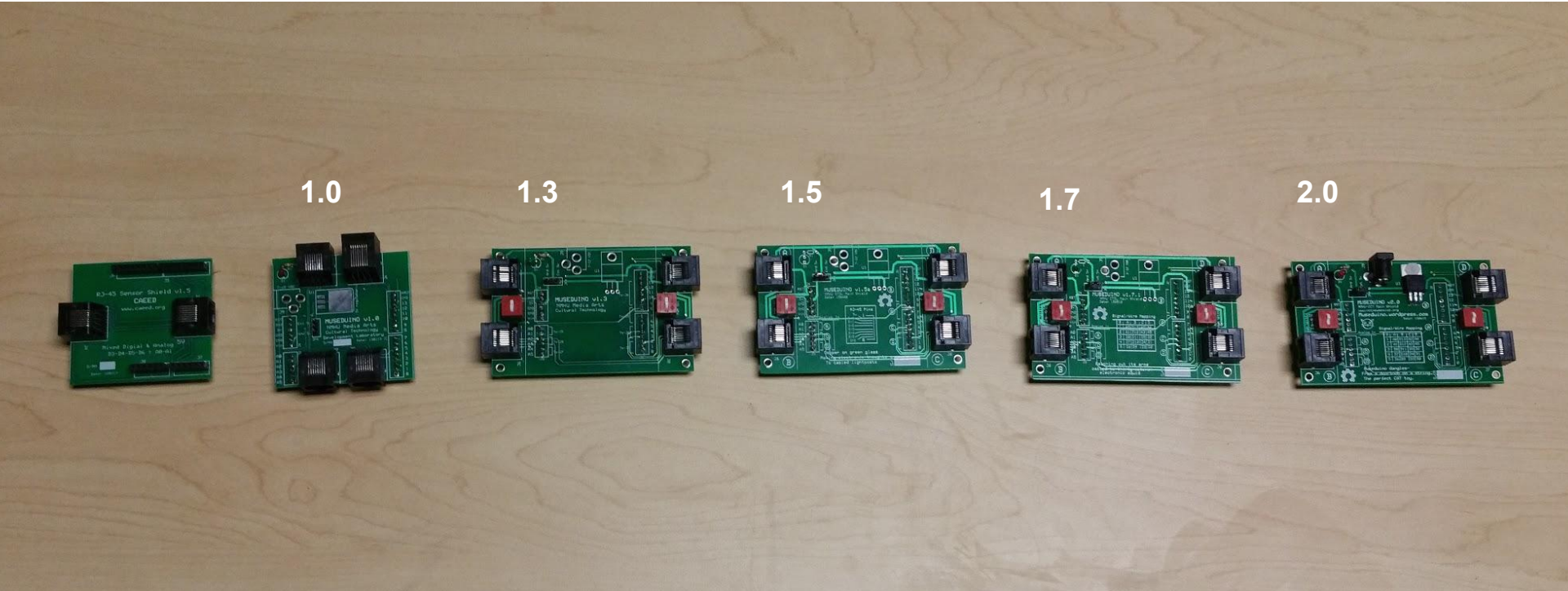
an open source electronics environment for exhibits

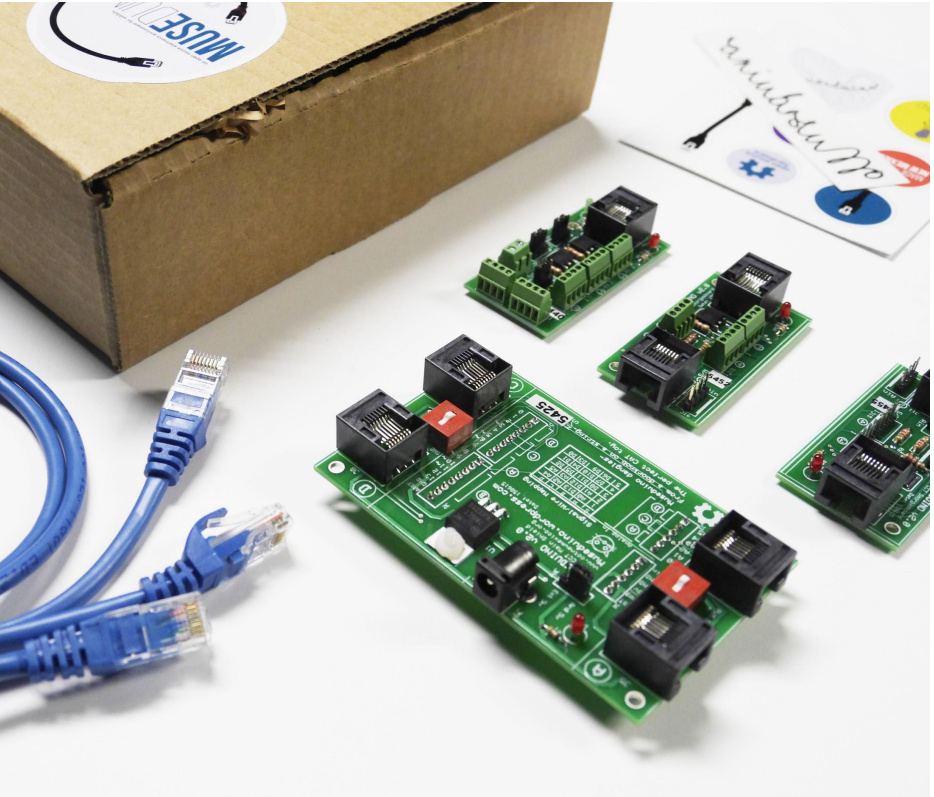
# MUSEDUINO

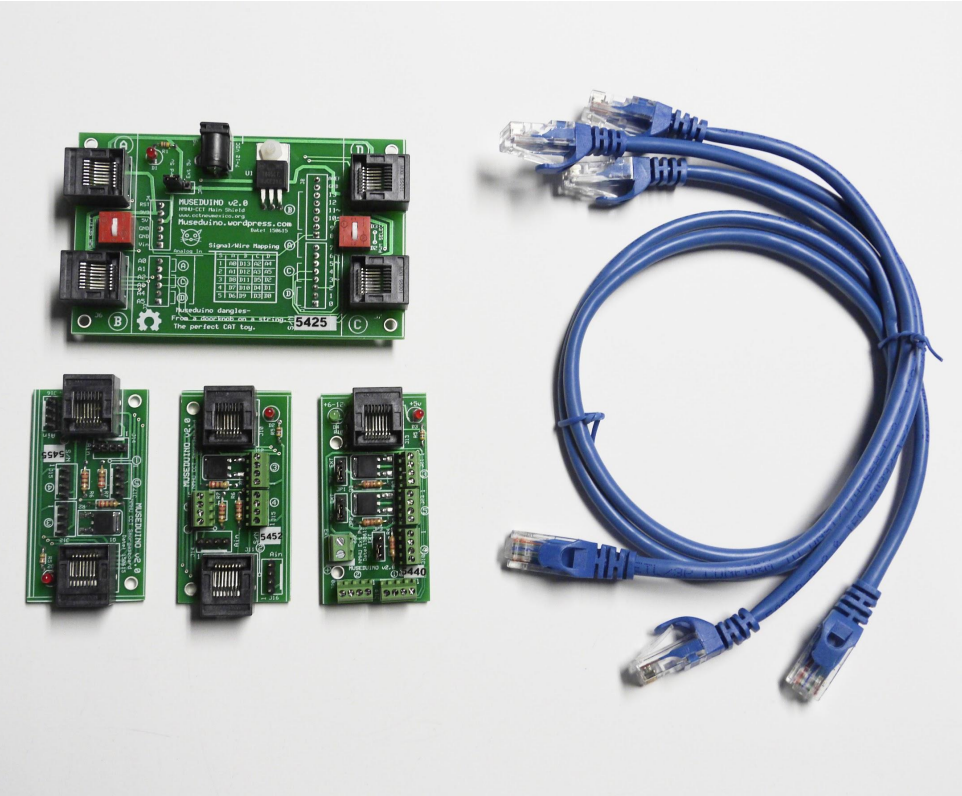
open source hardware

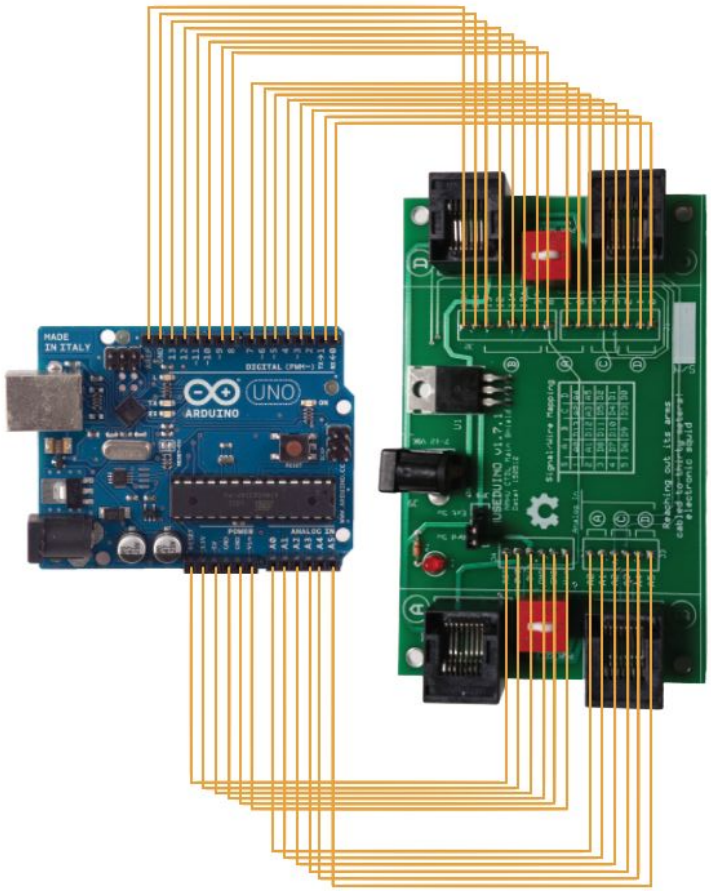
# History of Museduino



# Arduino + Museduino



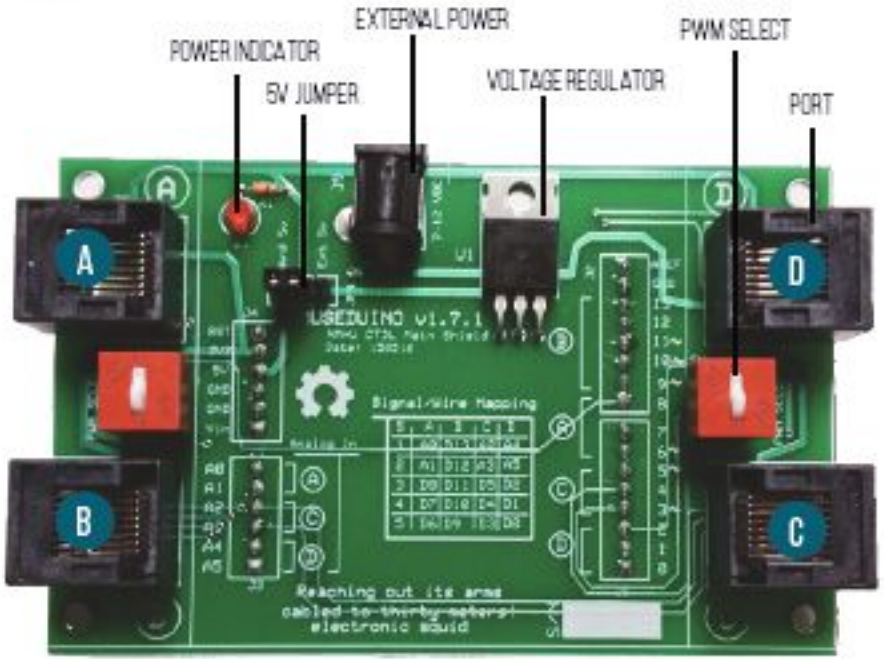





| PORTS | A | B | C | D |
|---|---|---|---|---|
| SATELITTE I/O 1 | A0 | D13 | A2 | A4 |
| 2 | A1 | D12 | A3 | A5 |
| 3 | *D8 | D11~ | D5~ | *D2 |
| 4 | D7 | D10~ | D4 | D1 |
| 5 | D6~ | *D9~ | *D3~ | D0 |

~ PWM  * PWM SELECT

## SHIELD




## 5V JUMPER

5V EXTERNAL POWER

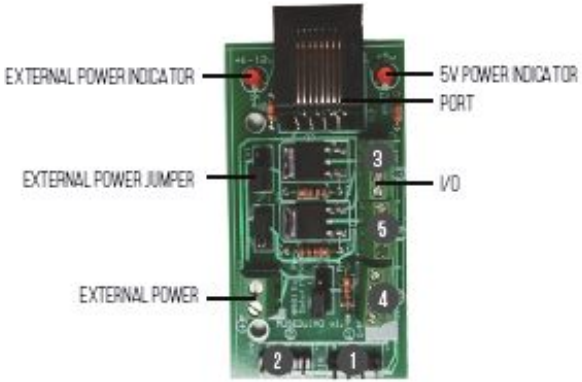
5V ARDUINO POWER

## * PWM SELECT

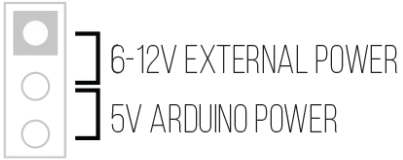
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | 3 | D8 | D9~ | D | 3 | D2 | D3~ |
| ↑ | | ↑ | ↓ | ↑ | | ↑ | ↓ |
| B | 5 | D9~ | D8 | C | 5 | D3~ | D2 |

## SMORGASBOARD


## EXTERNAL POWER BOARD



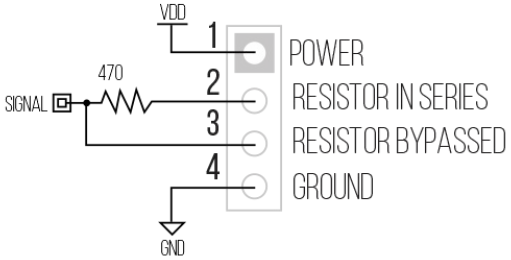

## EXTERNAL POWER JUMPER




## PIN CONFIGURATION

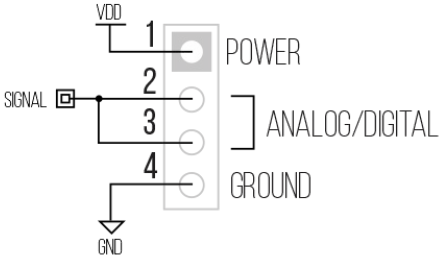
### SMORGASBOARD

4 5




### SMORGASBOARD/ EXTERNAL POWER BOARD

1 2

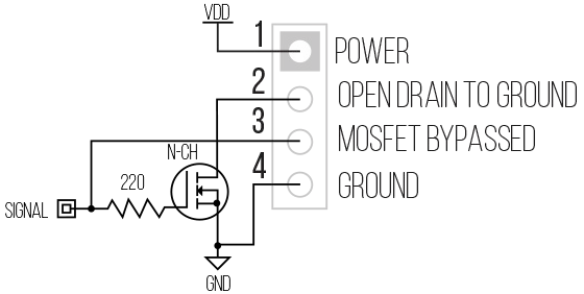



### SMORGASBOARD

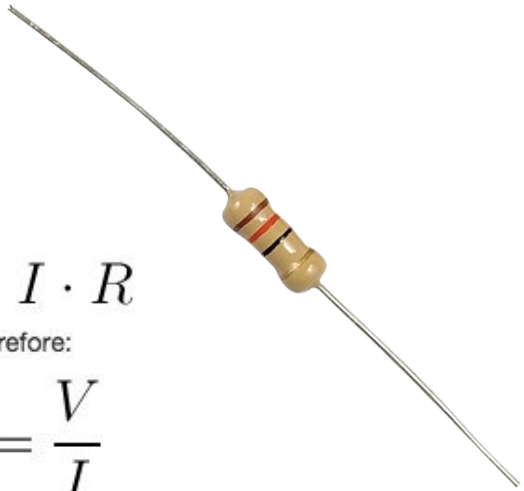
3

### EXTERNAL POWER BOARD

3 4 5



# LED (LIGHT EMITTING DIODE) & Resistors

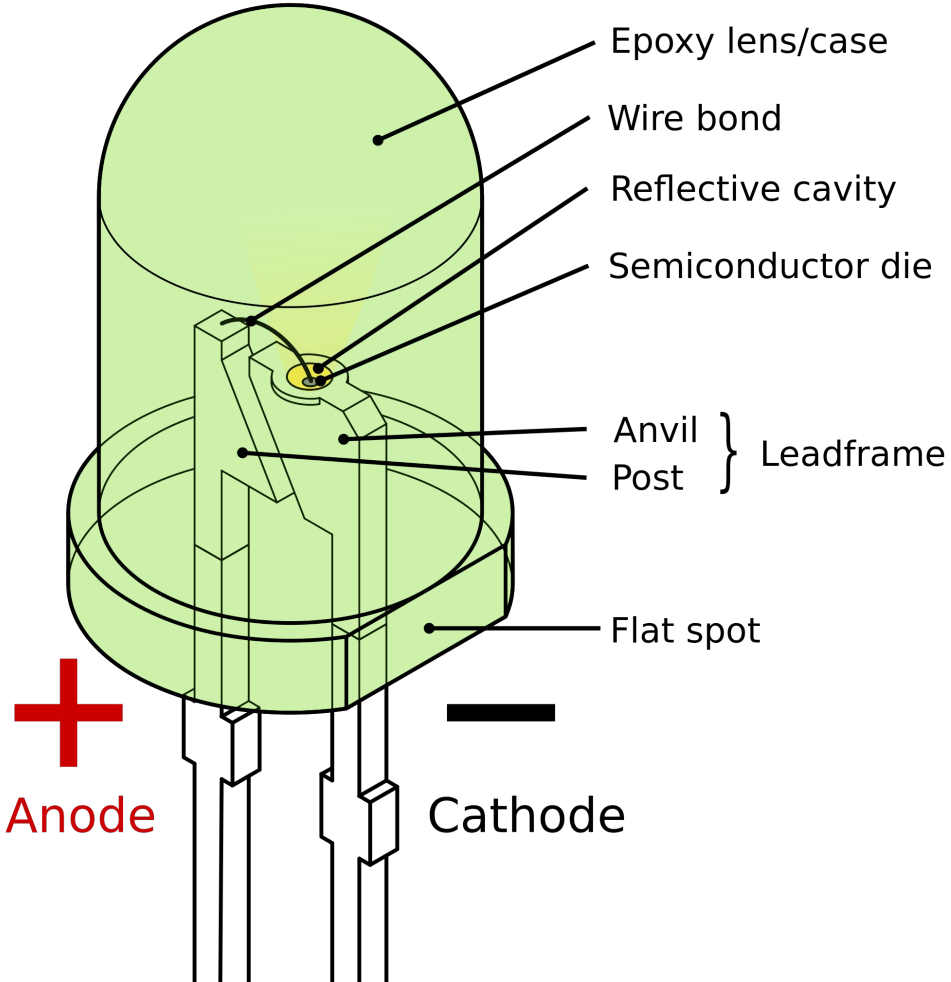


$$V = I \cdot R$$

therefore:

$$R = \frac{V}{I}$$

Resistors are used to limit the value of current.



# Blink Syntax

Set variable

```
// Digital Pin 6 on Satellite Pin 5 via Port A
int s5A = 6;
```

pinMode(pin,mode)

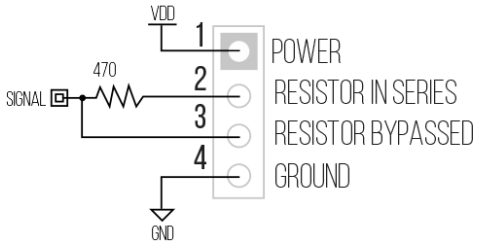
```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(s5A, OUTPUT);
}
```

digitalWrite(pin, value)

```
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(s5A, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(s5A, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

# Blink with Museduino

5 A D6~

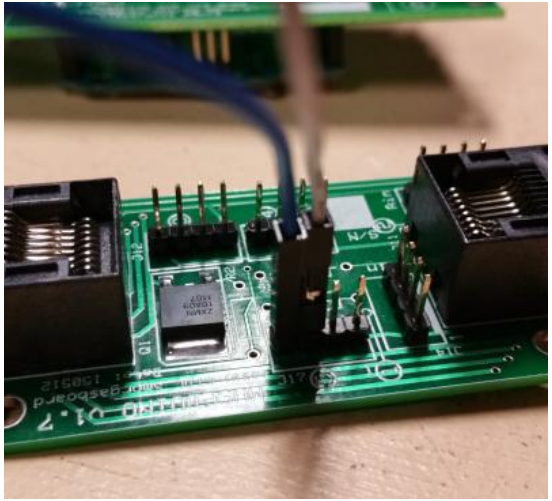


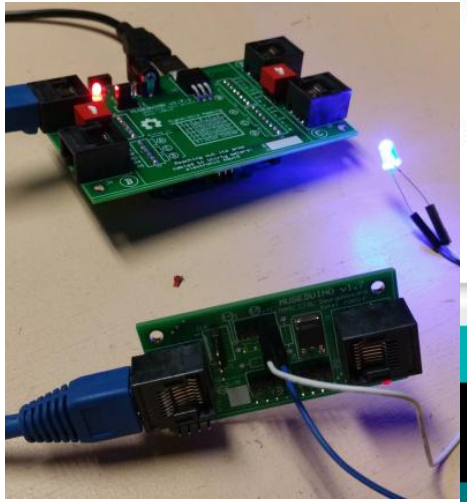

led | Arduino 1.0.5

led

```
/*
  Museduino | LED Tutorial
  Turns on LED for one second, then off for one second, repeatedly.
*/

// Digital Pin 6 on Satellite Pin 5 via Port A
int s5A = 6;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(s5A, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(s5A, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(s5A, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

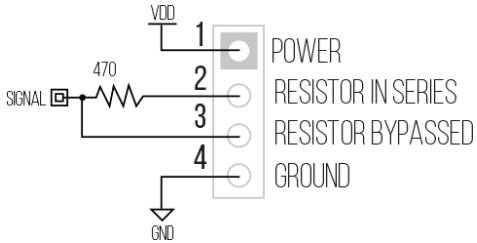
1 Arduino Uno on /dev/tty.usbmodem1411

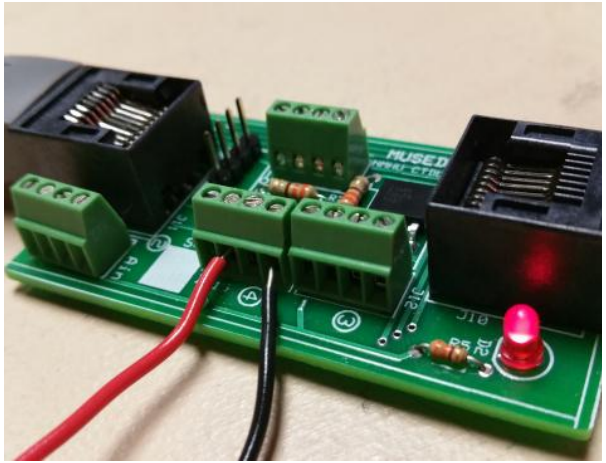


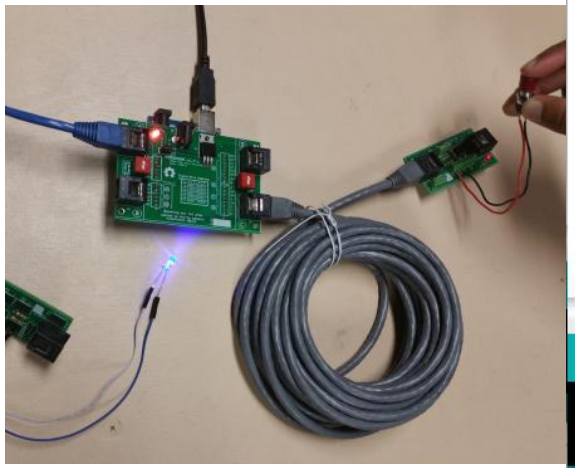


# Button with Museduino

4 C D4





button | Arduino 1.0.5

button

```
// Digital Pin 6 on Satellite Pin 5 via Port A
int s5A = 6;
// Digital Pin 4 on Satellite Pin 4 via Port C
int s4C = 4;

// variables
int buttonState; // variable for reading the button status

// the setup routine runs once when you press reset:
void setup() {
  // initialize the led pin as an output.
  pinMode(s5A, OUTPUT);
  // initialize the button pin as an output.
  pinMode(s4C, INPUT_PULLUP);

}

// the loop routine runs over and over again forever:
void loop() {

  // read the state of the button value:
  buttonState = digitalRead(s4C);

  // check if the button is pushed
  // if it is, the buttonState is LOW:
  if (buttonState == LOW) {
    // turn LED on:
    digitalWrite(s5A, HIGH);
  }
  else {
      // turn LED off:
      digitalWrite(s5A, LOW);
  }

}
```

10 Arduino Uno on /dev/tty.usbmodem1411

# Button Syntax

digitalRead(inPin)

```
if (condition is met) {

    //action A

}

else {

    //action B

}
```

```
// the loop routine runs over and over again forever:
void loop() {

  // read the state of the button value:
  buttonState = digitalRead(s4C);

  // check if the button is pushed
  // if it is, the buttonState is LOW:
  if (buttonState == LOW) {
    // turn LED on:
    digitalWrite(s5A, HIGH);
  }
  else {
      // turn LED off:
      digitalWrite(s5A, LOW);
  }

}
```

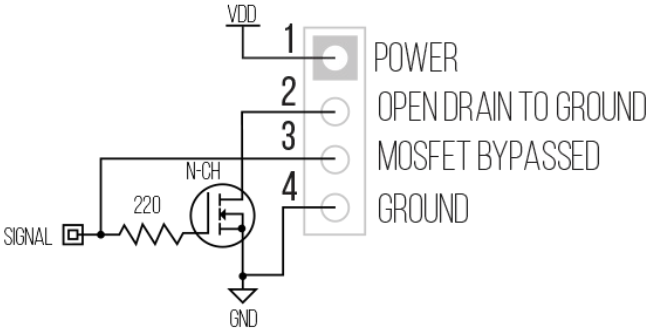
# Servo with Museduino

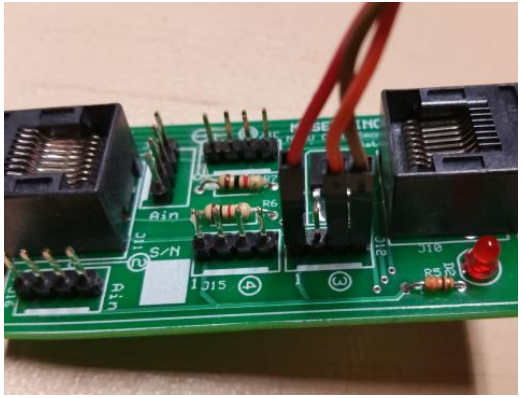
3 D *D3~

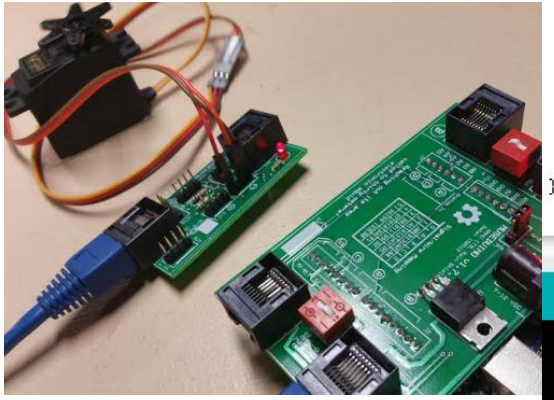
## * PWM SELECT

D 3 D3~

C 5 D2





servo | Arduino 1.0.5

servo

```
/*
  Museduino | Servo Motors (PWM Select) Tutorial
  Servo Motor repeatedly turns from 0 to 180 degrees, then 180 to 0 degrees.
*/

#include <Servo.h>

Servo servo;  // create servo object to control a servo
              // a maximum of eight servo objects can be created

int pos = 0;    // variable to store the servo position

//Use PWM Select switch to swap default pin D2 with D3~
int s3D = 3; // Digital Pin 3 on Satellite Pin 3 via Port D

void setup()
{
  servo.attach(s3D);  // attaches the servo on pin 3 to the servo object
}


void loop()
{
  for(pos = 0; pos < 180; pos += 1)  // 0 degrees to 180 degrees
  {                                  // in steps of 1 degree
    servo.write(pos);                // tell servo to go to position
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos>=1; pos-=1)     // 180 degrees to 0 degrees
  {
    servo.write(pos);                // tell servo to go to position
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

1 Arduino Uno on /dev/tty.usbmodem1411

# Analog Syntax

Initialize serial communication

analogRead(pin) = int(0 to 1023)

print to serial

```
int sensor = A0;
int value; //variable for storing value of sensor

// the setup routine runs once when you press reset:
void setup() {
  // begin serial communication
  Serial.begin(9600);
  // initialize the digital pin as an output.
  pinMode(sensor, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {

  // reads the value of the sensor (value between 0 and 1023)
  value = analogRead(sensor);
  Serial.println(value);

  // map value to range (value between 0 and 255)
  value = map(value, 0, 1023, 0, 255);
  Serial.println(value);

  delay(15);  //wait 15ms for next reading

}
```

# Where we source our electronics:

adafruit

sparkfun ELECTRONICS

Digi-Key ELECTRONICS

JAMECO ELECTRONICS

expresspcb

# Contact

---

info.ctdl@gmail.com

Rianne | rtruji89@gmail.com

Miles | mtokunow@gmail.com

Museduino.wordpress.com

#museduino