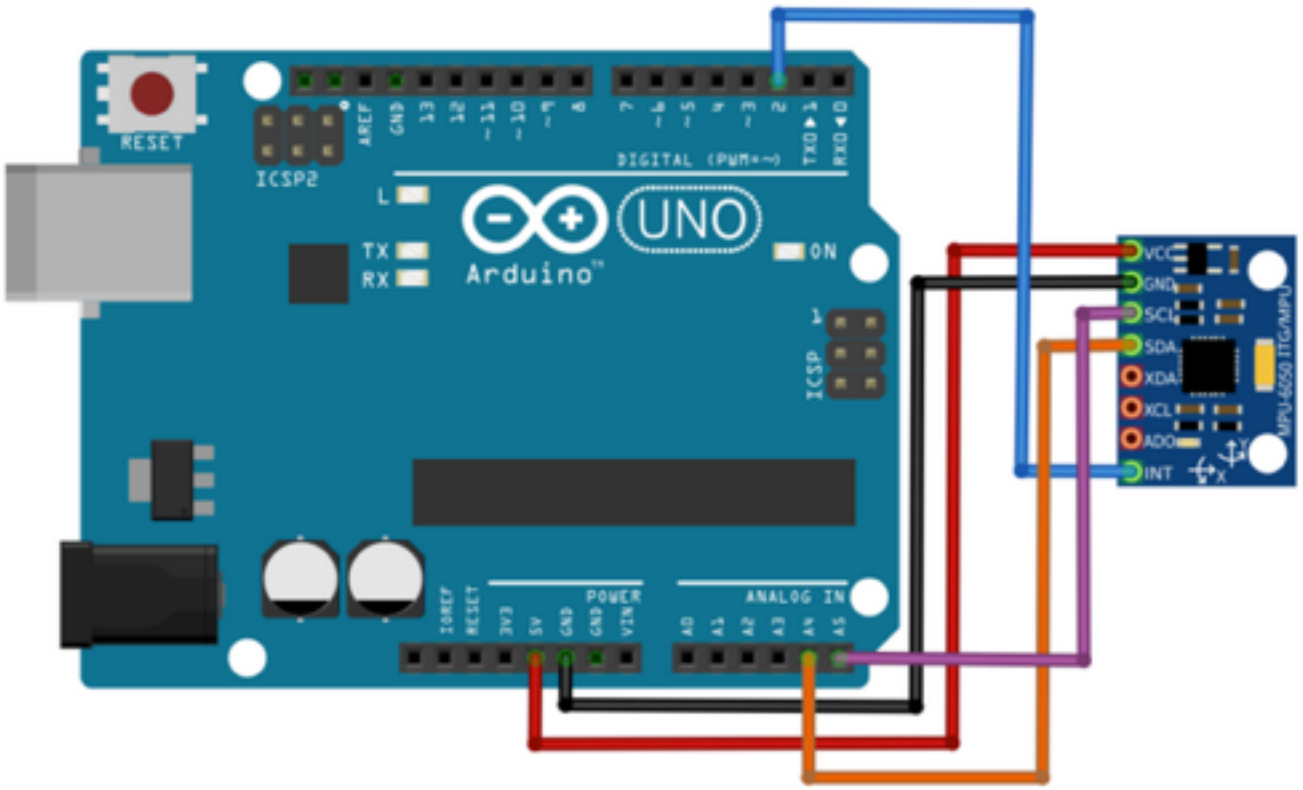
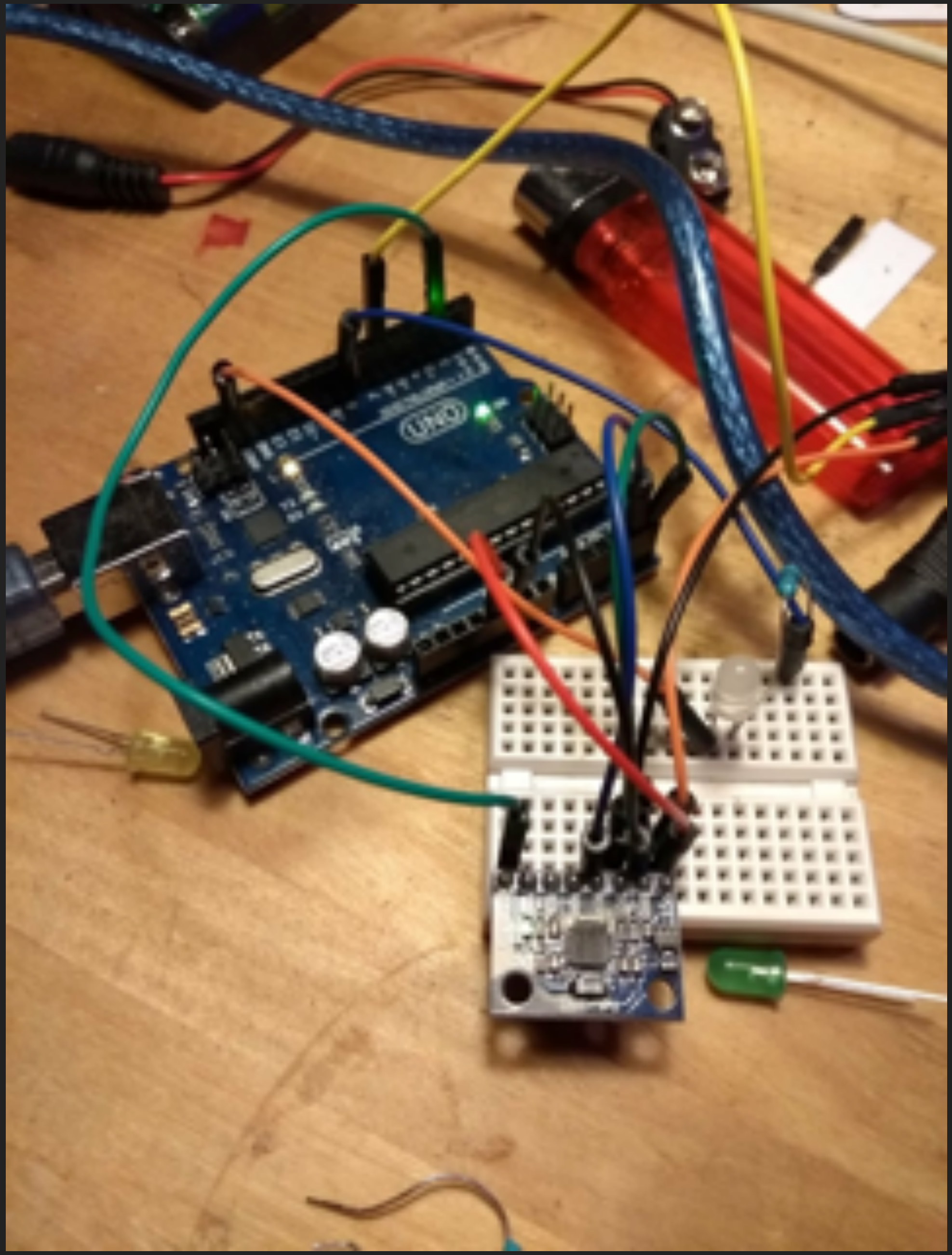
ONE DIMENSIONAL ARCADE GAME

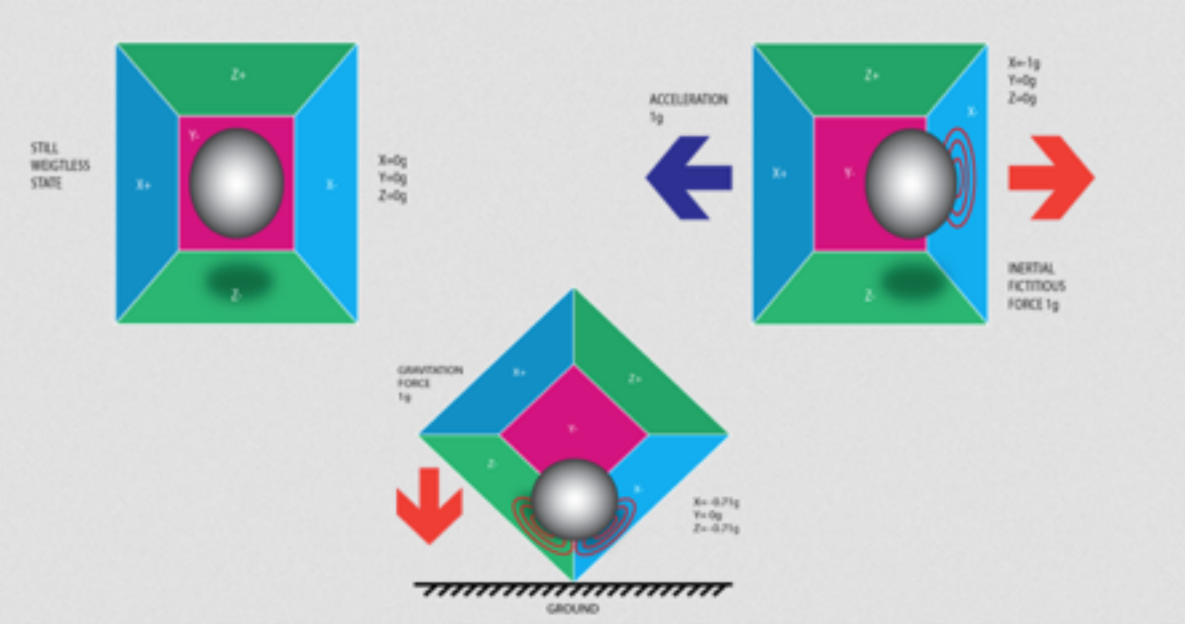
WOOBLI

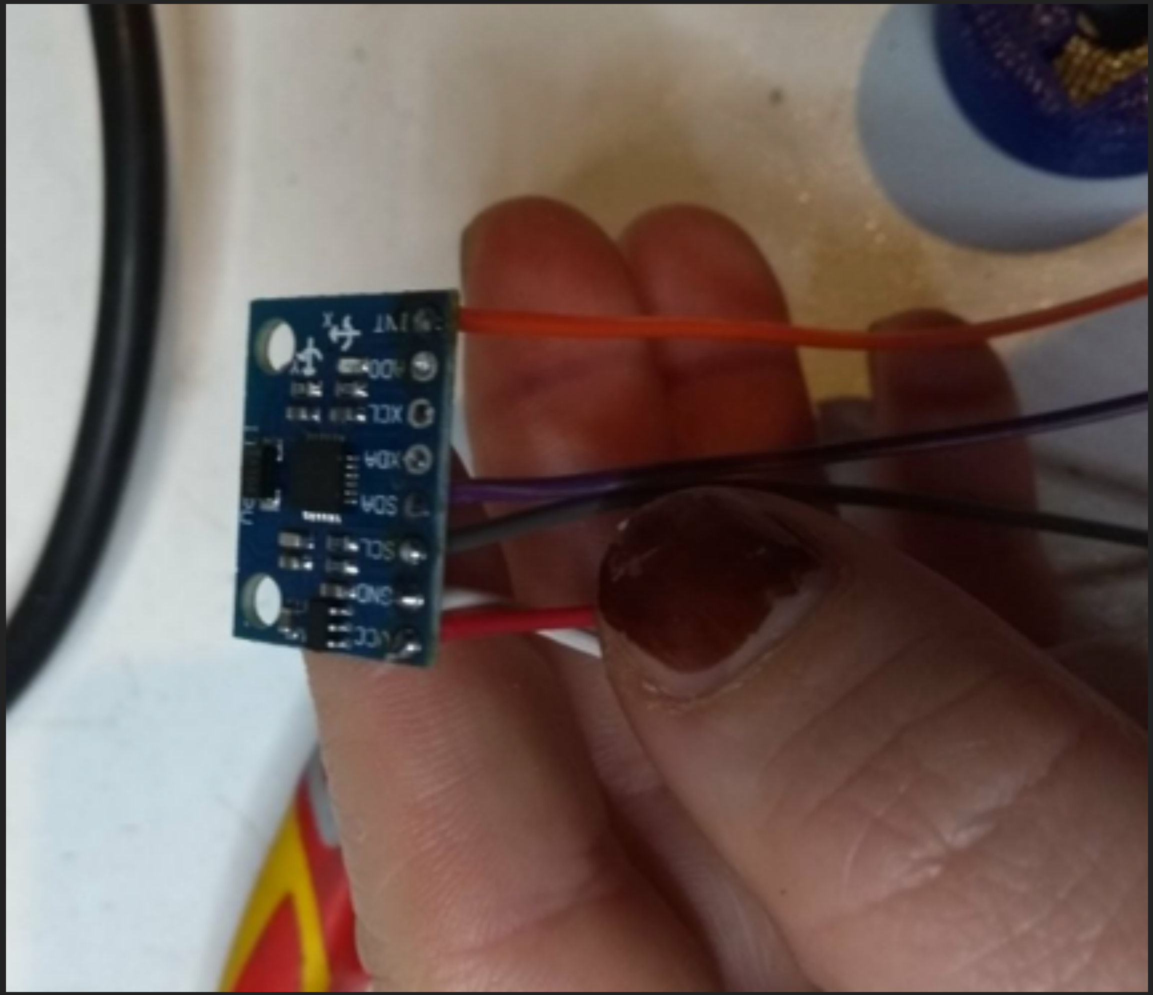
MPU6050



Arduino MPU 6050 connections



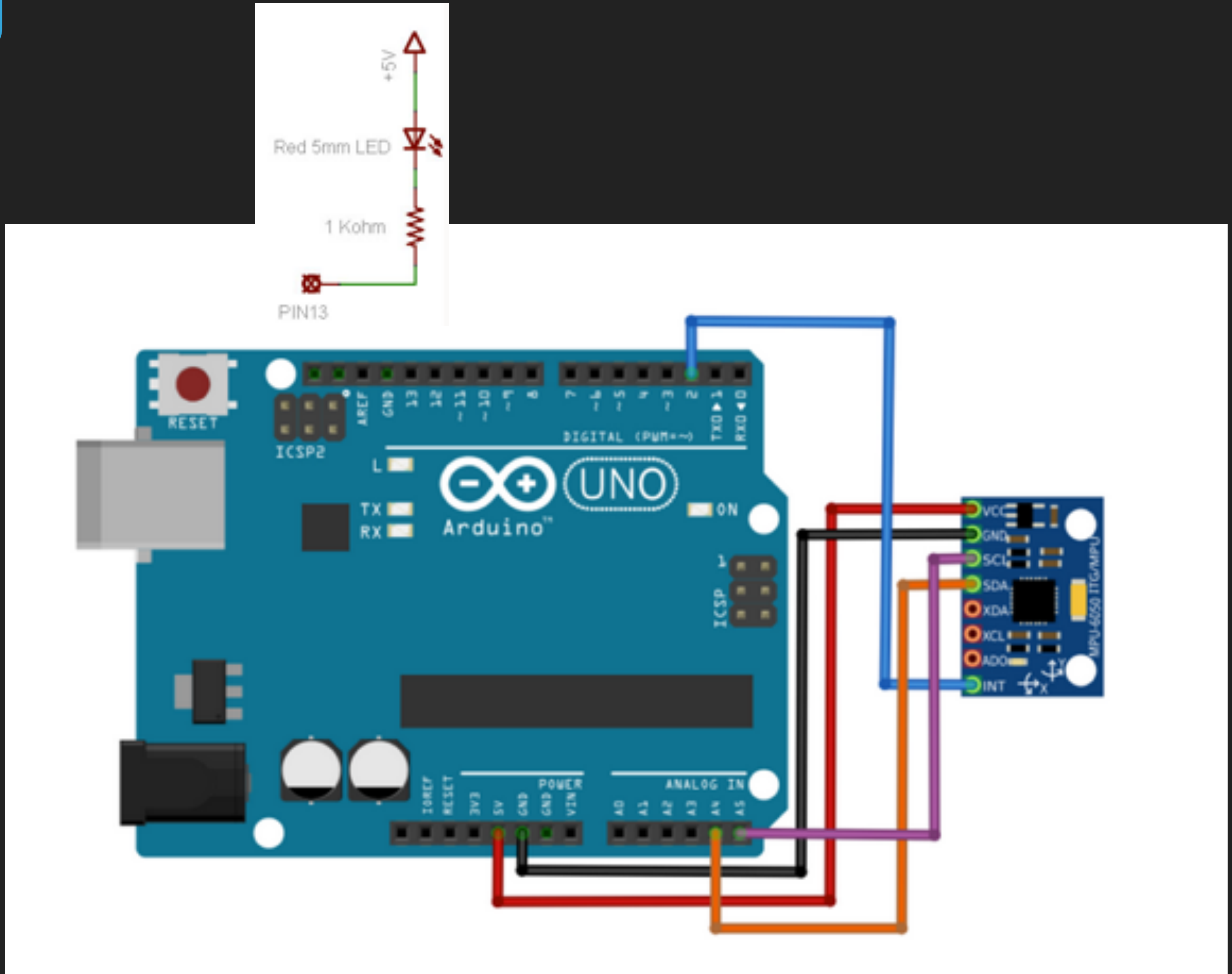

STILL WEIGTLESS STATE
Z+
Y-
X+
X-
Z-
X=0g
Y=0g
Z=0g
ACCELERATION 1g
Z+
X+
Y-
X-
Z-
X=-1g
Y=0g
Z=0g
INERTIAL FICTITIOUS FORCE 1g
GRAVITATION FORCE 1g
X+
Z+
Y-
Z-
X-
X= -0.71g
Y= 0g
Z= -0.71g
GROUND


INT
AD0
XCL
XDA
SDA
SCL
GND
VCC

tesa

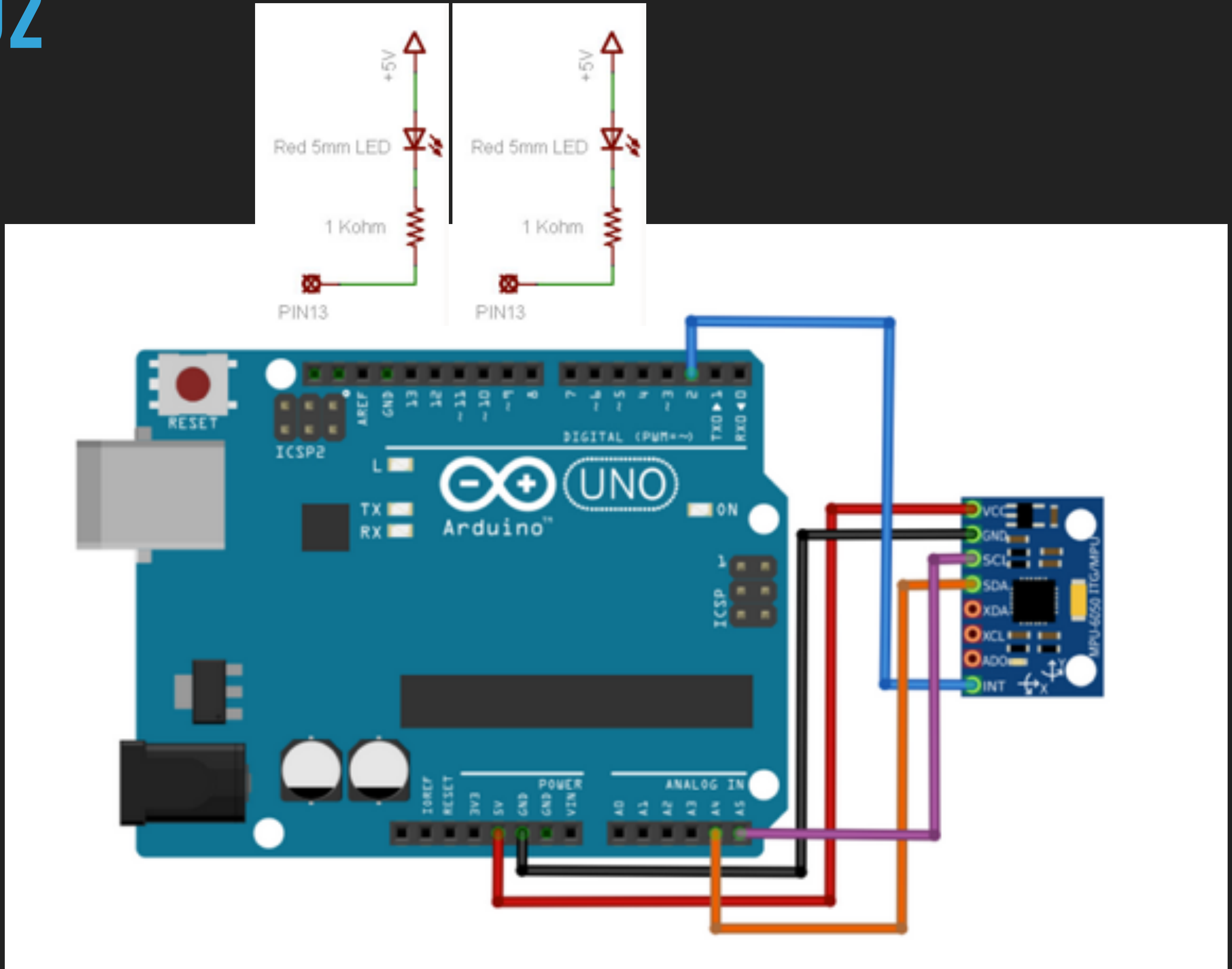
Trink

LED



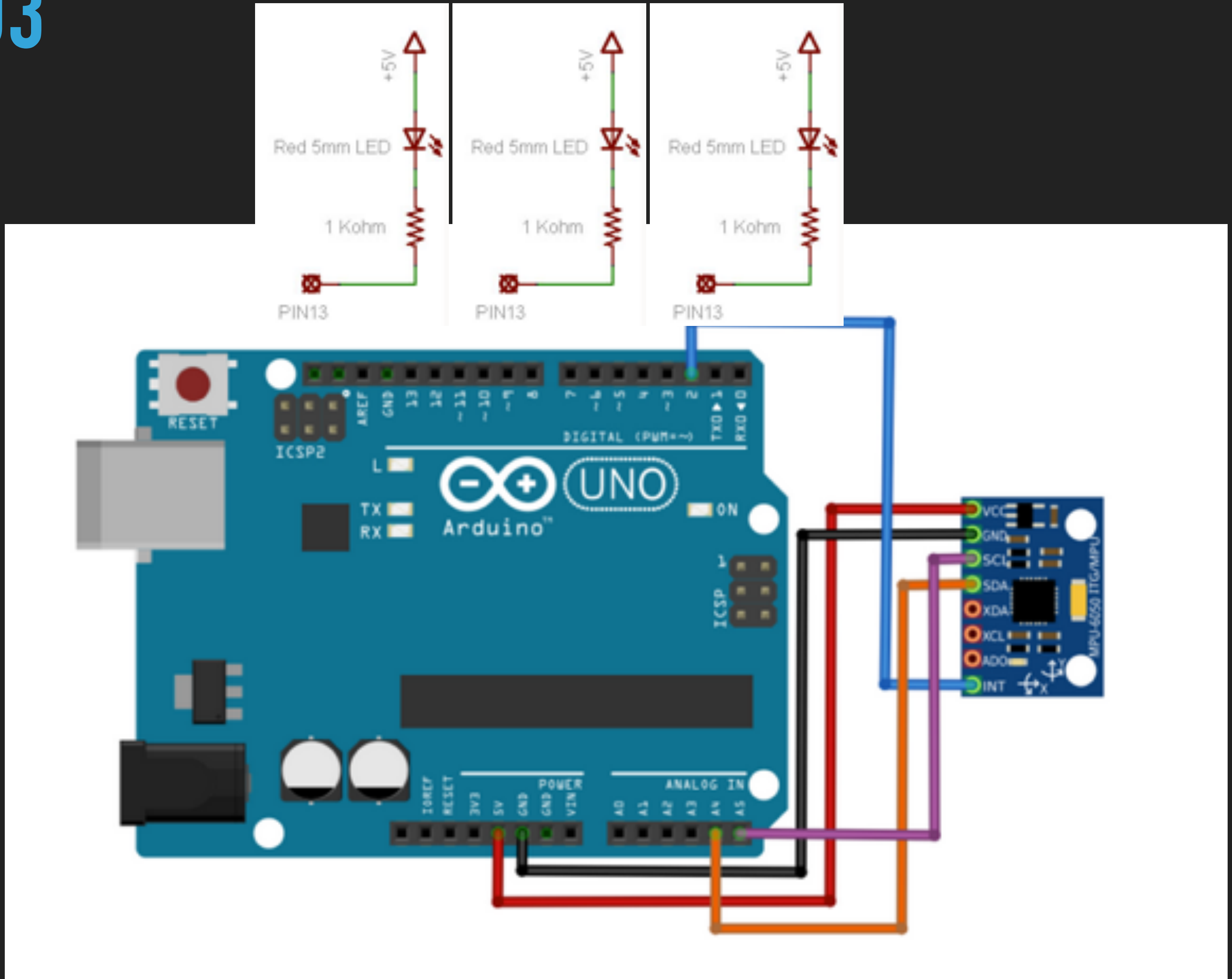
Arduino MPU 6050 connections

LED2

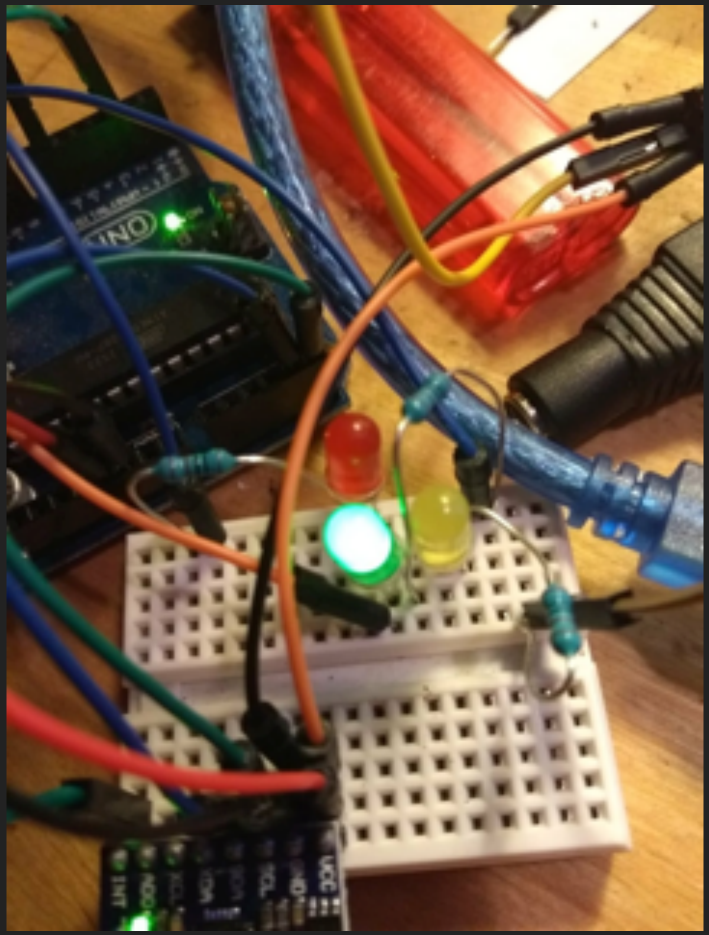


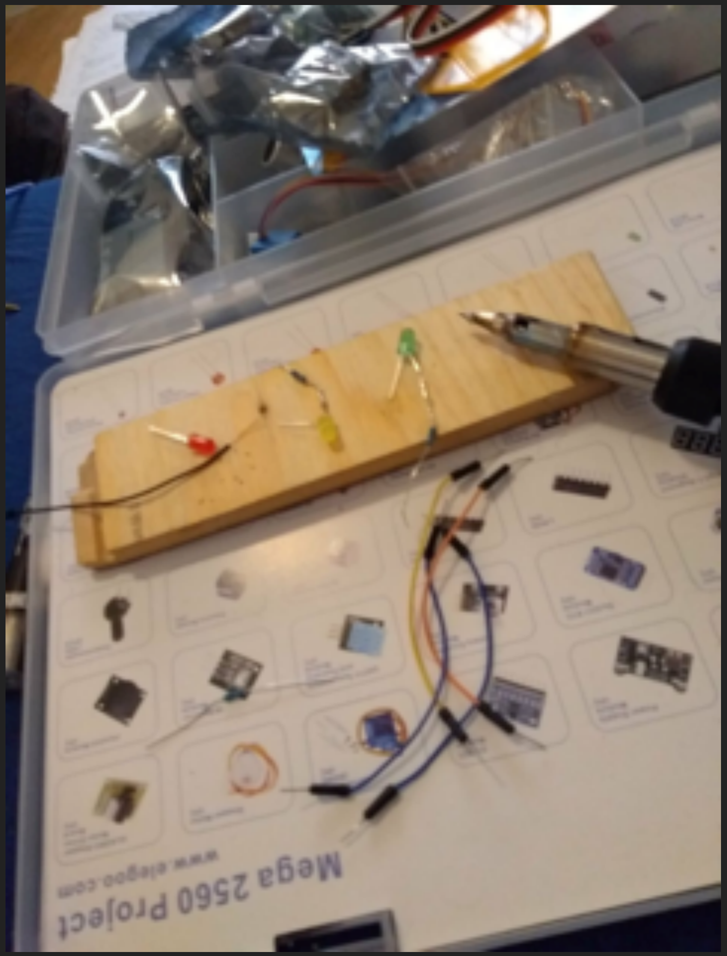
Arduino MPU 6050 connections

LED3

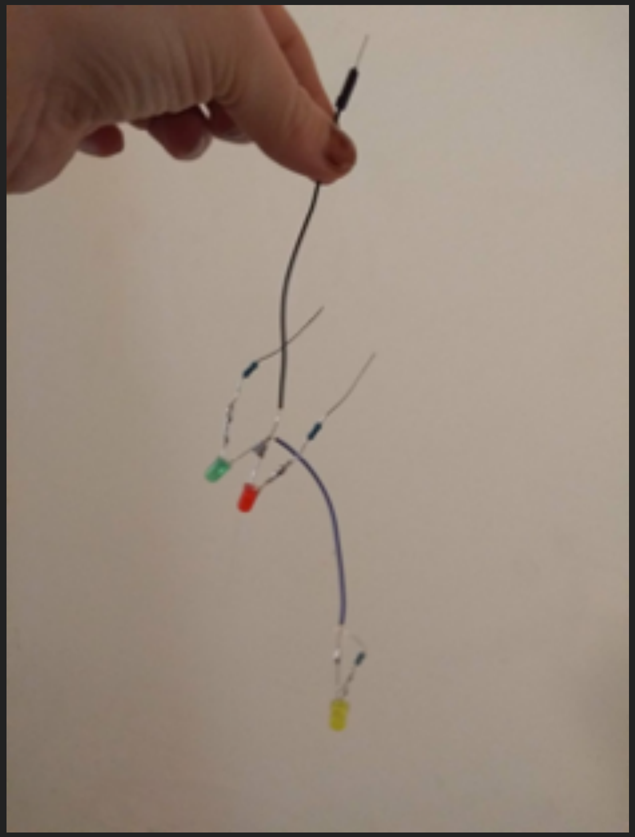


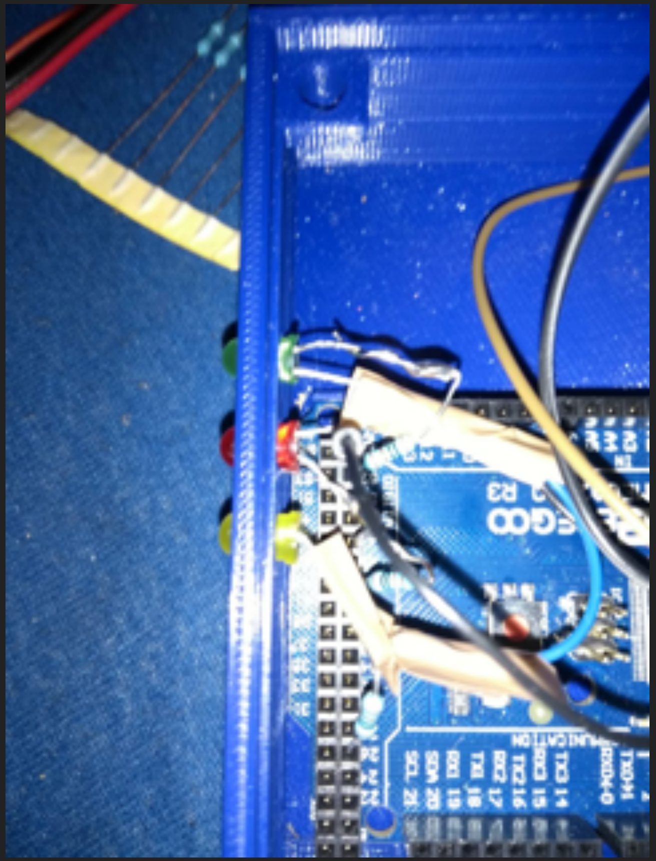
Arduino MPU 6050 connections



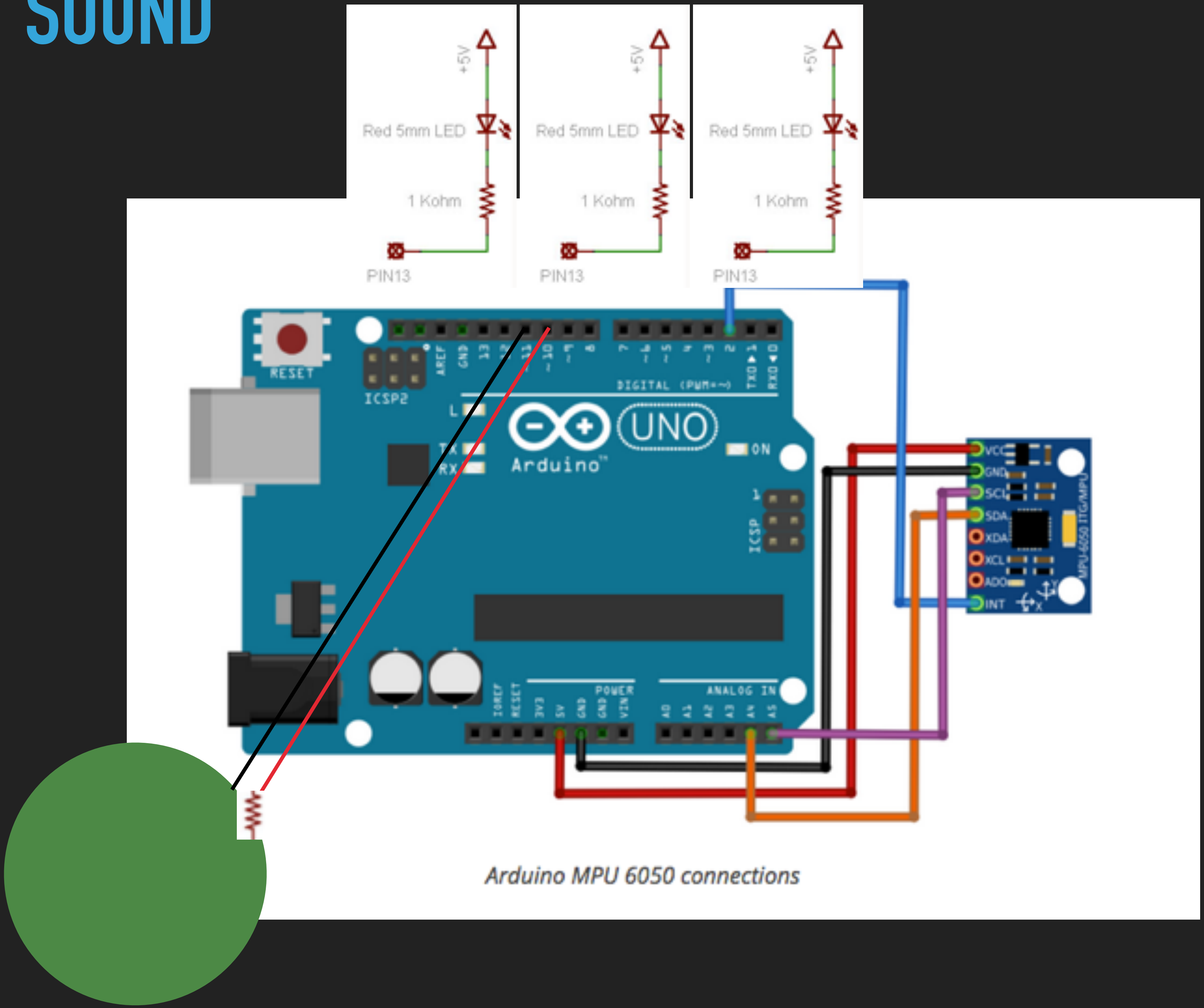


Mega 2560 Project
www.elegoo.com



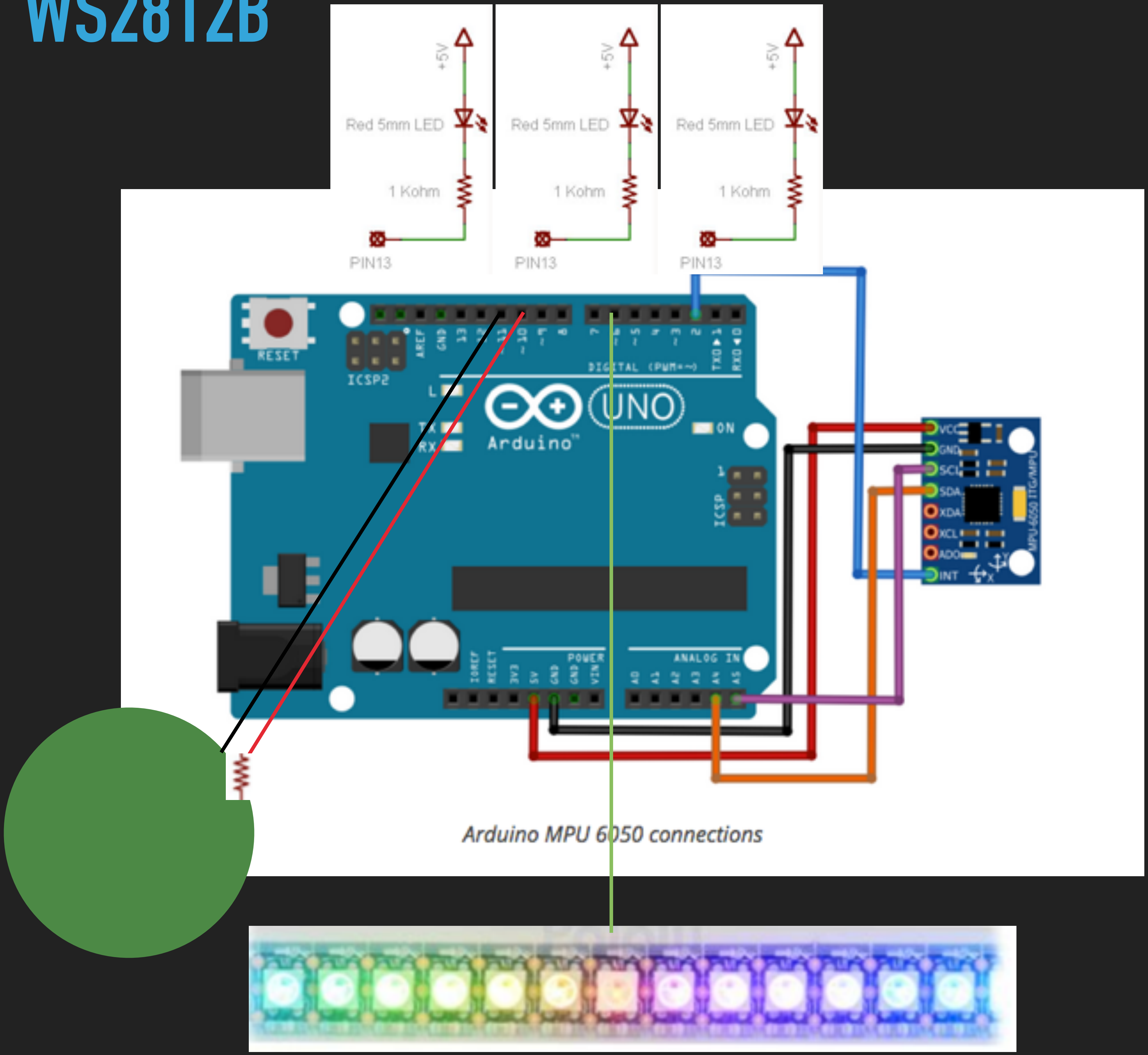


SOUND



Arduino MPU 6050 connections

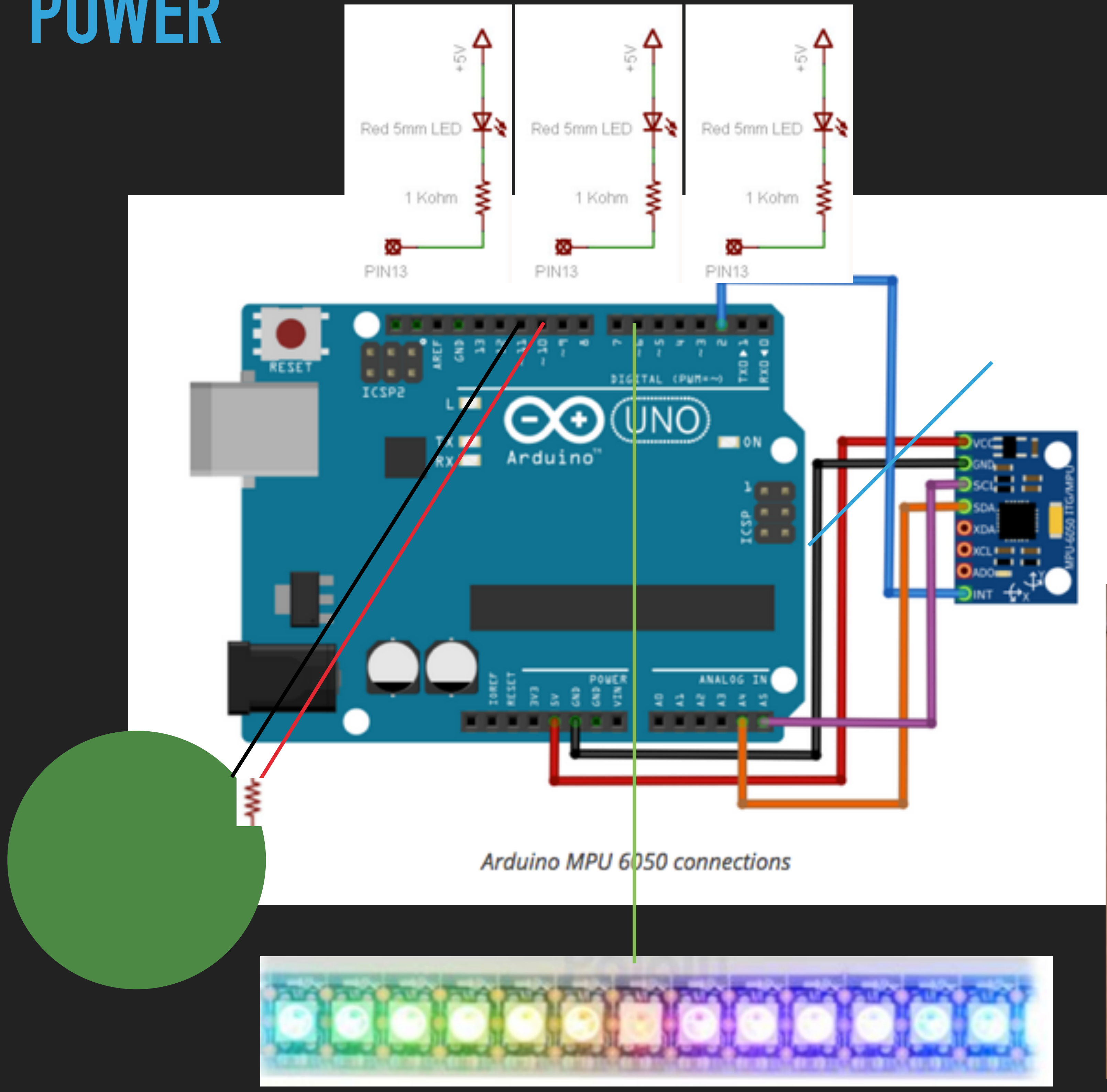
WS2812B



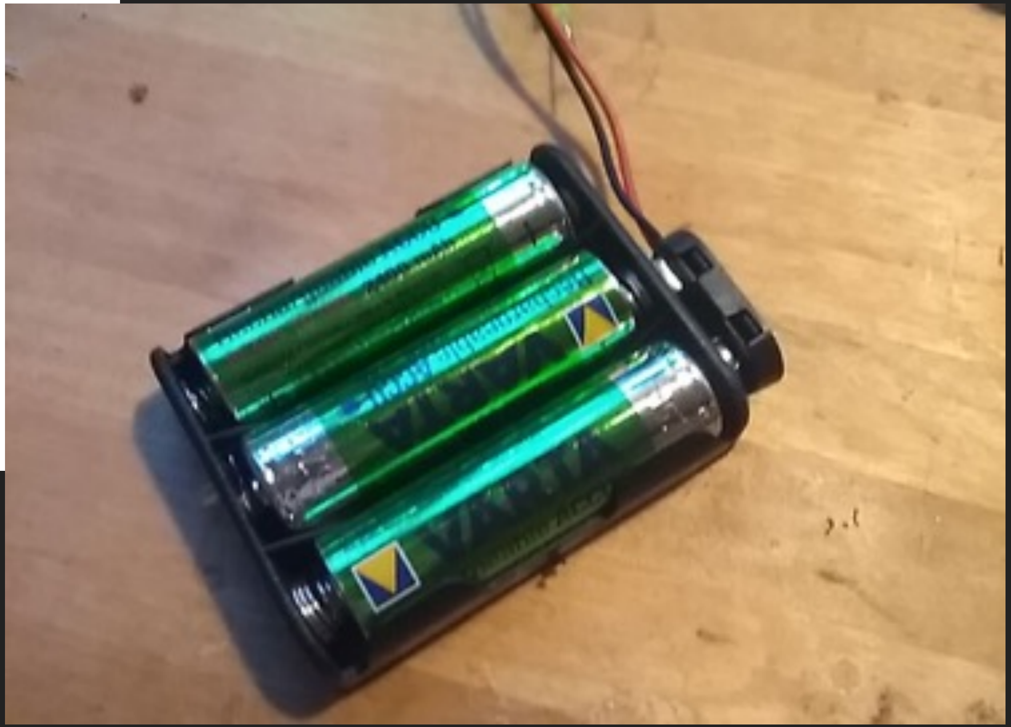
Arduino MPU 6050 connections

OOD
WILL
TEAR
US
APART

POWER

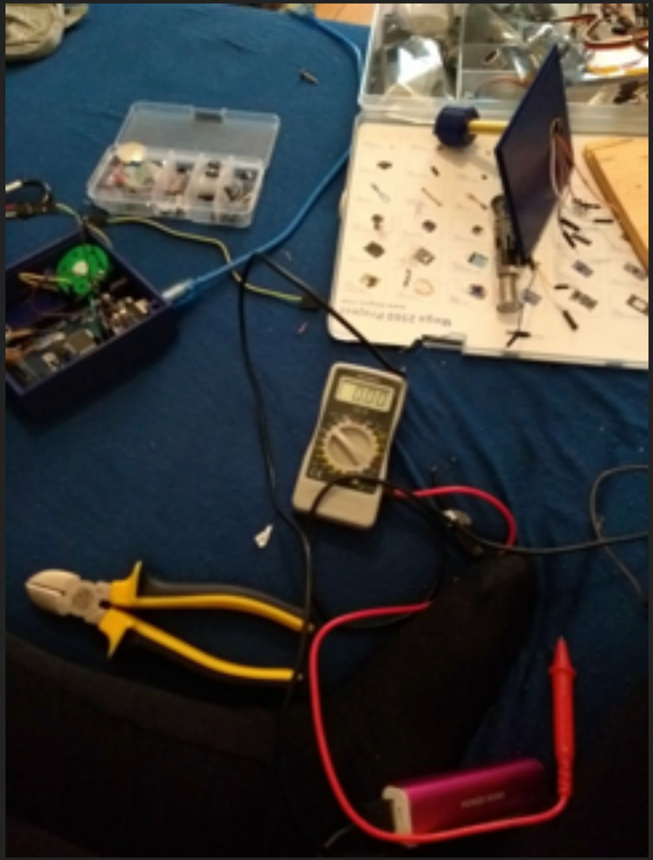


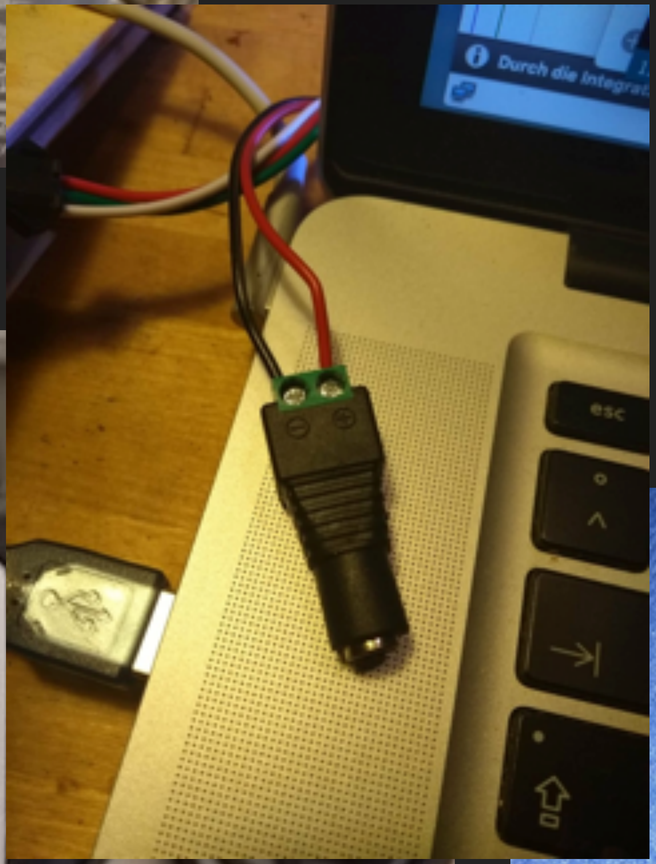
Arduino MPU 6050 connections



WATT ?!?

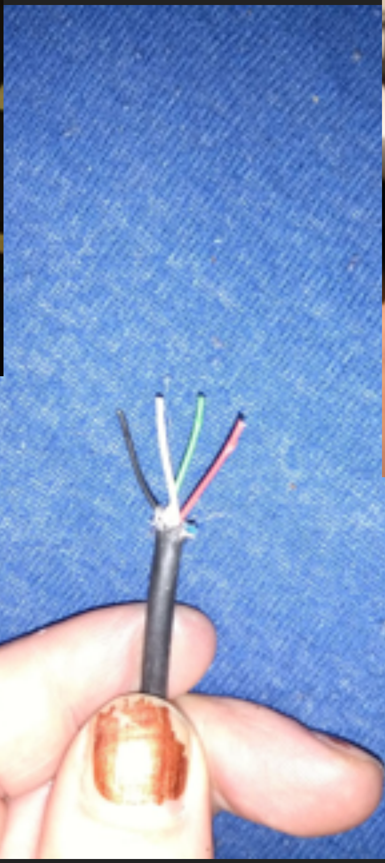
VOLTCRAFT.
Distributed by Conrad Electronic SE
Model no.: SPS24-48W-A
PRI:100-240 V/AC, 50/60Hz, 1000 mA
SEC:9/12/13.5/15/18/20/24 V/DC, 2000 mA
FOR OFFICE EQUIPMENT USE ONLY




Durch die Integrat
esc

C765
Intertek
GS
CE
MADE IN CHINA

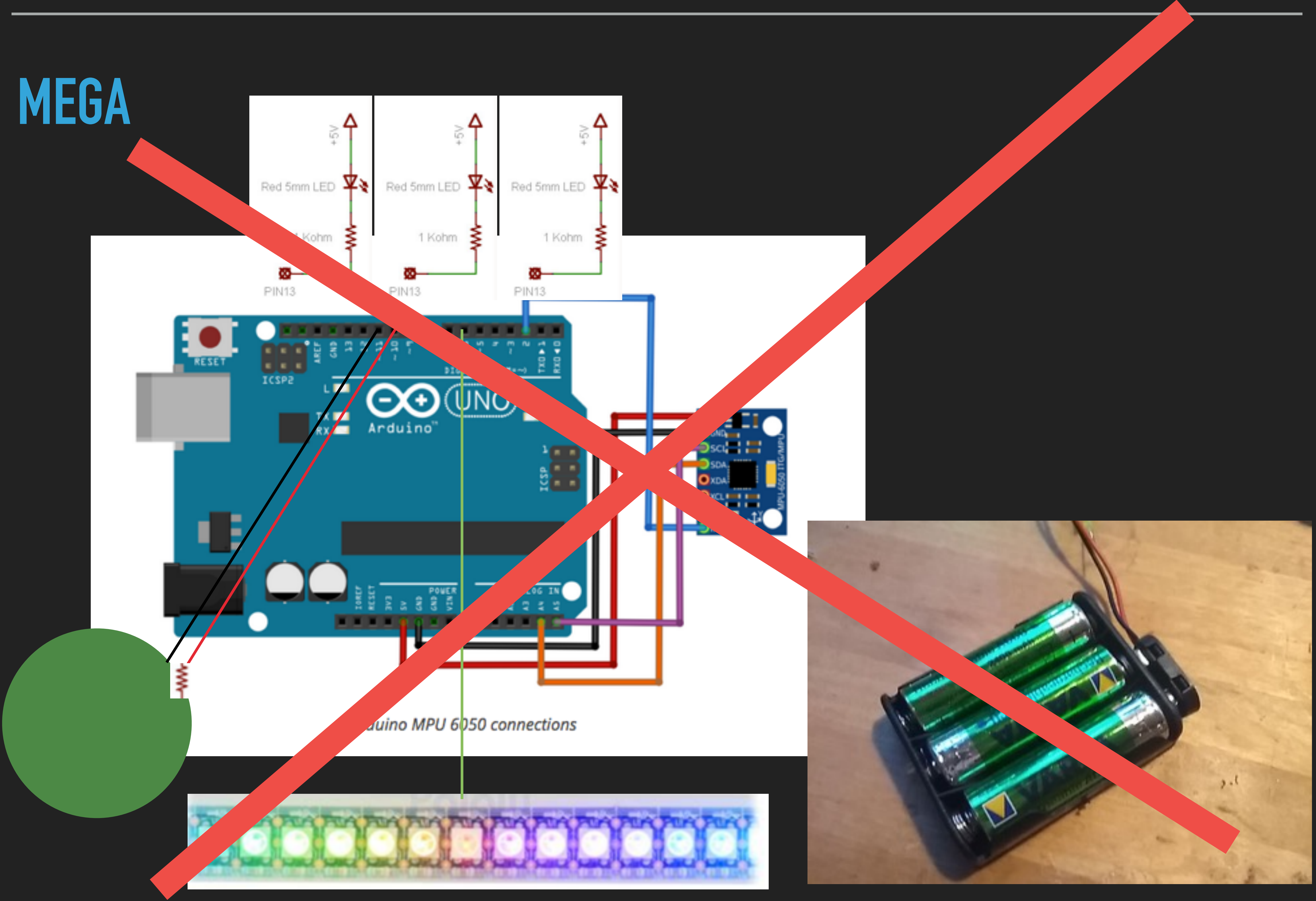
AC ADAPTER/ Adaptador de calco)
MODEL/Modelo A12-040N1A
SEC P/N: AD-4012NHF
INPUT/Entrada 100-240V~1A, 50-60Hz
OUTPUT/Salida 12V ⎓ 3.33A(3.33A)
SAFETY MARK
121098-11
GS
NOM
Energy Verified
Rendement
Energétique Vérifié



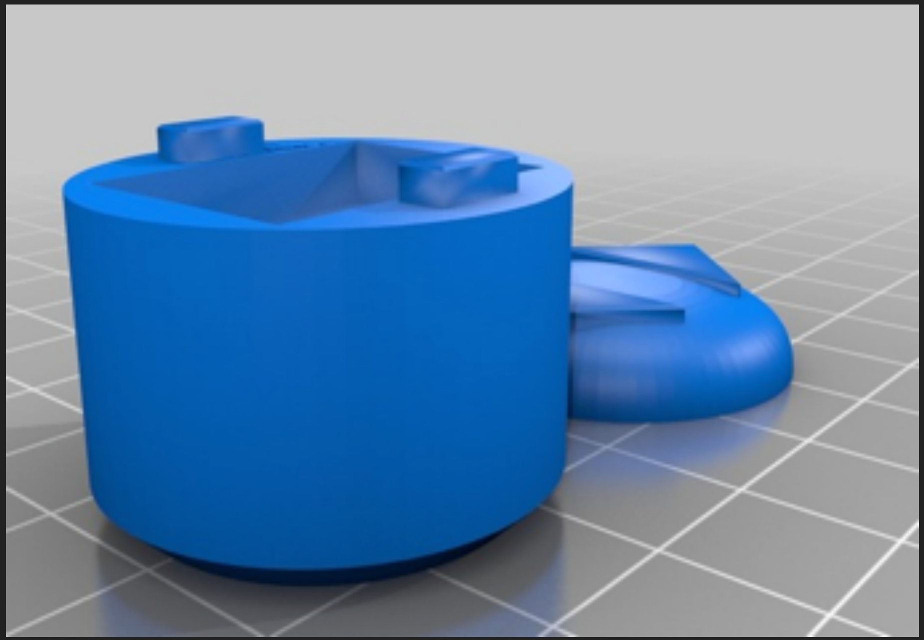
C765
Intertek
GS
CE
MADE IN CHINA

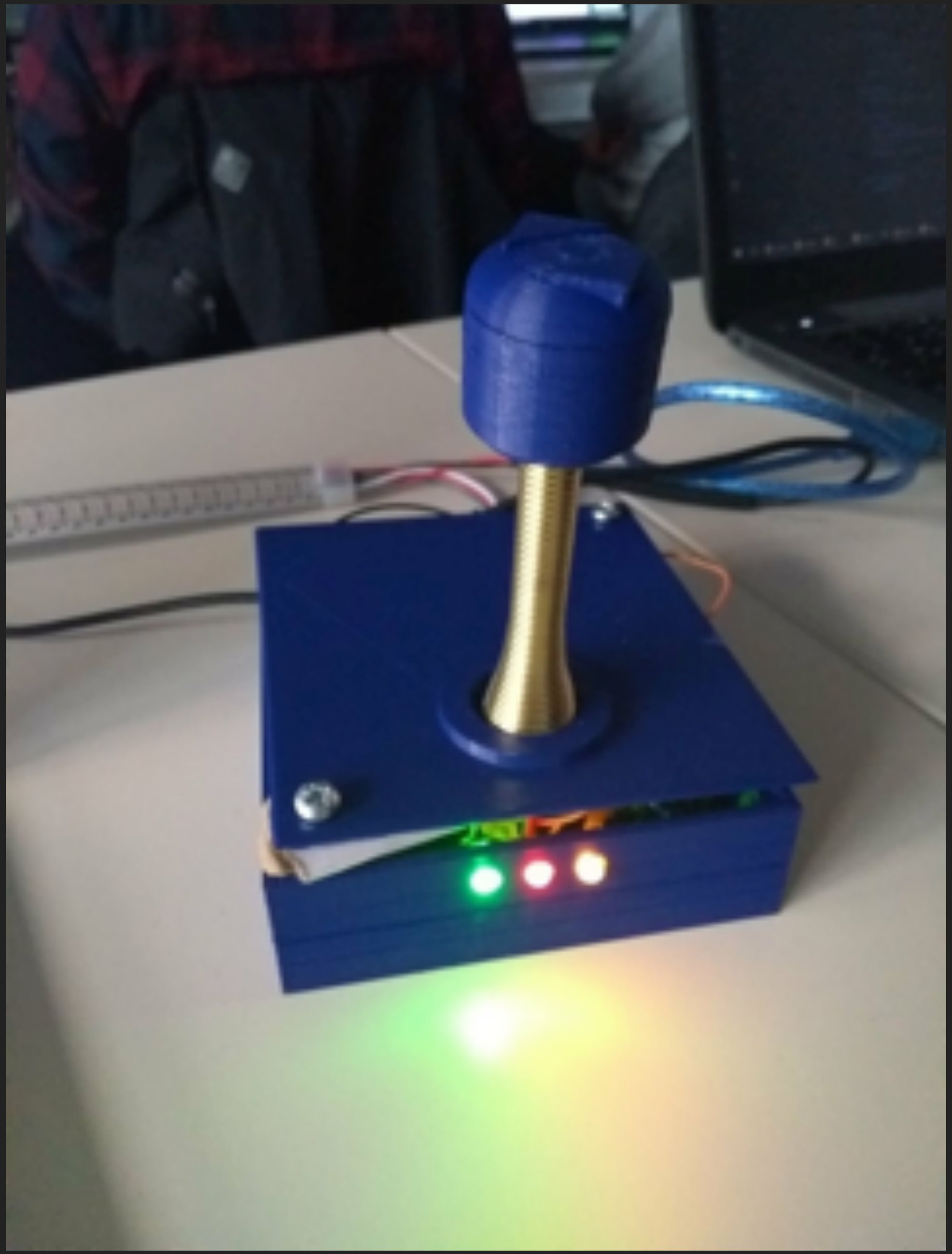
ENTEC TECHNOLOGY CO.,LTD
CE
Made in China

MEGA

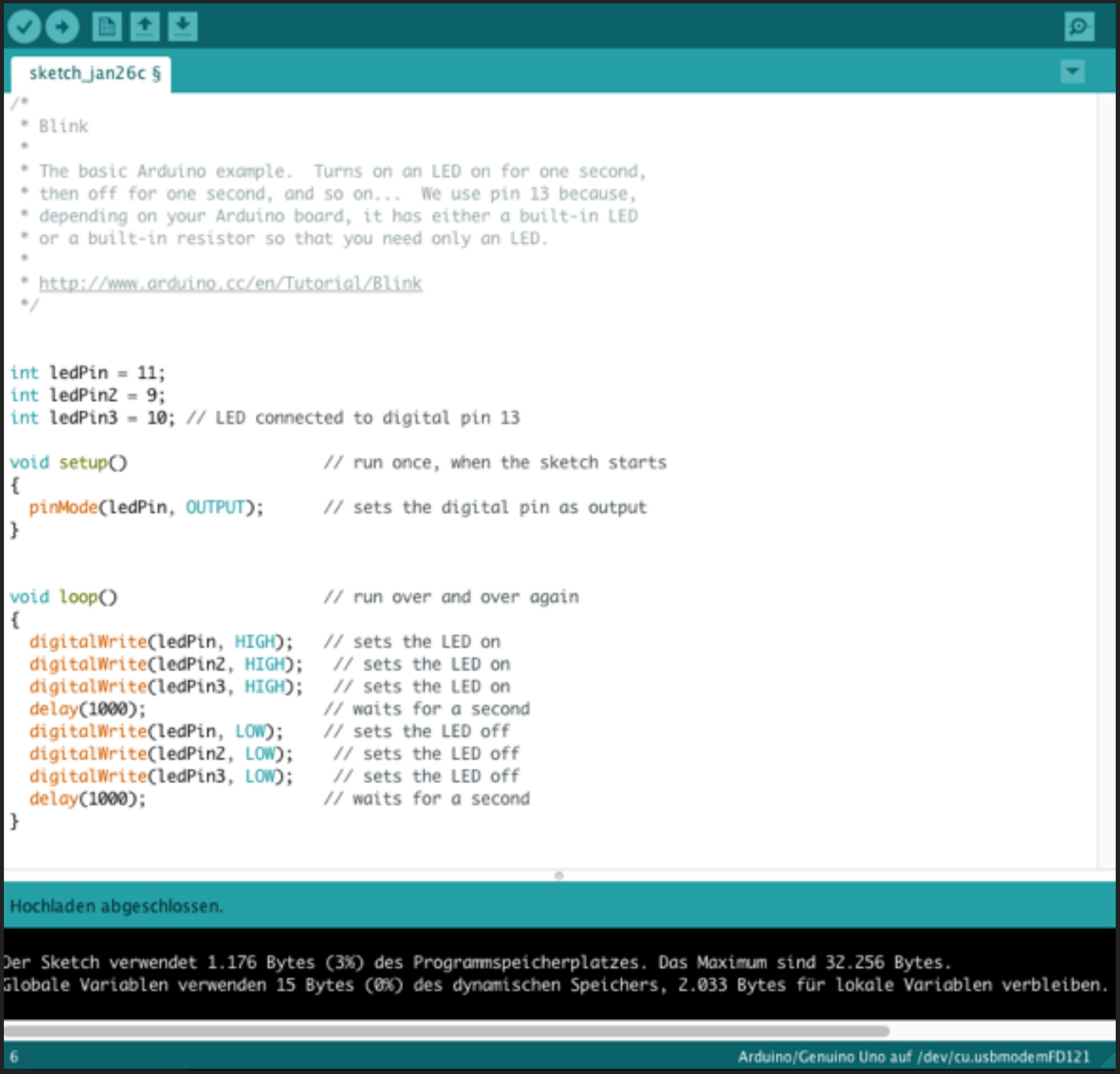

+5V
Red 5mm LED
1 Kohm
PIN13
+5V
Red 5mm LED
1 Kohm
PIN13
+5V
Red 5mm LED
1 Kohm
PIN13
RESET
ICSP2
AREF
GND
13
TX▶1
RX◀0
UNO
Arduino
ICSP
IOREF
RESET
3V3
5V
GND
VIN
POWER
ANALOG IN
A3
A4
A5
GND
SCL
SDA
XDA
XCL
MPU-6050 ITG/MPU
uino MPU 6050 connections

CASE





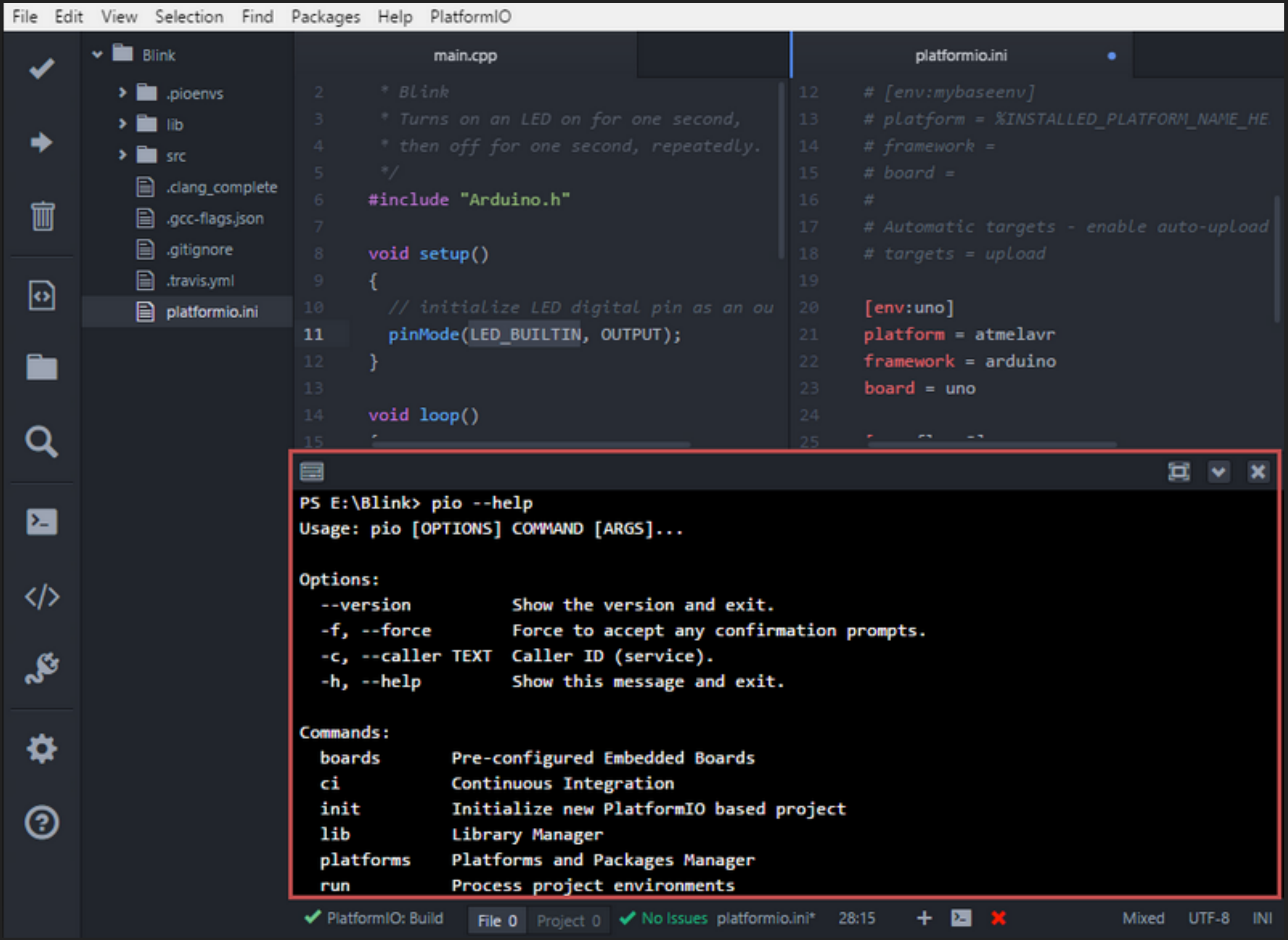
ARDUINO IDE



```
/*
 * Blink
 *
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */


int ledPin = 11;
int ledPin2 = 9;
int ledPin3 = 10; // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}


void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  digitalWrite(ledPin2, HIGH);   // sets the LED on
  digitalWrite(ledPin3, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  digitalWrite(ledPin2, LOW);    // sets the LED off
  digitalWrite(ledPin3, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

Hochladen abgeschlossen.

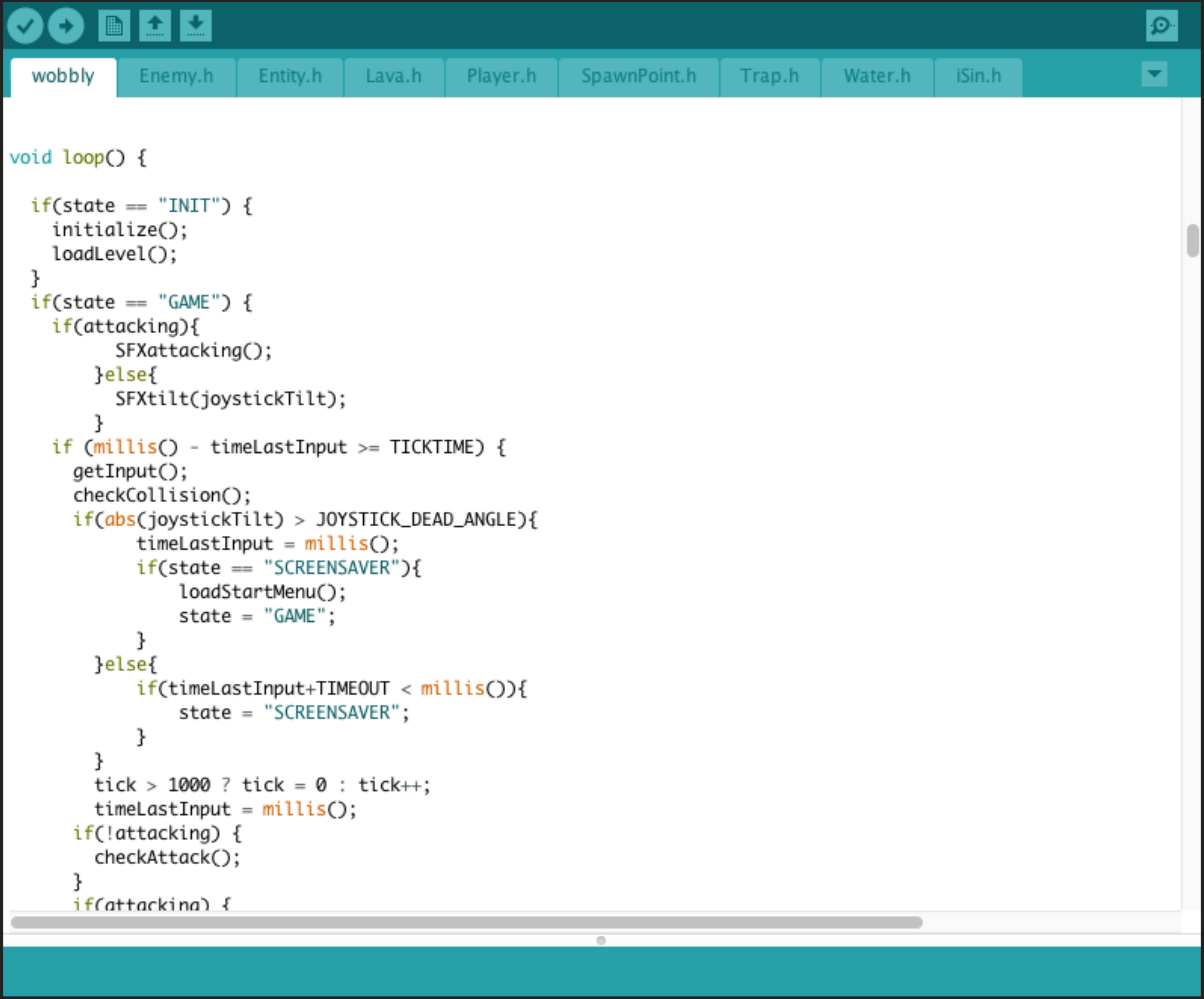
```
Der Sketch verwendet 1.176 Bytes (3%) des Programmspeicherplatzes. Das Maximum sind 32.256 Bytes.
Globale Variablen verwenden 15 Bytes (0%) des dynamischen Speichers, 2.033 Bytes für lokale Variablen verbleiben.
```



```
 * Blink
 * Turns on an LED on for one second,
 * then off for one second, repeatedly.
 */
#include "Arduino.h"

void setup()
{
  // initialize LED digital pin as an ou
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
```

```
# [env:mybaseenv]
# platform = %INSTALLED_PLATFORM_NAME_HE
# framework =
# board =
#
# Automatic targets - enable auto-upload
# targets = upload

[env:uno]
platform = atmelavr
framework = arduino
board = uno
```

```
PS E:\Blink> pio --help
Usage: pio [OPTIONS] COMMAND [ARGS]...

Options:
  --version          Show the version and exit.
  -f, --force        Force to accept any confirmation prompts.
  -c, --caller TEXT  Caller ID (service).
  -h, --help         Show this message and exit.

Commands:
  boards       Pre-configured Embedded Boards
  ci           Continuous Integration
  init         Initialize new PlatformIO based project
  lib          Library Manager
  platforms    Platforms and Packages Manager
  run          Process project environments
```



```
void loop() {

  if(state == "INIT") {
    initialize();
    loadLevel();
  }
  if(state == "GAME") {
    if(attacking){
          SFXattacking();
        }else{
          SFXtilt(joystickTilt);
        }
    if (millis() - timeLastInput >= TICKTIME) {
      getInput();
      checkCollision();
      if(abs(joystickTilt) > JOYSTICK_DEAD_ANGLE){
            timeLastInput = millis();
            if(state == "SCREENSAVER"){
                loadStartMenu();
                state = "GAME";
            }
        }else{
            if(timeLastInput+TIMEOUT < millis()){
                state = "SCREENSAVER";
            }
        }
        tick > 1000 ? tick = 0 : tick++;
        timeLastInput = millis();
      if(!attacking) {
        checkAttack();
      }
      if(attacking) {
```

MAKING OF