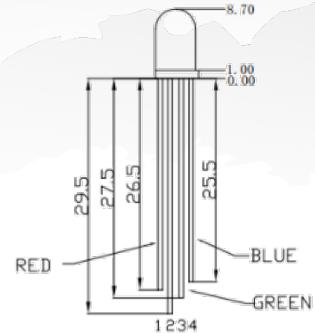
Common Cathode LED

This means the negative side of the LED is all tied to Ground.





Project 3 – RGB LED



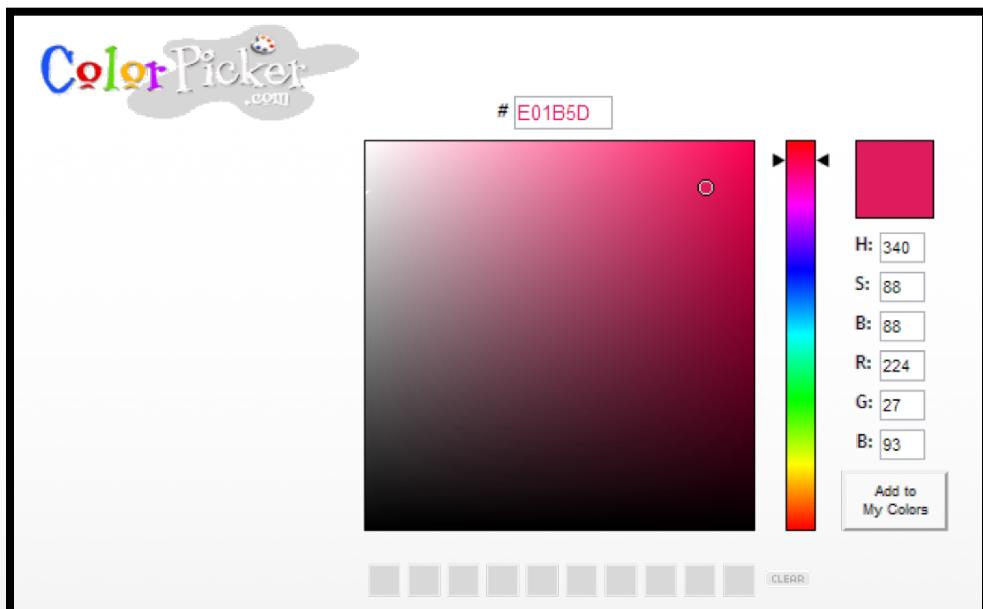
Note: The longest leg of the RGB LED is the Common Cathode. This goes to GND.

Use pins 5, 6, & 9



How many unique colors can you create?

of unique colors = $256 \cdot 256 \cdot 256$
= 16,777,216 colors!



Use Colorpicker.com or experiment on your own.

Pick out a few colors that you want to try re-creating for a lamp or lighting display...

Play around with this with the analogWrite() command.



RGB LED Color Mixing

```
int redPin = 5;  
int greenPin = 6;  
int bluePin = 9;  
  
void setup()  
{  
    pinMode(redPin, OUTPUT);  
    pinMode(greenPin, OUTPUT);  
    pinMode(bluePin, OUTPUT);  
}
```

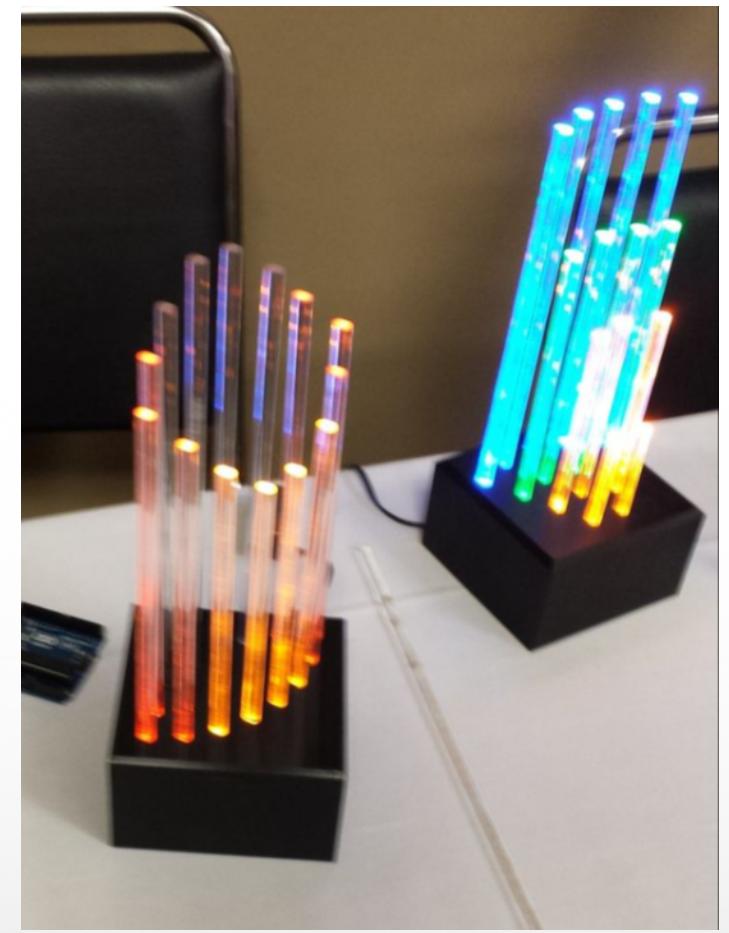
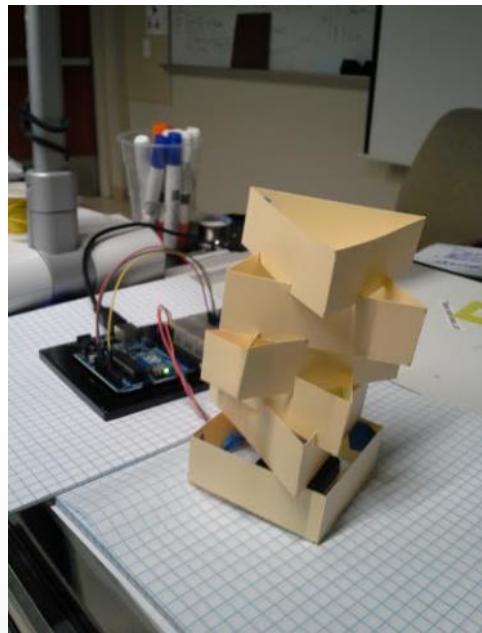


RGB LED Color Mixing

```
void loop()
{
    analogWrite(redPin, 255);
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 255);
}
```



Project: Mood Lamp / Light Sculpture





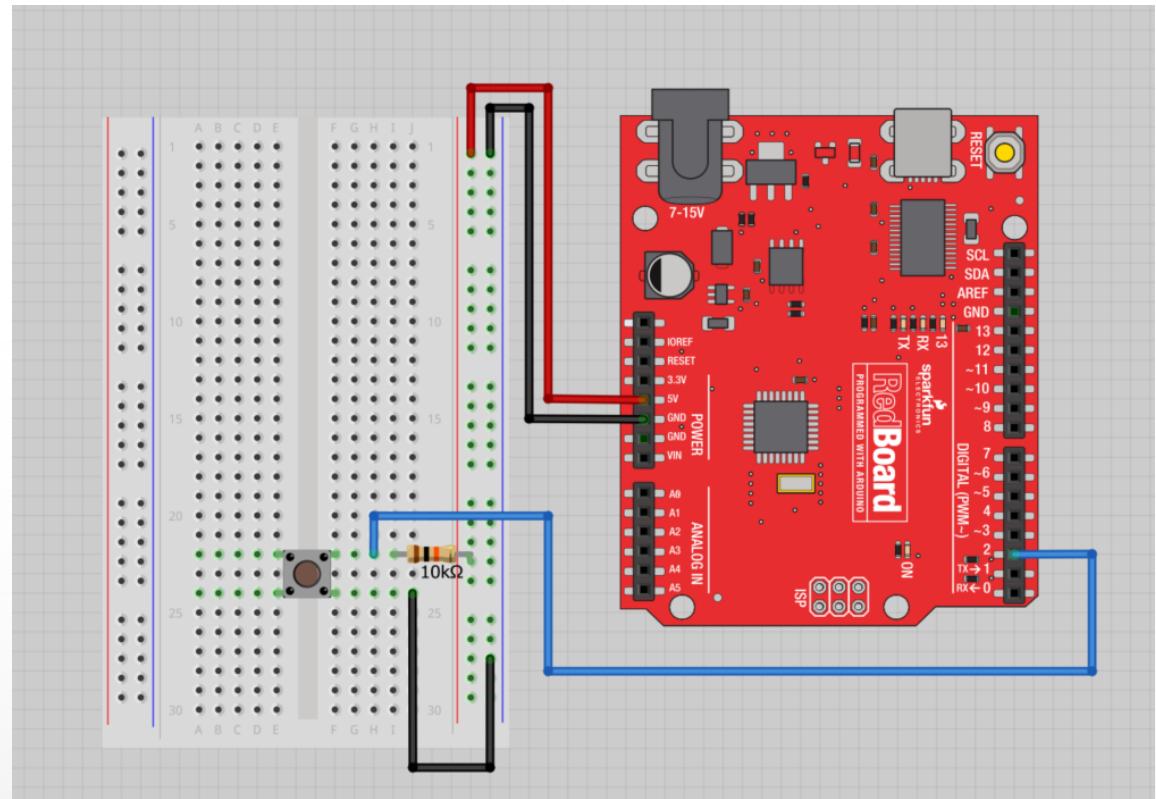
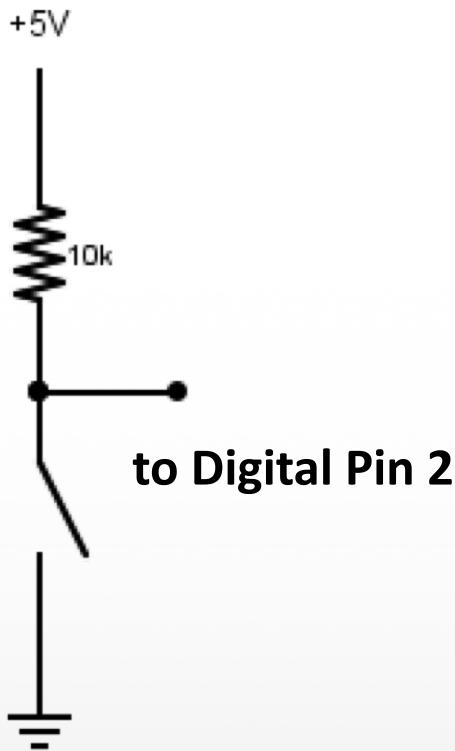
Project #4 – Digital Input

- In Arduino, open up:
- File → Examples → 02.Digital → Button



Digital Sensors (a.k.a. Switches)

Pull-up Resistor ([circuit](#))

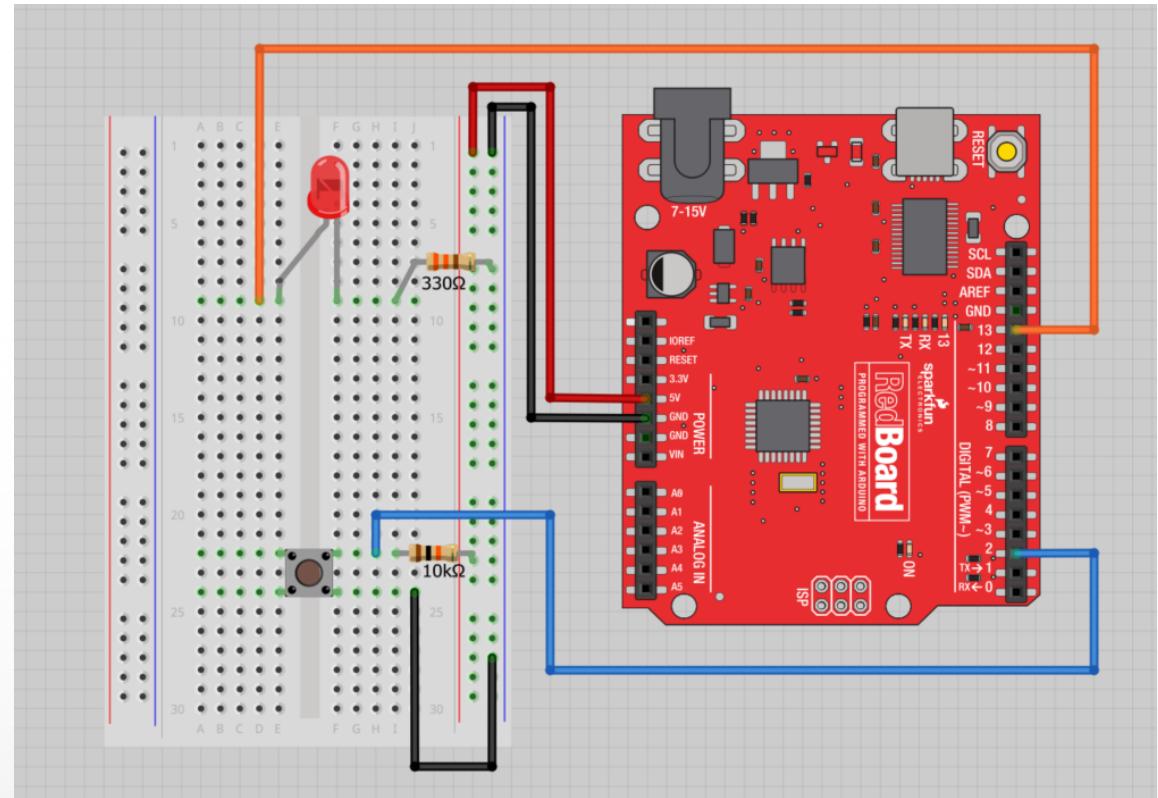
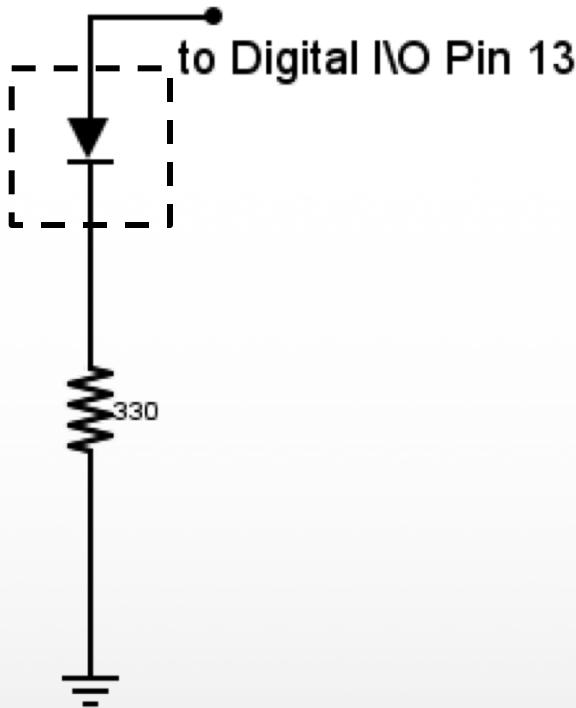




Digital Sensors (a.k.a. Switches)

Add an indicator LED to Pin 13

This is just like our 1st circuit!





Digital Input

- Connect digital input to your Arduino using Pins # 0 – 13 (Although pins # 0 & 1 are also used for programming)
- Digital Input needs a pinMode command:
 - **pinMode (pinNumber, INPUT);**
 - *Make sure to use ALL CAPS for INPUT*
- To get a digital reading:
 - **int buttonState = digitalRead (pinNumber);**
- Digital Input values are only **HIGH** (On) or **LOW** (Off)



- Digital sensors are more straight forward than Analog
- No matter what the sensor there are only two settings: On and Off
- Signal is always either HIGH (On) or LOW (Off)
- Voltage signal for HIGH will be a little less than 5V on your Uno
- Voltage signal for LOW will be 0V on most systems



We declare a variable as an integer.

We set it equal to the function `digitalRead(pushButton)`

The function `digitalRead()` will return the value 1 or 0, depending on whether the button is being pressed or not being pressed.

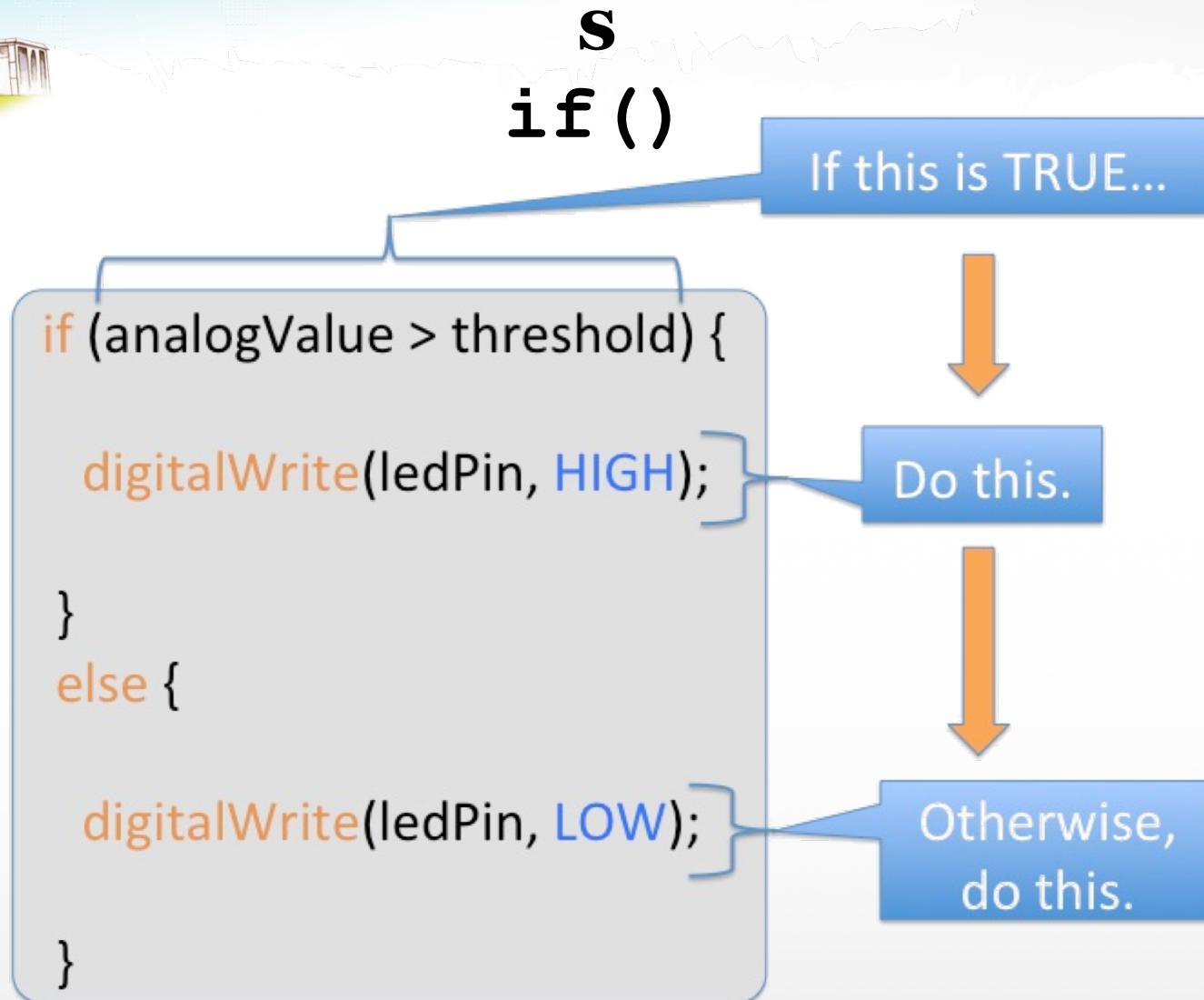
```
int buttonState = digitalRead(pushButton);
```

We name it `buttonState`

Recall that the `pushButton` variable stores the number 2

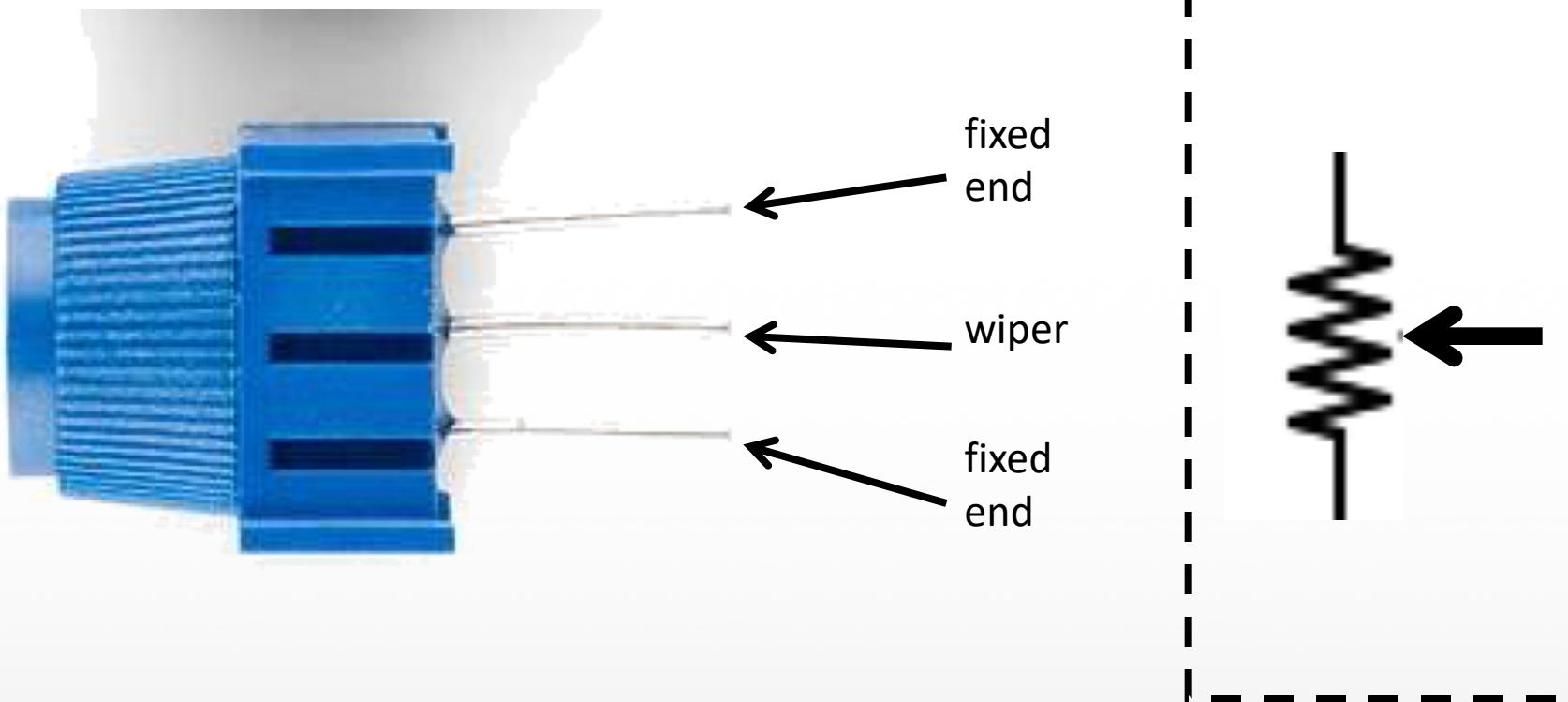
The value 1 or 0 will be saved in the variable `buttonState`.

Programming: Conditional Statement



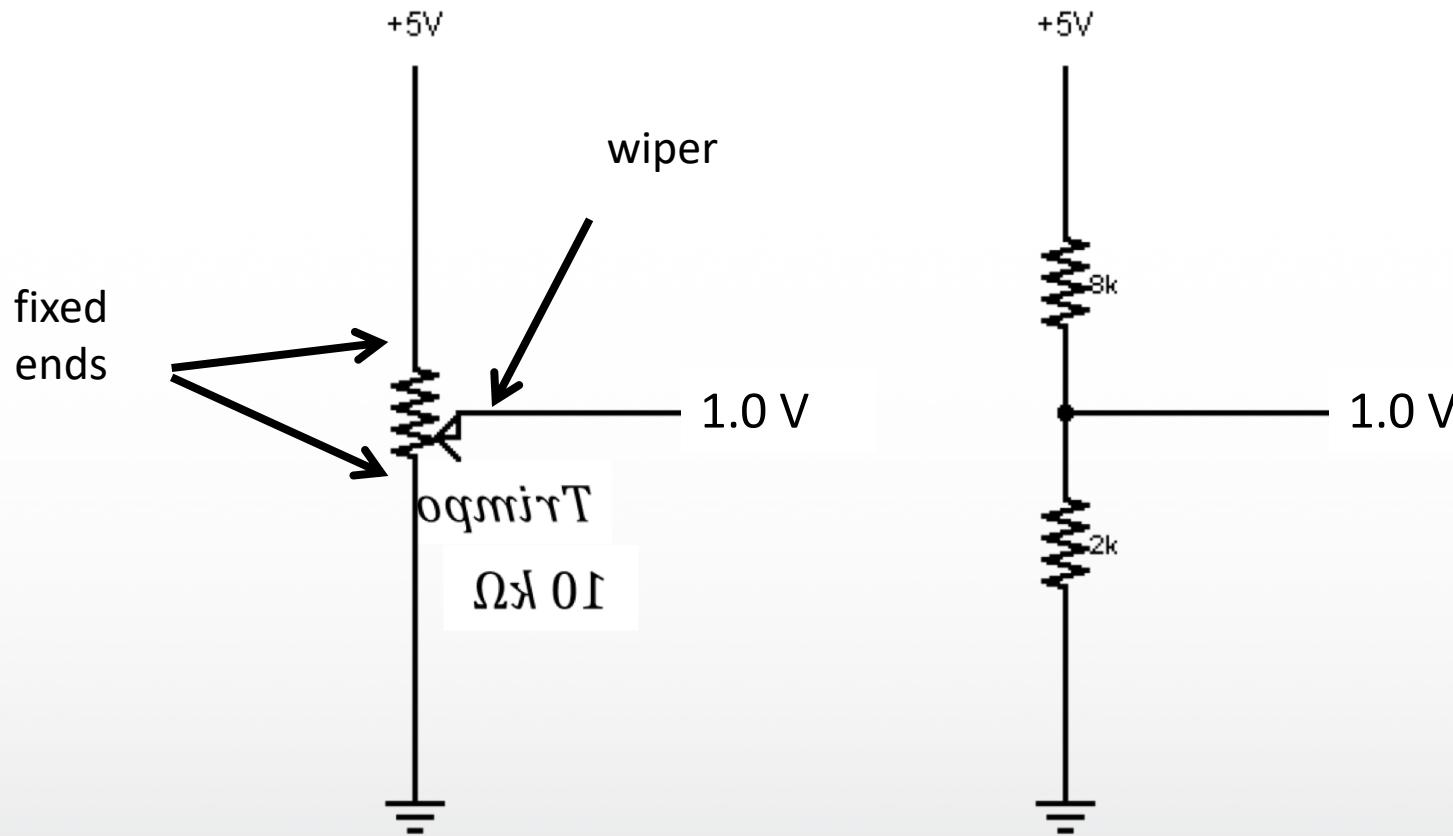


Trimpot (Potentiometer) Variable Resistor



Analog Sensors

3 Pin Potentiometer = var. resistor ([circuit](#))
a.k.a. *Voltage Divider Circuit*





Ohms Law... (just the basics)

Actually, this is the “voltage divider”

$$V_{R1} = V_{CC} \cdot \left(\frac{R_1}{R_{Total}} \right)$$

$$V_{R2} = V_{CC} \cdot \left(\frac{R_2}{R_{Total}} \right)$$

$$R_{Total} = R_1 + R_2$$

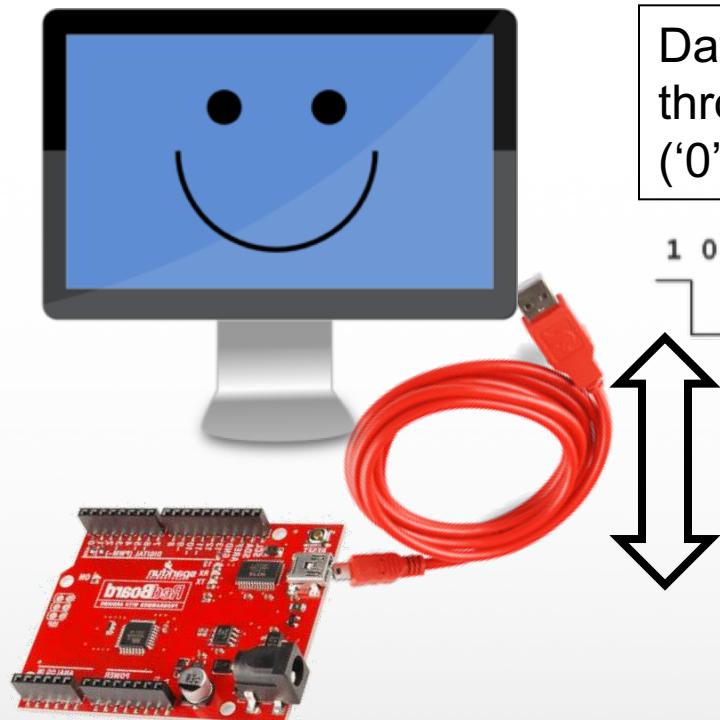


- Arduino uses a 10-bit A/D Converter:
- this means that you get input values from 0 to 1023
 - 0 V → 0
 - 5 V → 1023
- Ex:
 - `int sensorValue = analogRead(A0);`



Using Serial Communication

Method used to transfer data between two devices.



Data passes between the computer and Arduino through the USB cable. Data is transmitted as zeros ('0') and ones ('1') sequentially.



Arduino dedicates Digital I/O pin # 0 to receiving and Digital I/O pin #1 to transmit.

Serial Monitor & analogRead()



sketch_apr02a | Arduino 1.0.3

```
File Edit Sketch Tools Help
sketch_apr02a $ 
// analogRead() & Serial.print()
// 
// 

int sensorValue = 0;
int sensorPin = A0;

void setup()
{
    Serial.begin(9600); ←
    pinMode(A0, INPUT); ←
}

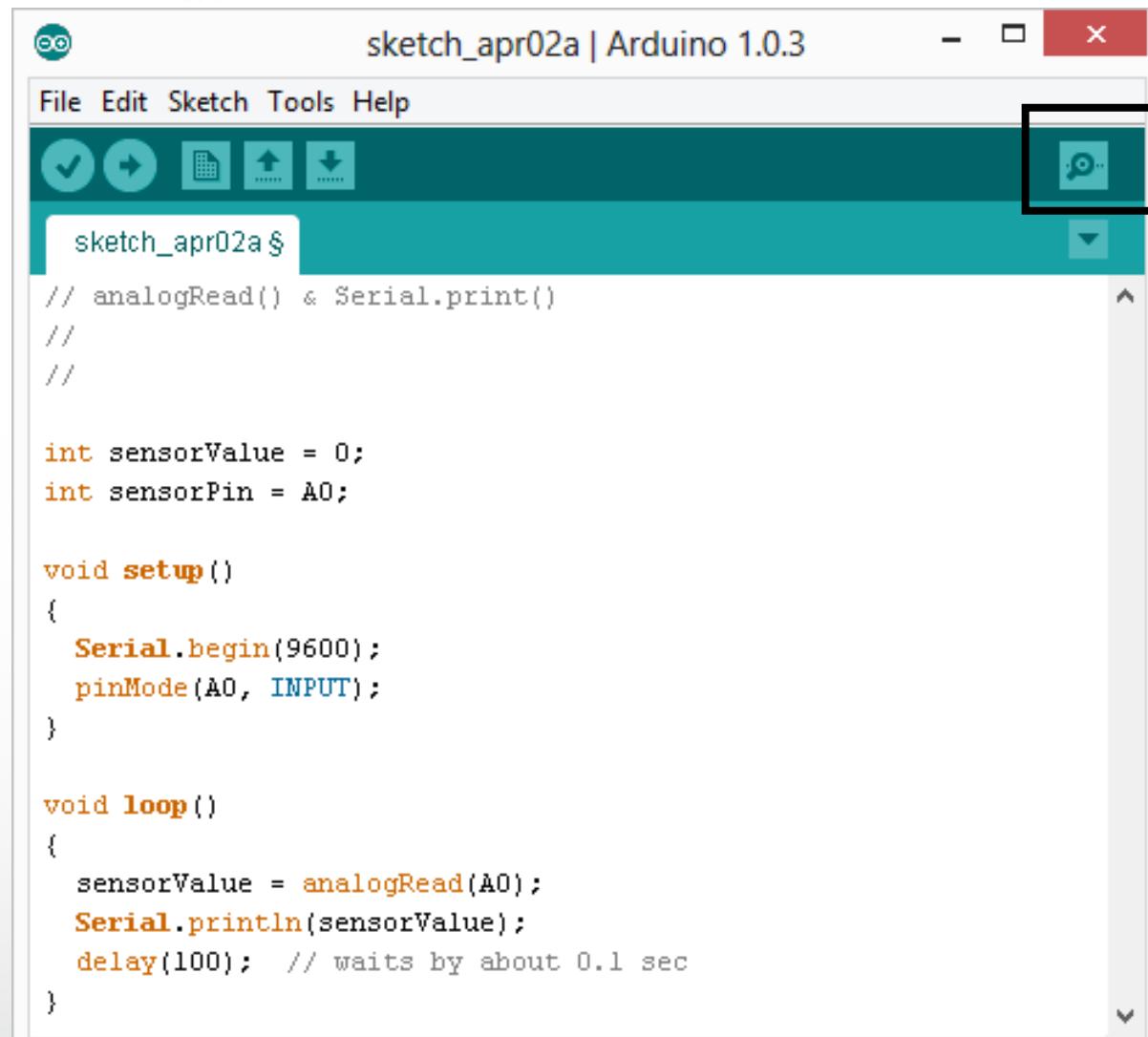
void loop()
{
    sensorValue = analogRead(A0);
    Serial.println(sensorValue); ←
    delay(100); // waits by about 0.1 sec
}
```

Initializes the Serial Communication

9600 baud data rate

prints data to serial bus

Serial Monitor & analogRead()



The screenshot shows the Arduino IDE interface with a sketch titled "sketch_apr02a". The code uses the `analogRead()` function to read an analog pin (A0) and prints the result to the Serial port. The IDE version is 1.0.3.

```
sketch_apr02a | Arduino 1.0.3

File Edit Sketch Tools Help

sketch_apr02a $ // analogRead() & Serial.print()
// 
// 

int sensorValue = 0;
int sensorPin = A0;

void setup()
{
    Serial.begin(9600);
    pinMode(A0, INPUT);
}

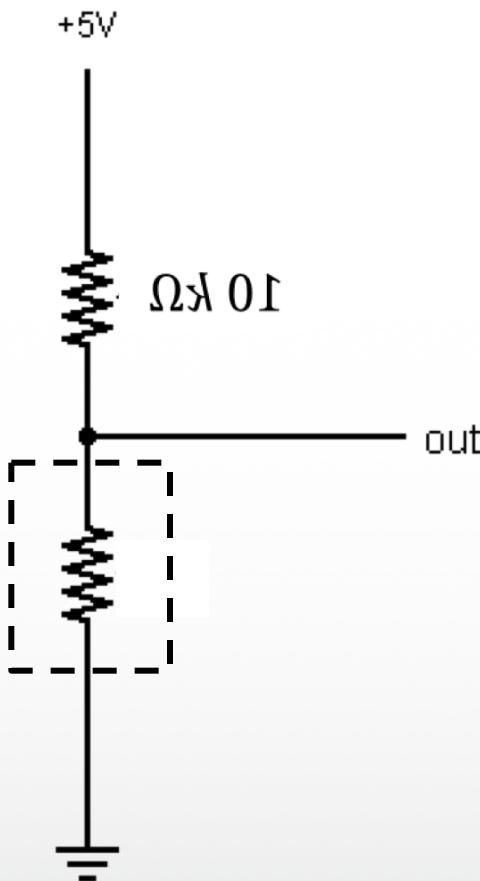
void loop()
{
    sensorValue = analogRead(A0);
    Serial.println(sensorValue);
    delay(100); // waits by about 0.1 sec
}
```



Opens up a Serial Terminal Window

Analog Sensors

2 Pin Analog Sensors = var. resistor



- Take two sensors -- Use the Serial Monitor and find the range of input values you get for each sensor.

- `MaxAnalogRead = _____`

- `MinAnalogRead = _____`



Analog Sensors

Examples:

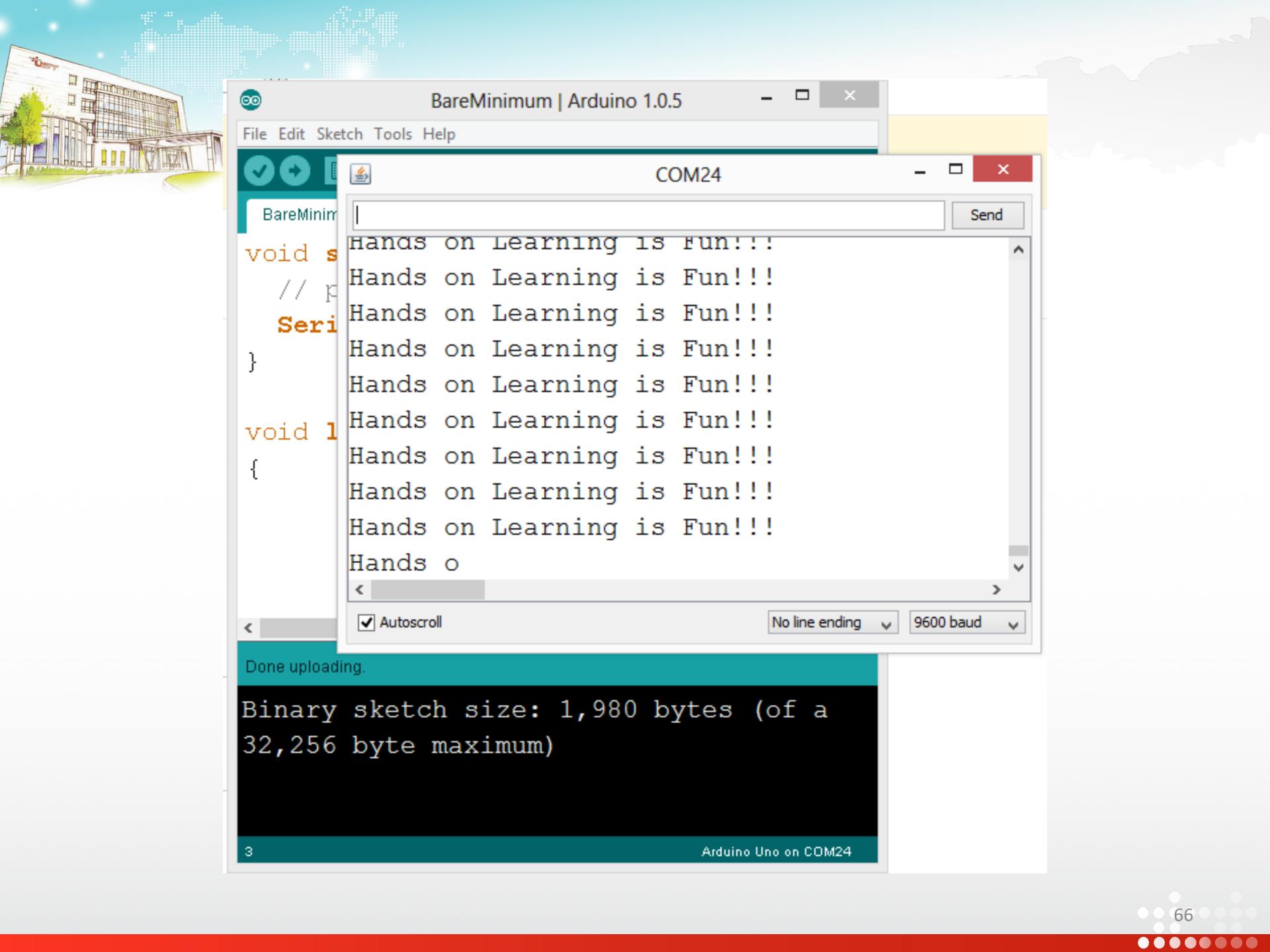
Sensors	Variables
Mic	soundVolume
Photoresistor	lightLevel
Potentiometer	dialPosition
Temp Sensor	temperature
Flex Sensor	bend
Accelerometer	tilt/acceleration



Additional Serial Communication

Sending a Message

```
void loop ( )  
{  
    Serial.print("Hands on ") ;  
    Serial.print("Learning ") ;  
    •   Serial.println("is Fun!!!!") ;  
}  
}
```



BareMinimum | Arduino 1.0.5

File Edit Sketch Tools Help

Serial Monitor

COM24

Send

```
Hands on Learning is fun!!!
Hands o
```

< >

Autoscroll

No line ending 9600 baud

Done uploading.

Binary sketch size: 1,980 bytes (of a 32,256 byte maximum)

3 Arduino Uno on COM24

The screenshot shows the Arduino IDE interface with a sketch named "BareMinimum". The code contains two functions: setup() and loop(). The setup() function initializes the serial communication at 9600 baud. The loop() function sends the string "Hands on Learning is fun!!!". The serial monitor window shows the output of the sketch being printed repeatedly. A message at the bottom indicates the upload was successful, showing the binary sketch size as 1,980 bytes out of a maximum of 32,256 bytes.