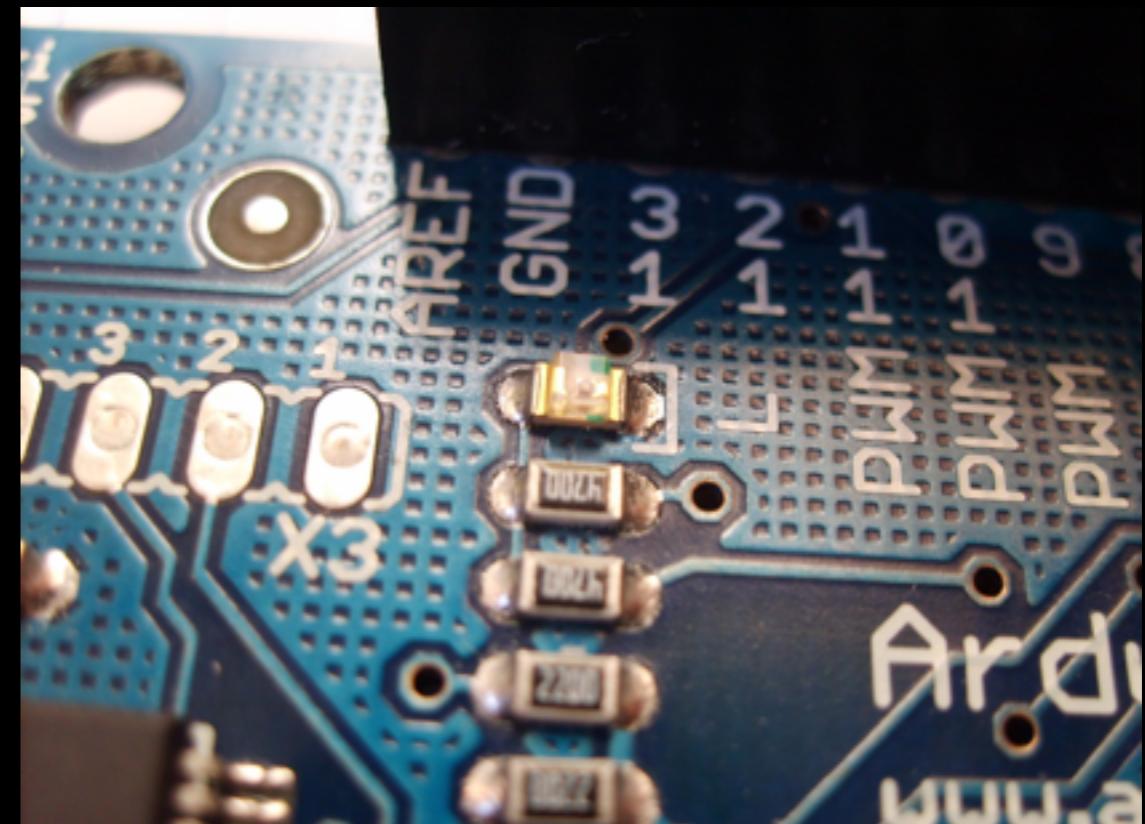


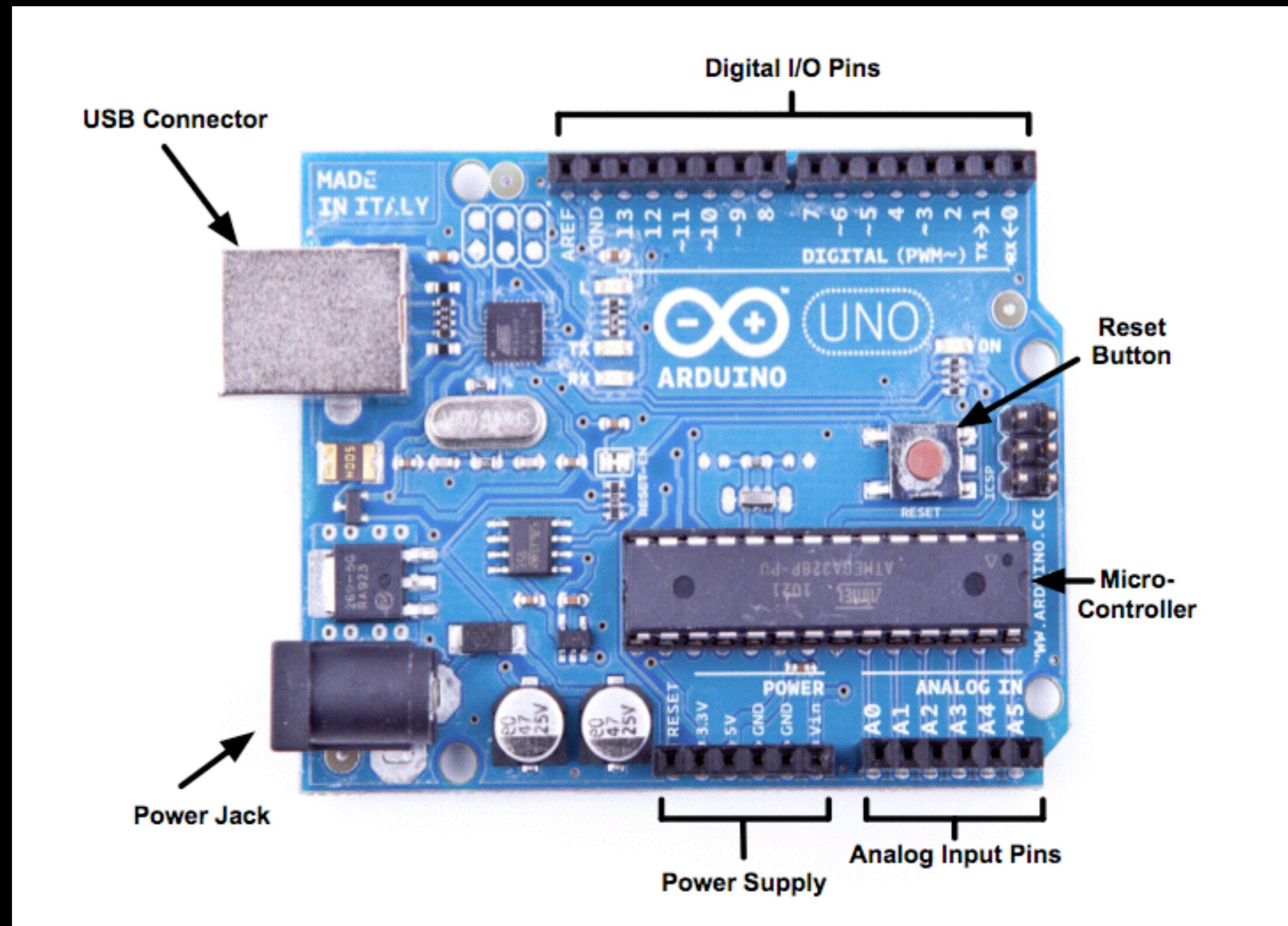


Arduino 101

What is Arduino

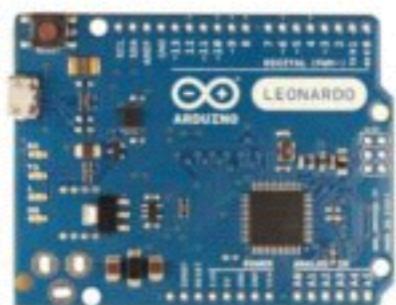
- Electronic boards
- Development environment
- Una community







Arduino Uno



Arduino Leonardo



Arduino Mega ADK



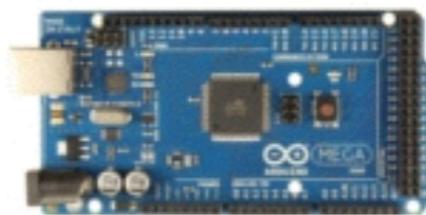
Arduino Tre



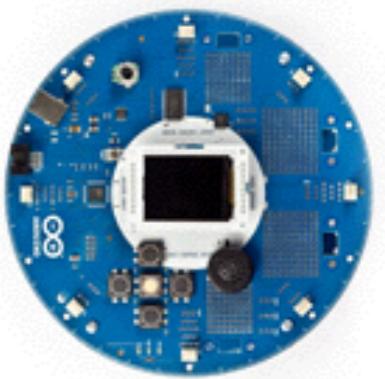
Arduino Due



Arduino Yún



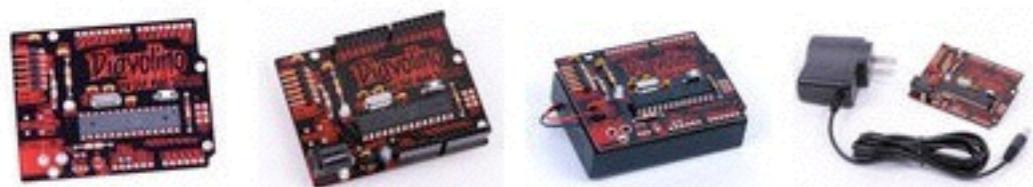
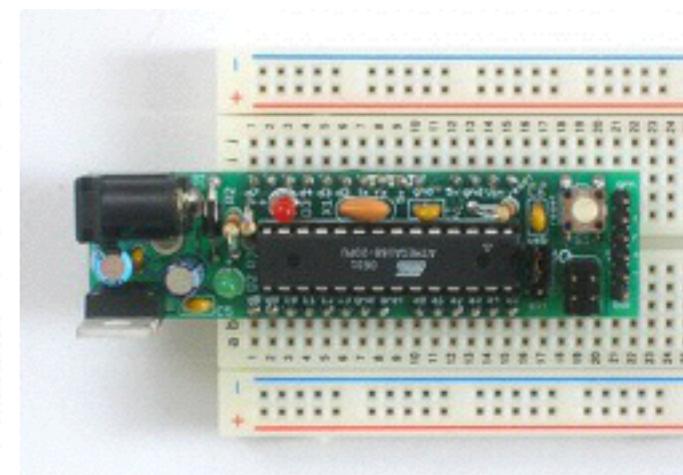
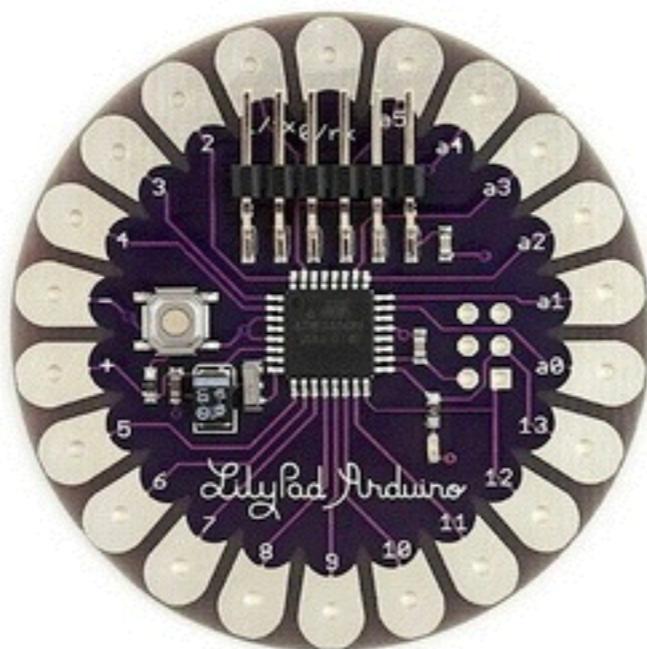
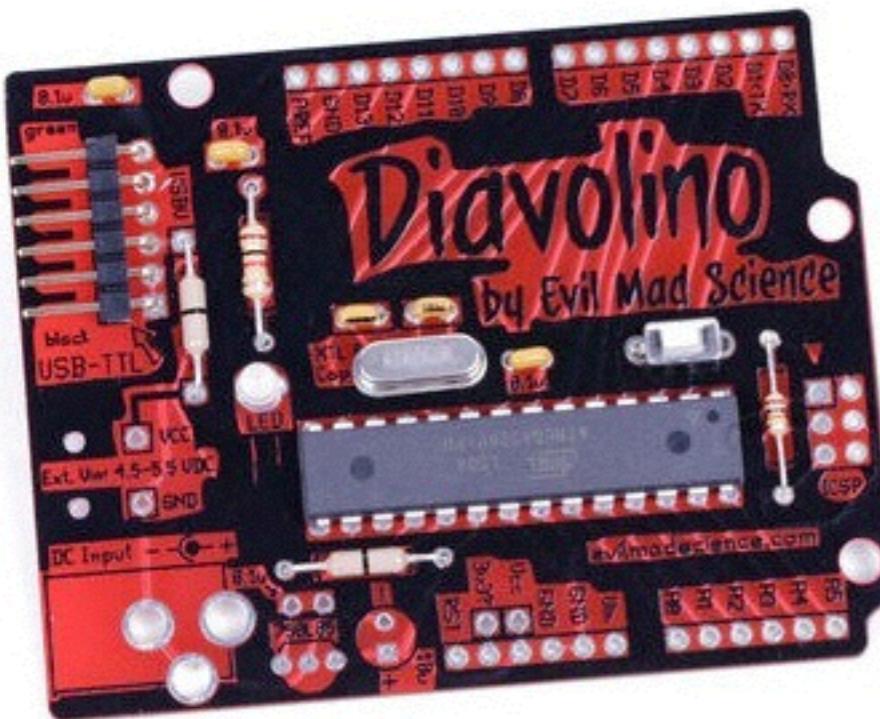
Arduino Mega 2560



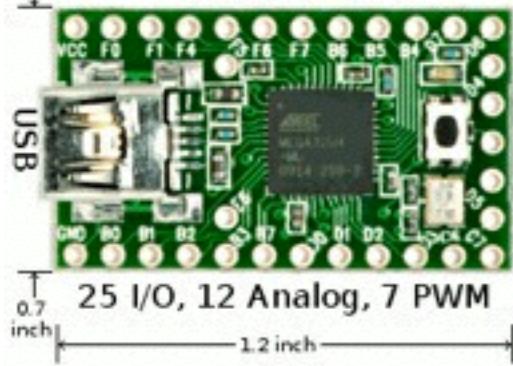
Arduino Robot



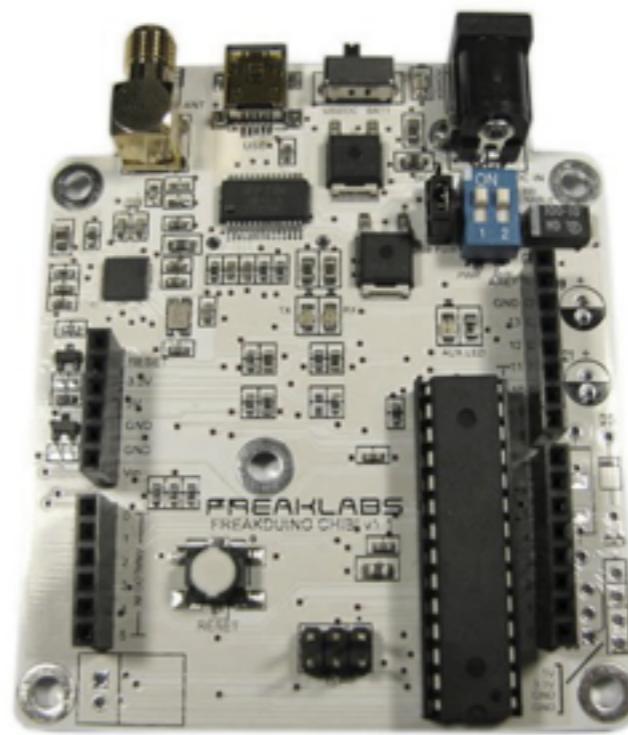
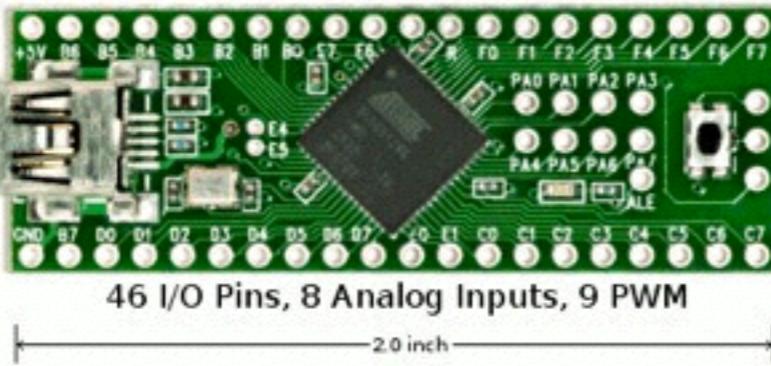
open source
hardware



Teensy 2.0



Teensy++ 2.0

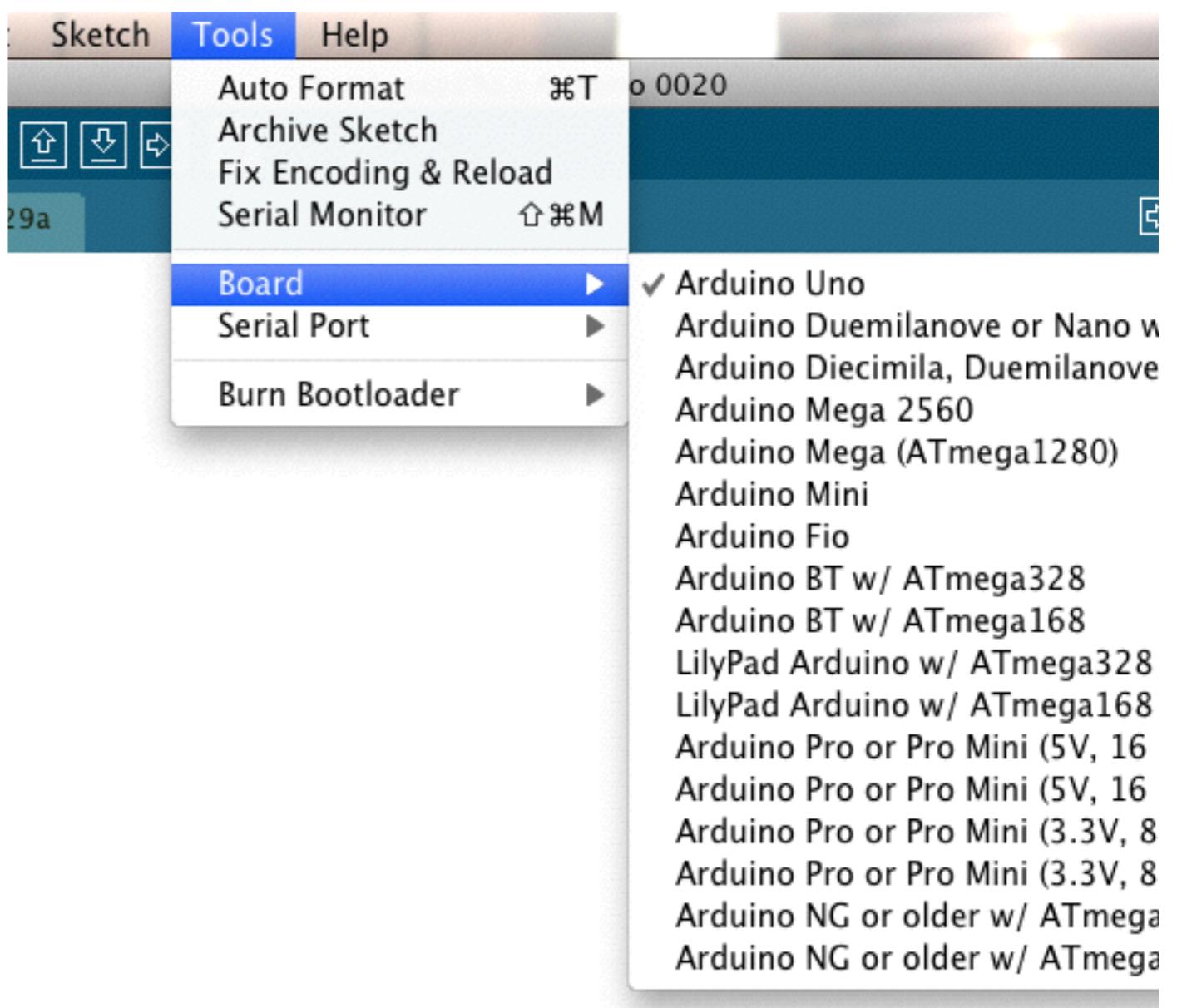


The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0". The main window displays the "Blink" sketch code. The code is a classic "Blink" example that turns an LED on for one second and off for one second, repeatedly. It includes comments explaining the purpose of the code and the setup of pin 13 as an output. The code uses the `digitalWrite` function to control the LED and `delay` functions to wait for one second between each state change. The status bar at the bottom indicates "1" and "Arduino Uno on /dev/tty.usbmodemfd131".

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/
void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has an LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

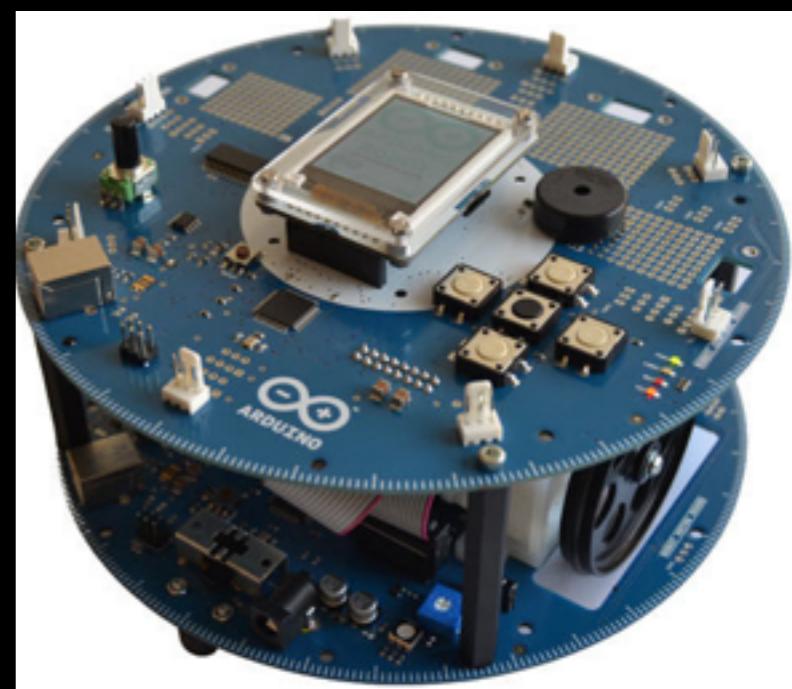
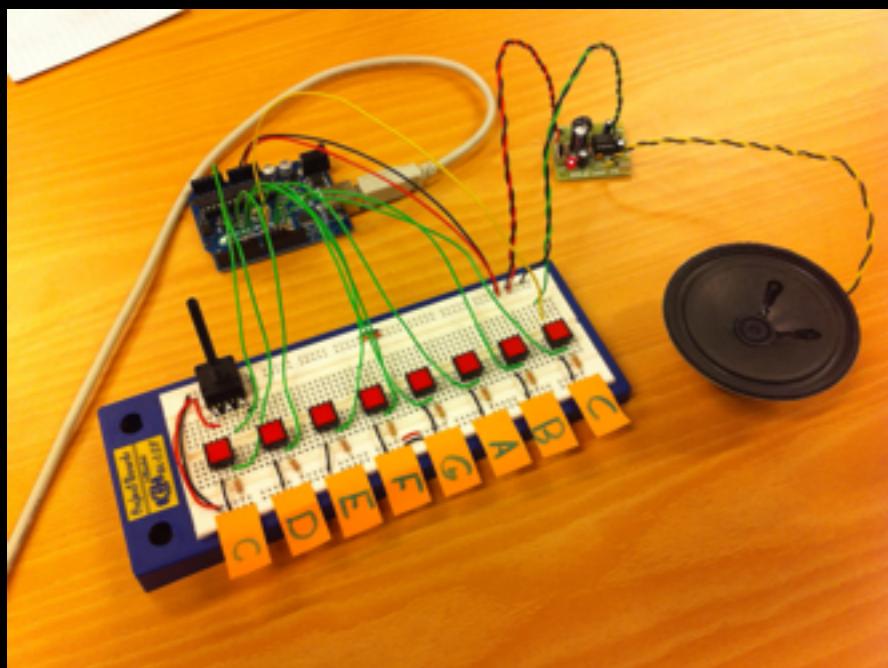
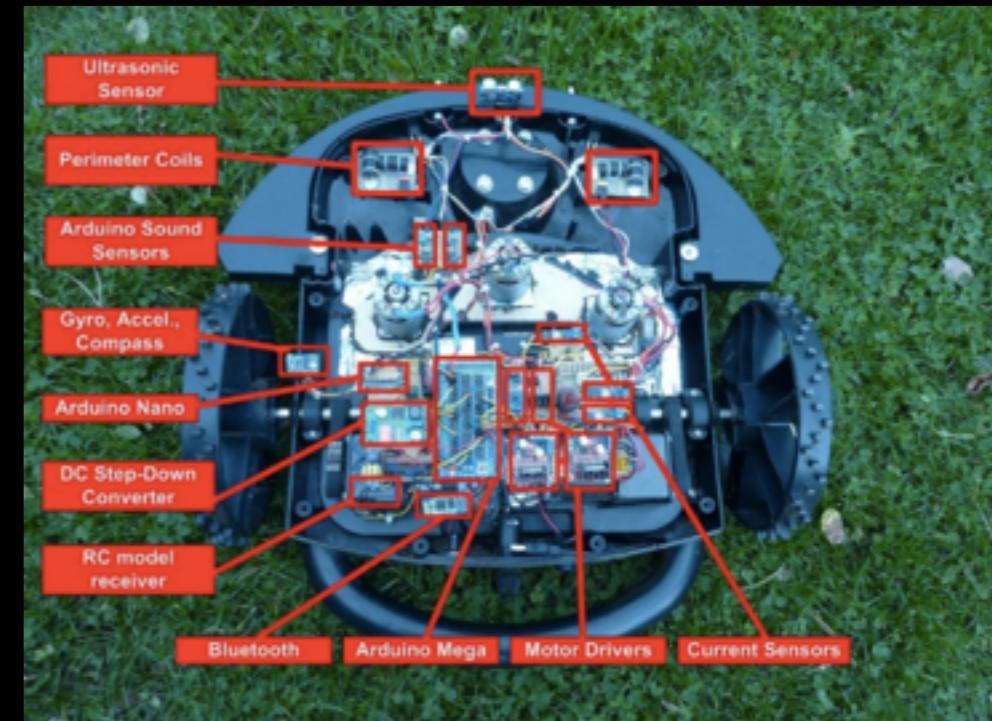
void loop() {
    digitalWrite(13, HIGH); // set the LED on
    delay(1000);           // wait for a second
    digitalWrite(13, LOW);  // set the LED off
    delay(1000);           // wait for a second
}
```



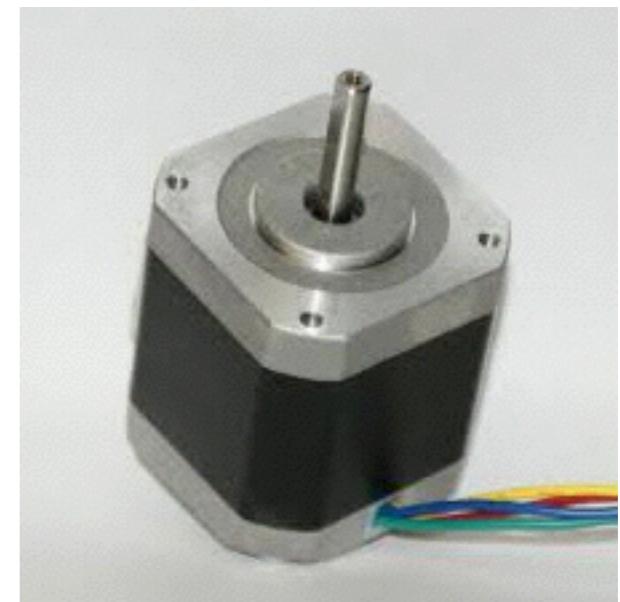


<http://playground.arduino.cc>

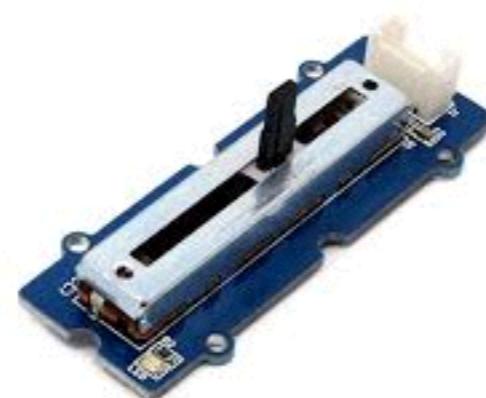
Applications

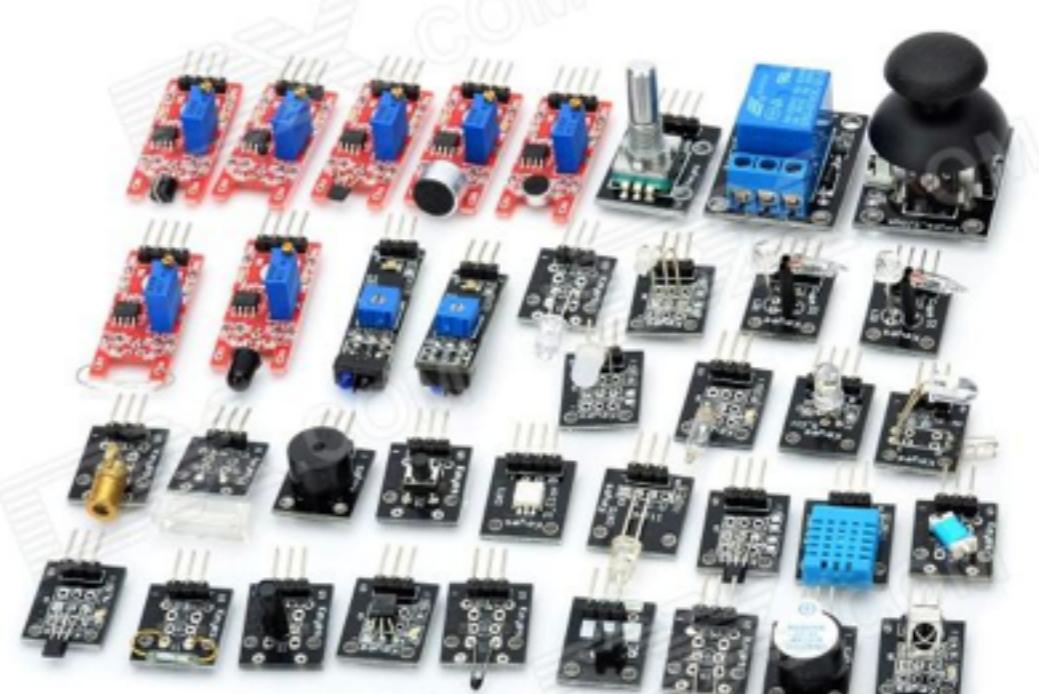
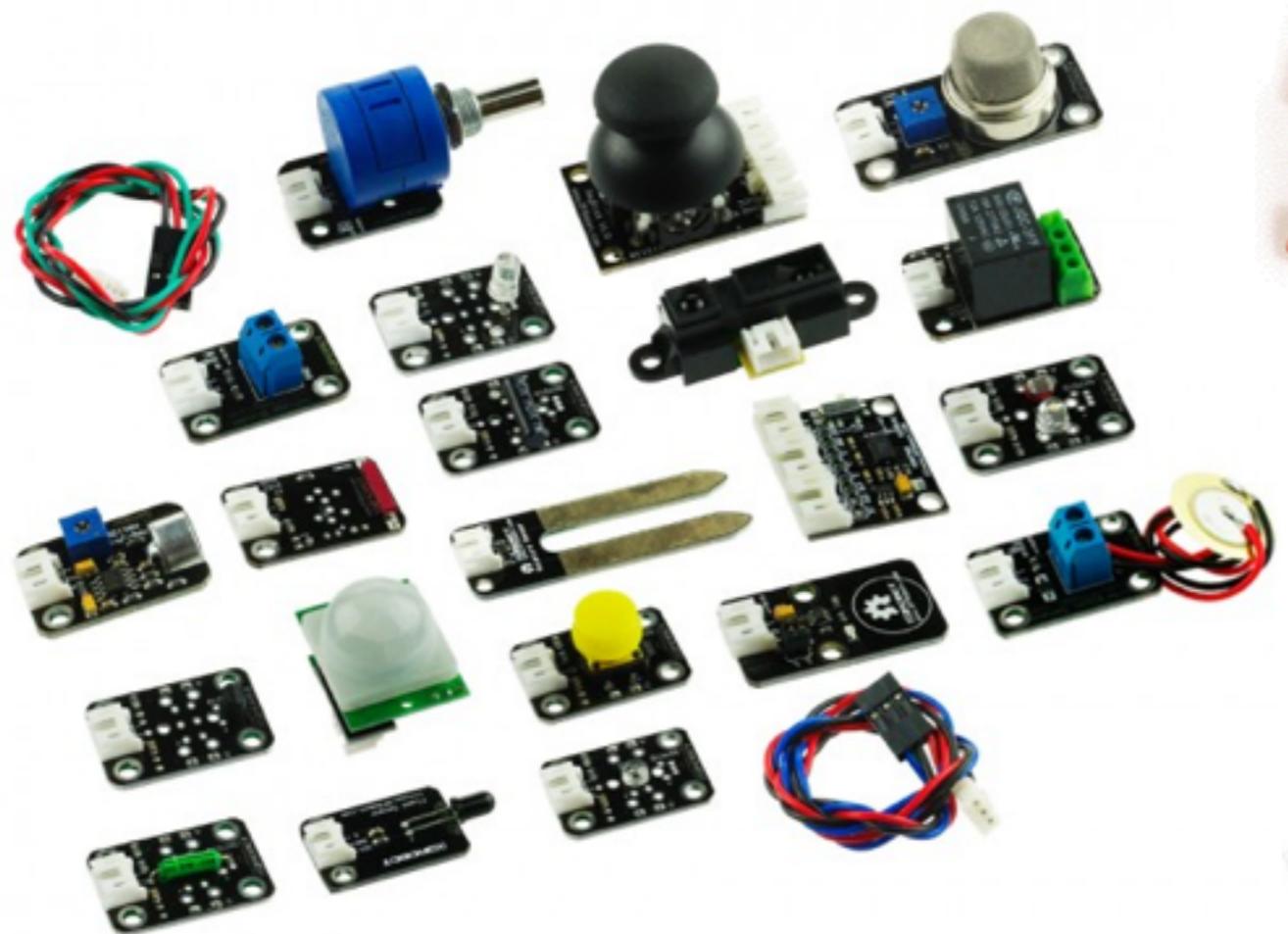


Output

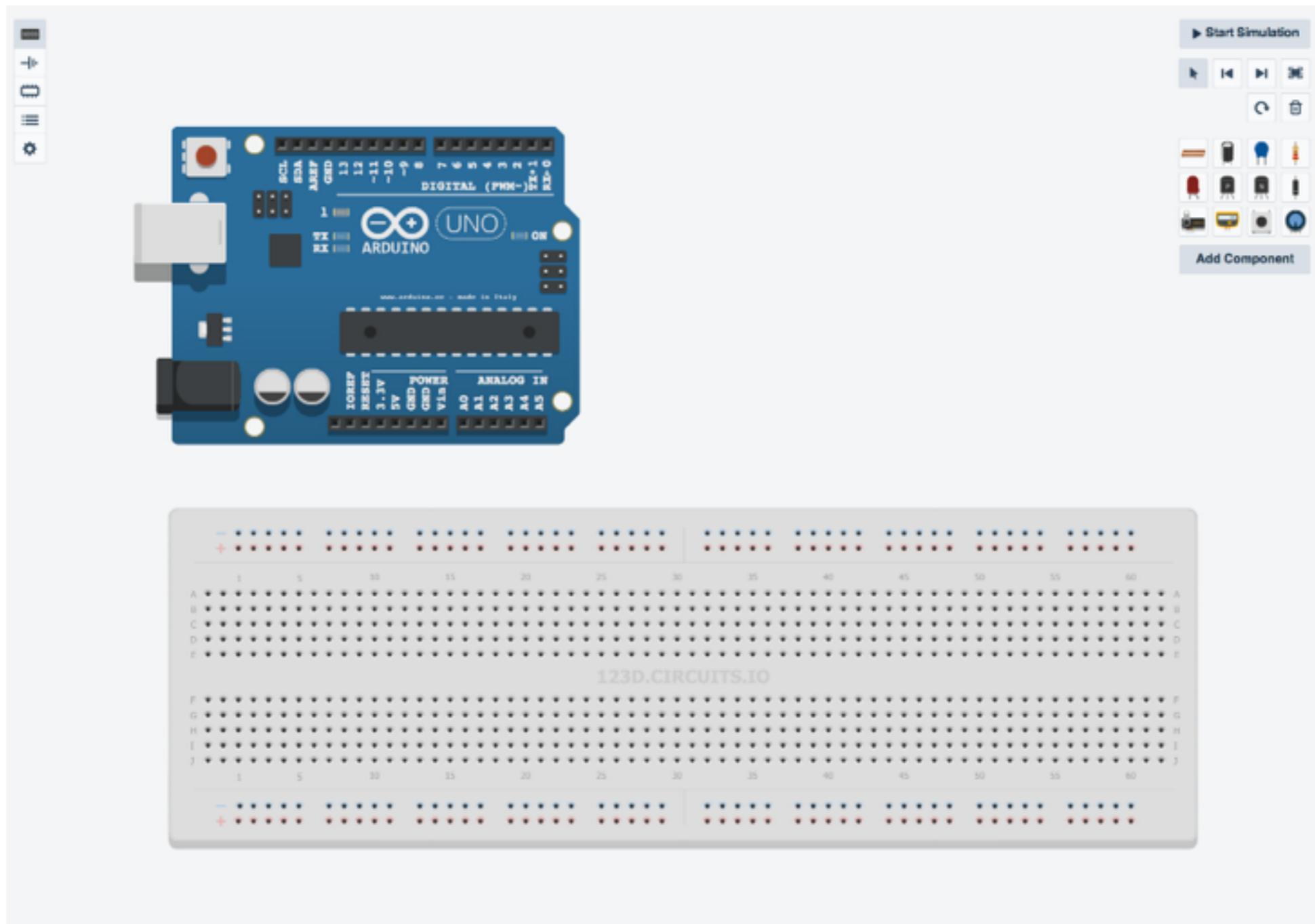


Input





Serial	Bluetooth 4.0	I2C
USB	Bluetooth 2.0	SPI
Ethernet	GPS	TWI
Wifi	RF	CAN
Zigbee	Midi	



```
// this is a comment
```

comment

```
int led = 13;
```

variable

```
void setup()
```

```
{
```

```
  pinMode(ledPin, OUTPUT);
```

setup

```
}
```

```
void loop()
```

```
{
```

```
  pinMode(ledPin, OUTPUT);
```

infinite loop

```
  digitalWrite(ledPin, HIGH);
```

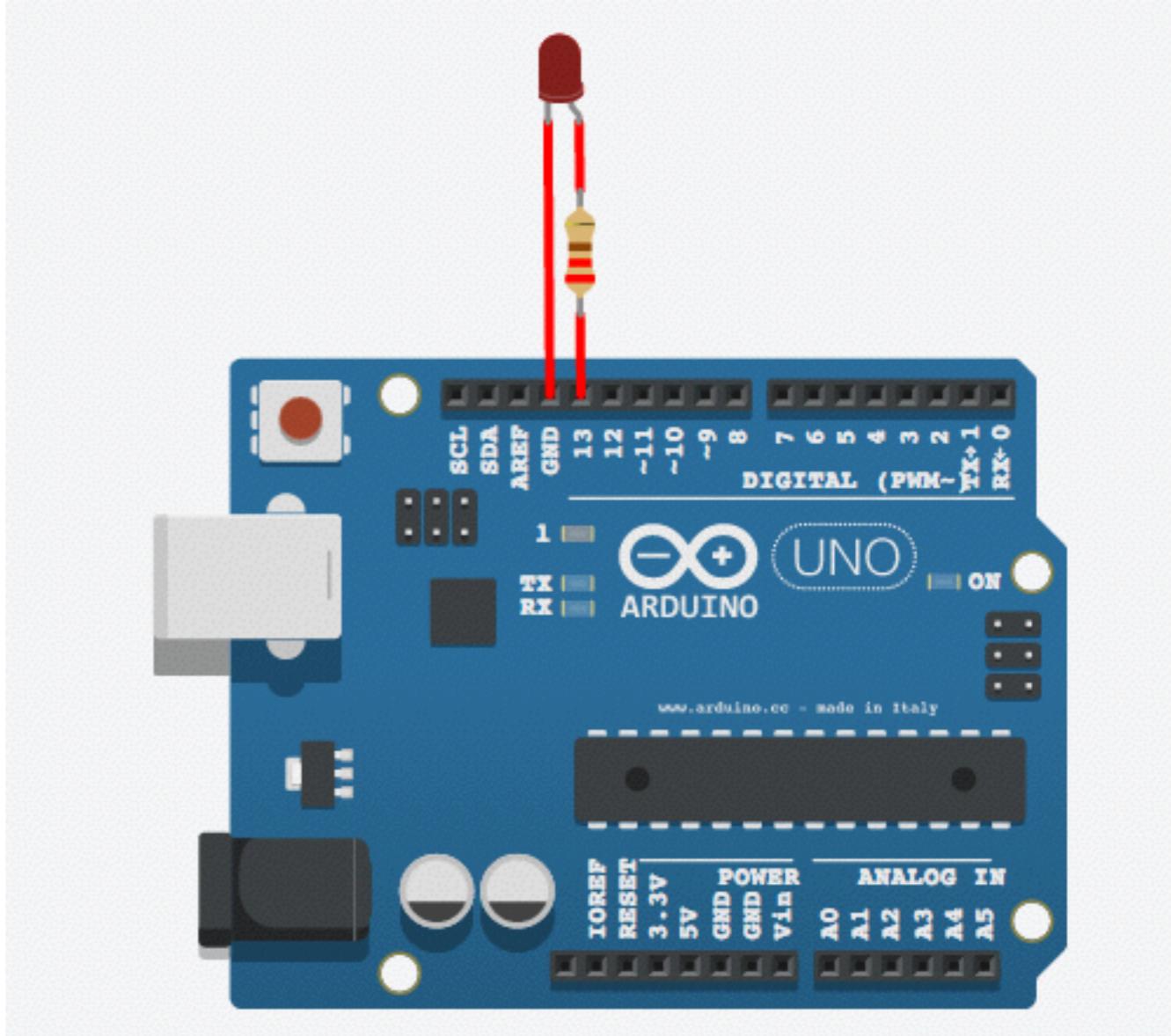
```
}
```

Light an LED

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
  delay(1000);                // wait for a second
}
```



Debug and talking to the PC



```
int led = 13;

void setup() {
    Serial.begin(9600);
    pinMode(led, OUTPUT);
}

void loop() {
    Serial.println("LED ON");
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second

    Serial.println("LED OFF");
    digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

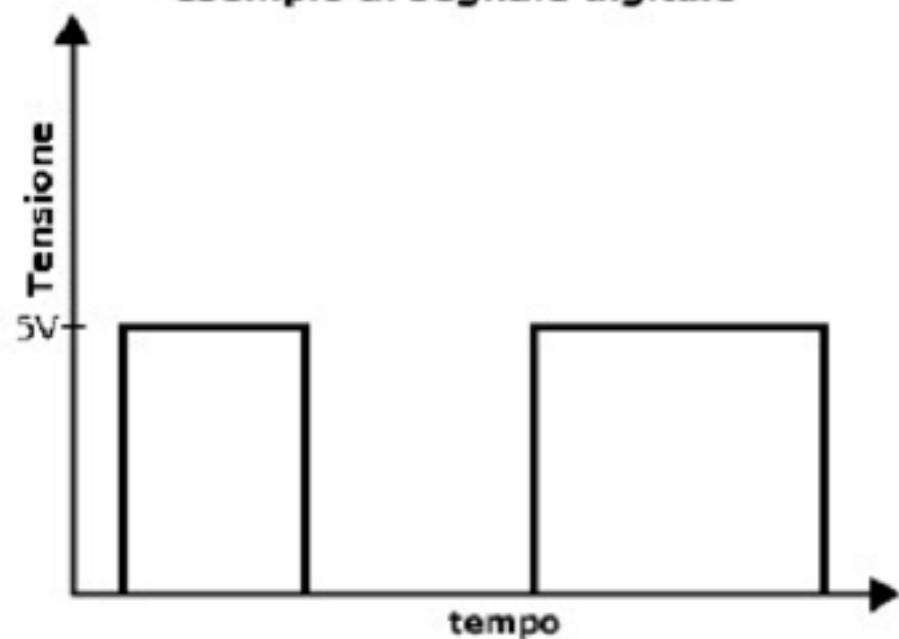


Sensors allow to “Read”
the environment

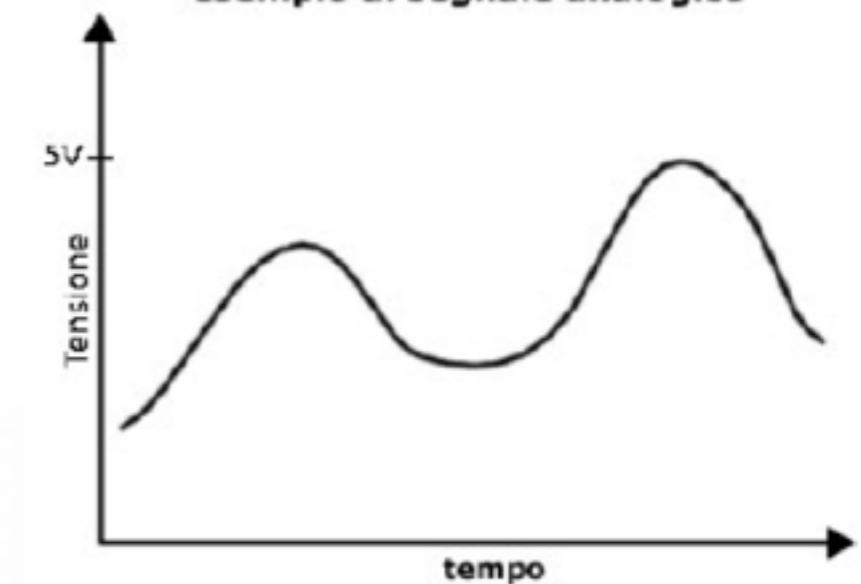
Sensors

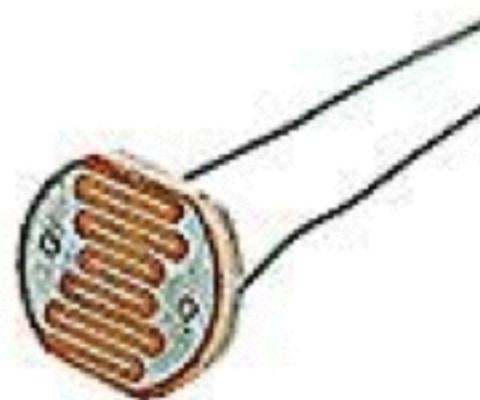
- **ANALOG:** Signal is measured as a voltage between 0 and 1024
- **DIGITAL:** Signal is a voltage (0-5v) transformed into 0 (0v) or 1 (5v)

esempio di segnale digitale

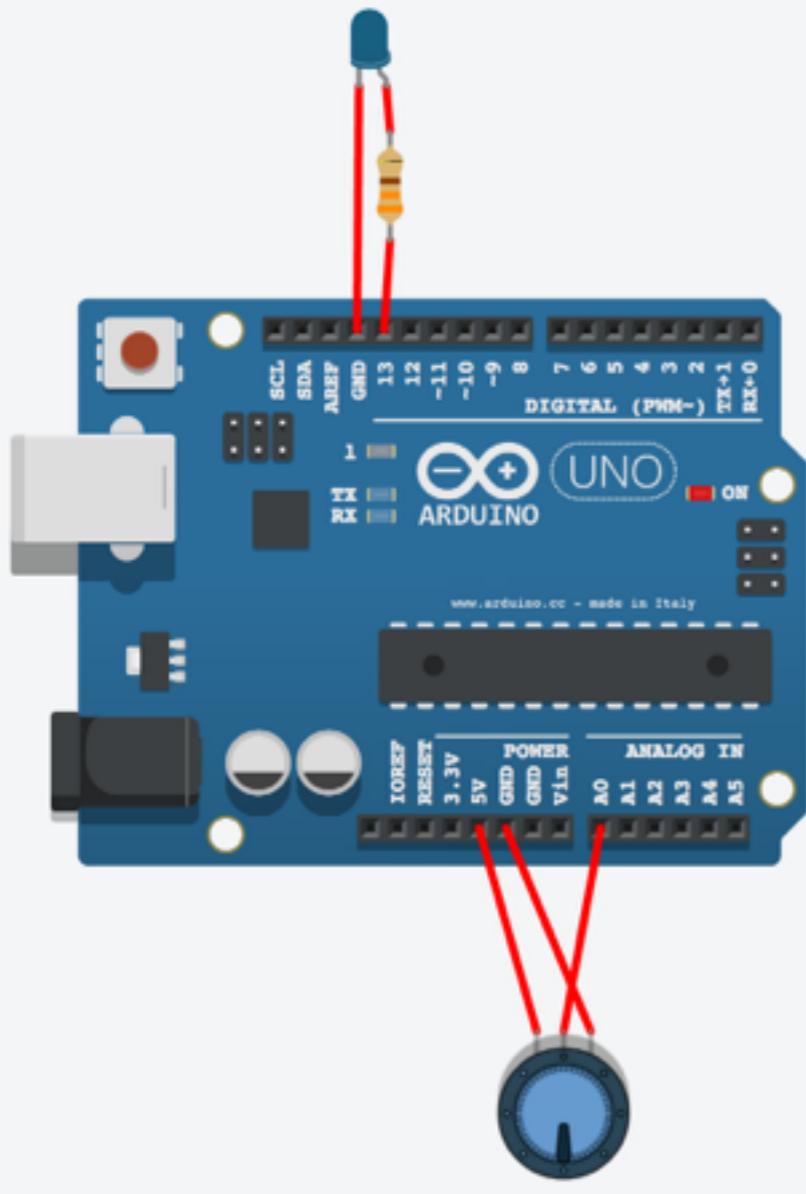


esempio di segnale analogico





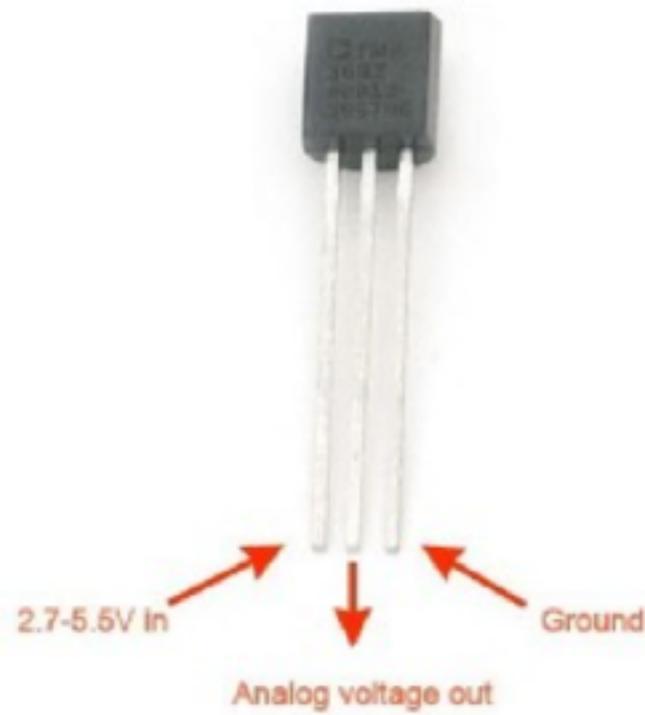
Sensors Variable Resistance



```
int sensorPin = A0; // POT  
int ledPin = 13; // LED  
int sensorValue = 0; // VALUE
```

```
void setup() {  
    pinMode(ledPin, OUTPUT);  
}
```

```
void loop() {  
    sensorValue = analogRead(sensorPin)  
    digitalWrite(ledPin, HIGH);  
    delay(sensorValue);  
    digitalWrite(ledPin, LOW);  
    delay(sensorValue);  
}
```



Temperature



Acceleration

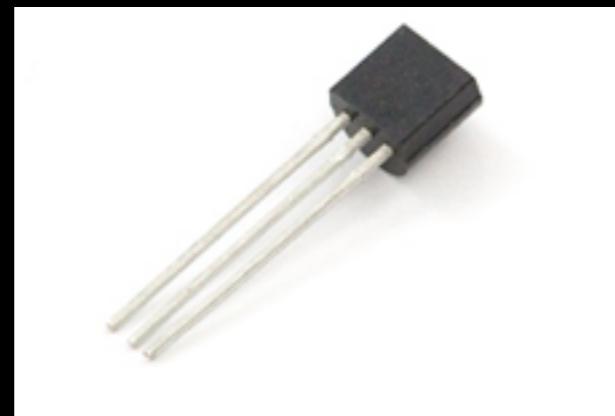


Distance

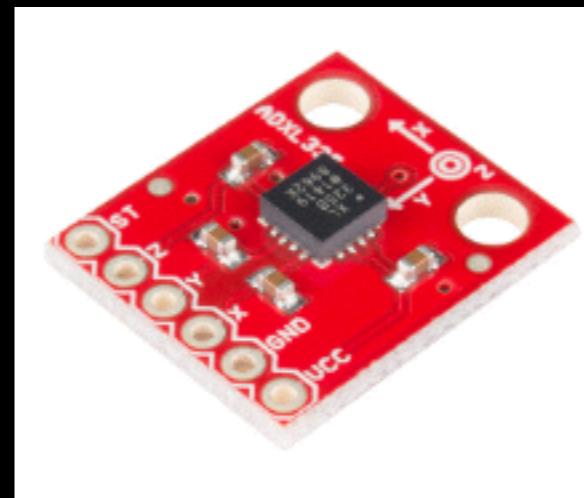
INTEGRATED SENSORS



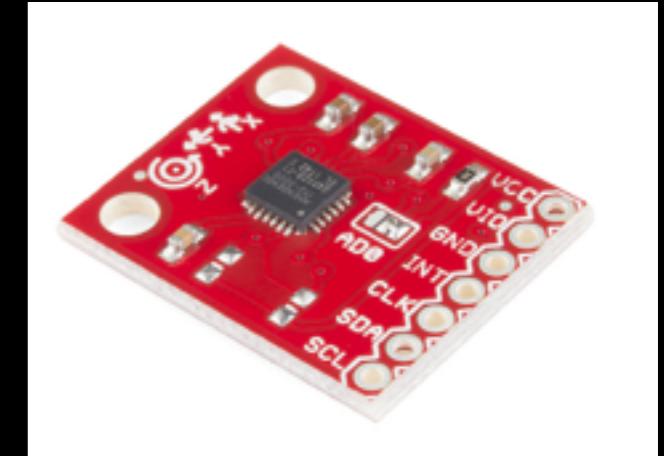
Buttons



Temp



Accel.

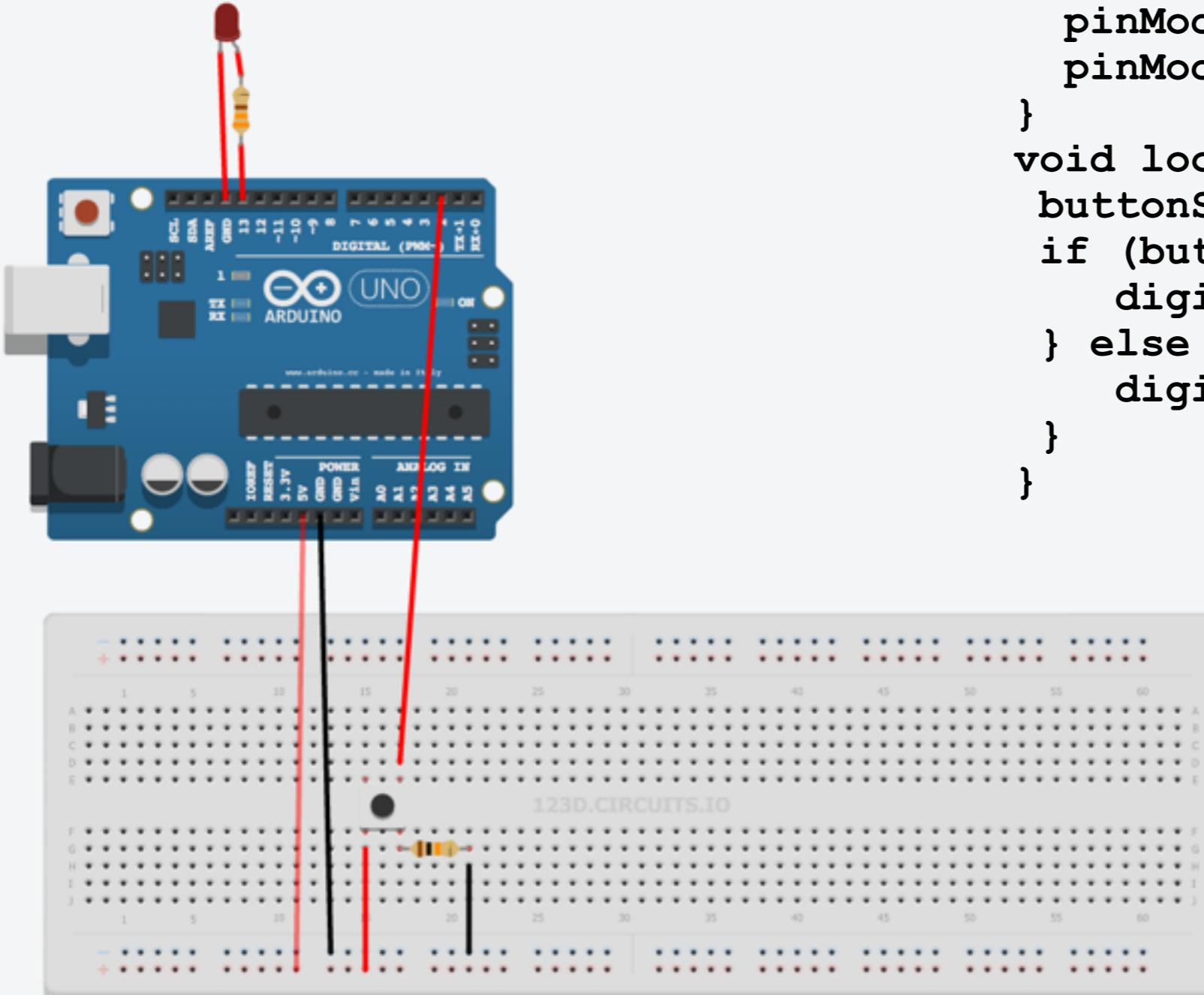


Gyroscope

Advanced Sensors

Talk using “PROTOCOLS”
Two Wire - i2c - SPI - Serial

```
int led = 13;  
int button = 2;  
int buttonState = 0;  
  
void setup() {  
    pinMode(led, OUTPUT);  
    pinMode(button, INPUT);  
}  
void loop() {  
    buttonState = digitalRead(button);  
    if (buttonState == HIGH) {  
        digitalWrite(led, HIGH);  
    } else {  
        digitalWrite(led, LOW);  
    }  
}
```

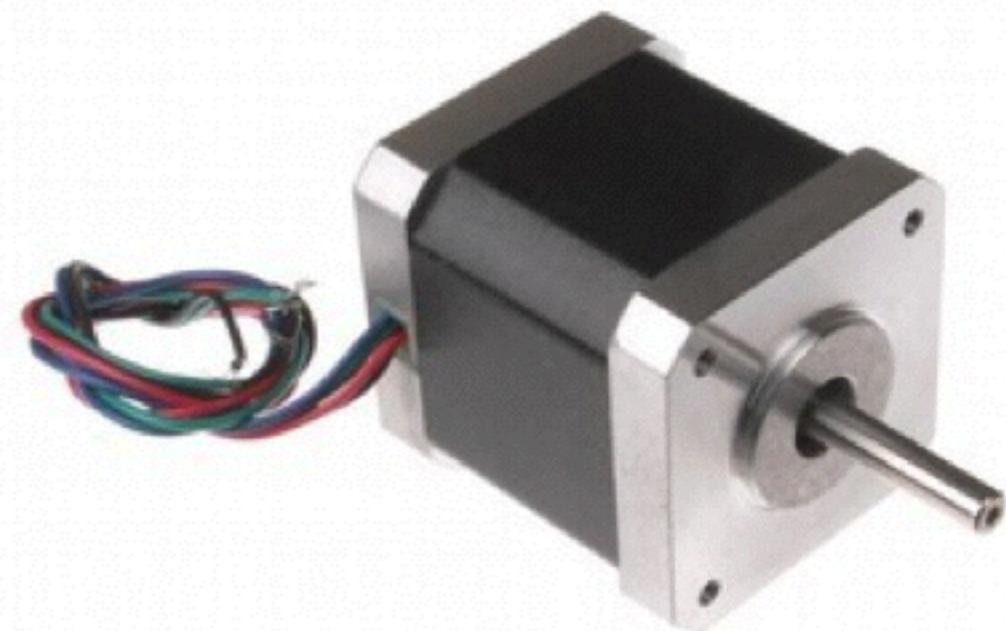




Led



DC Motor



Motore Stepper



Display LCD



Motore servo

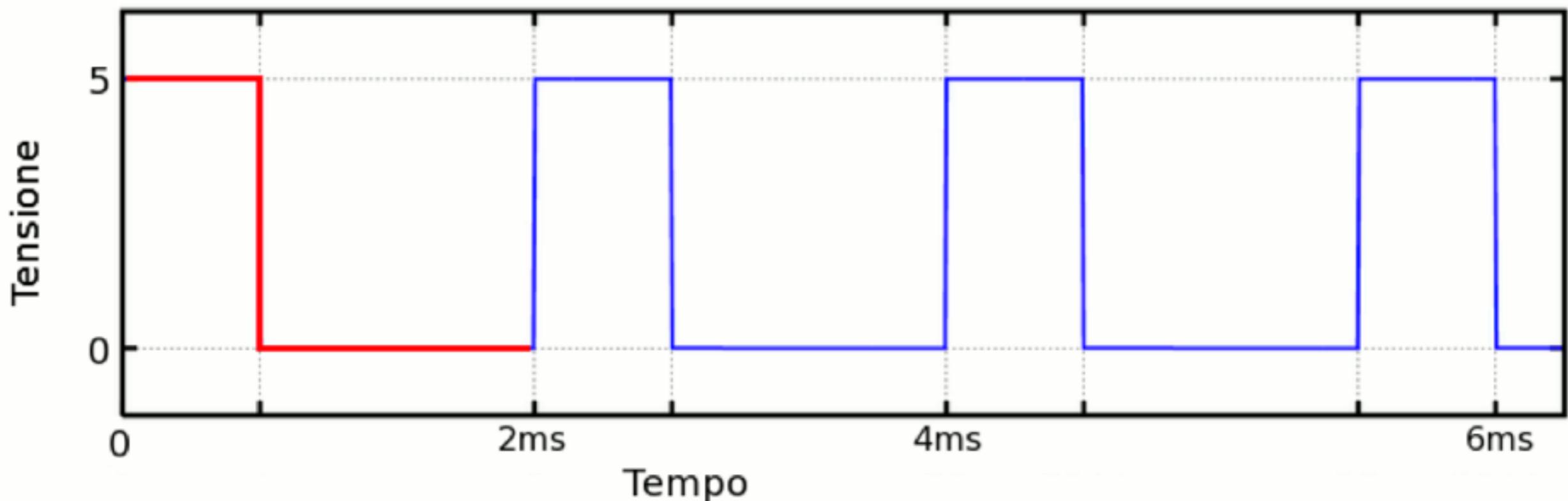


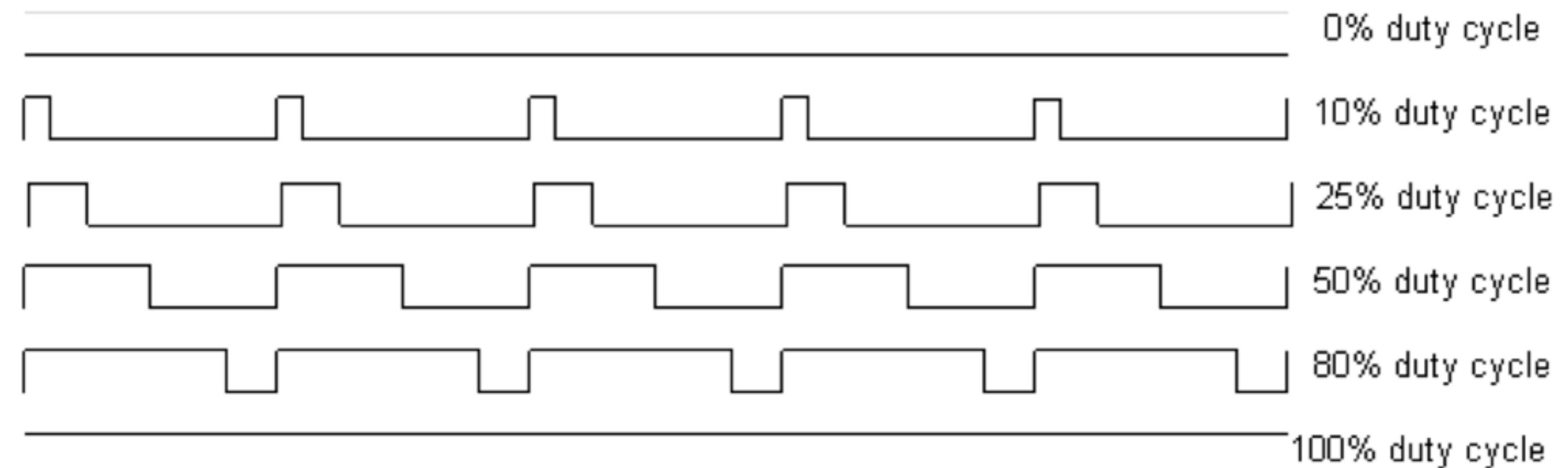
Speaker

Output using Arduino

PWM

modulazione a larghezza d'impulso

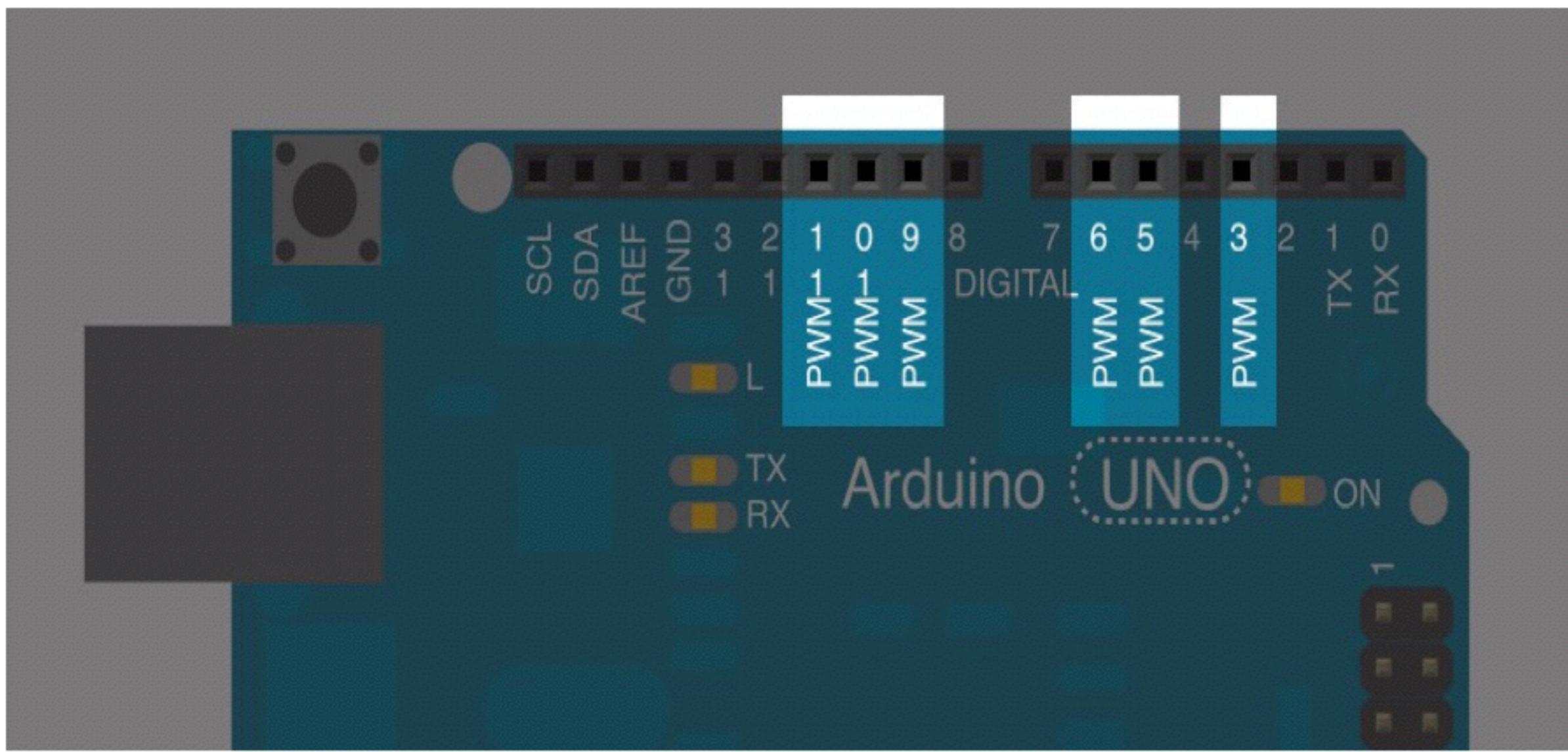




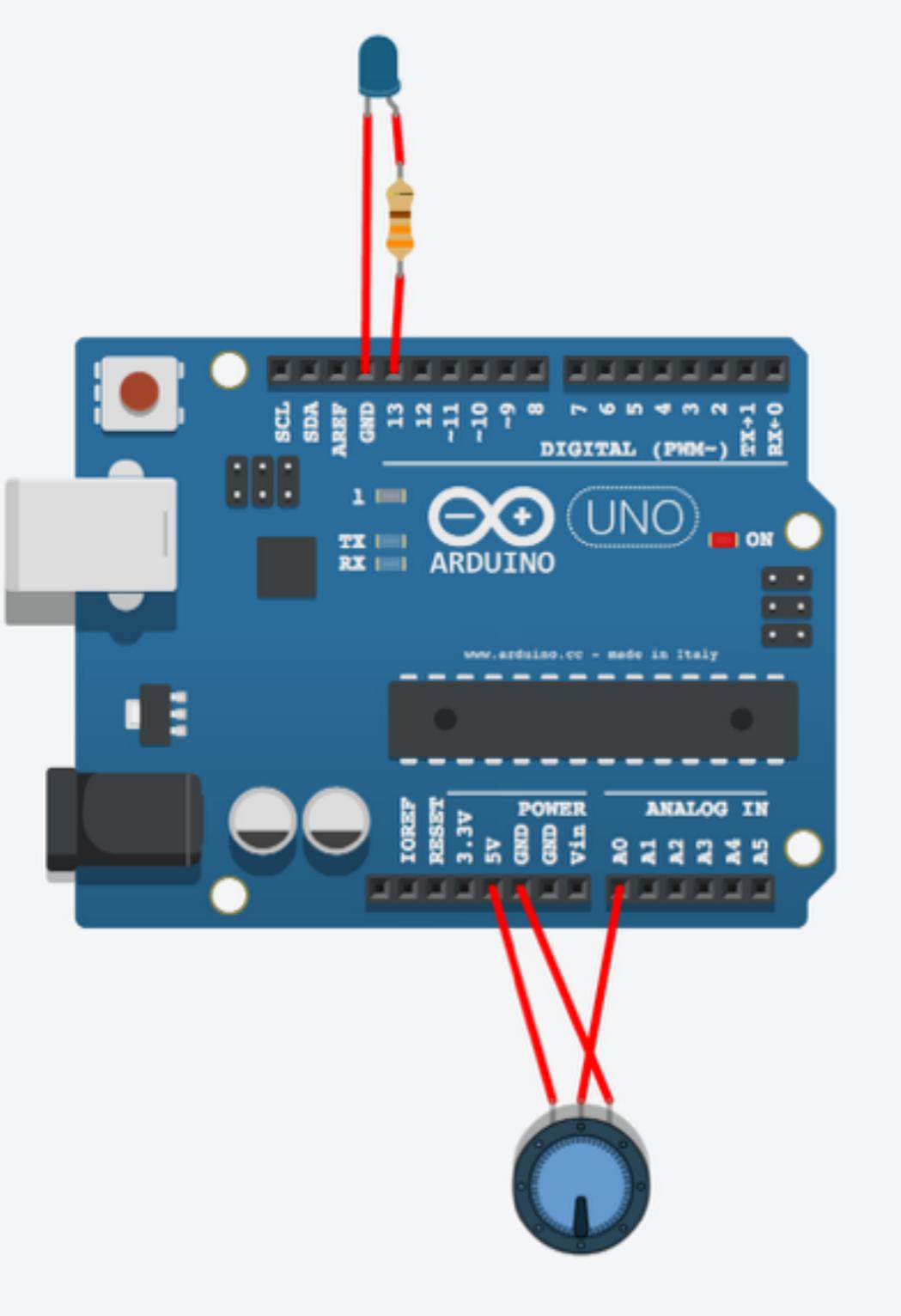
PWM is Based on the on-off interval

PWM

- Change LED intensity
- Drive a stepper motor
- Drive a servo motor
- Drive a speaker



LED Dimmer



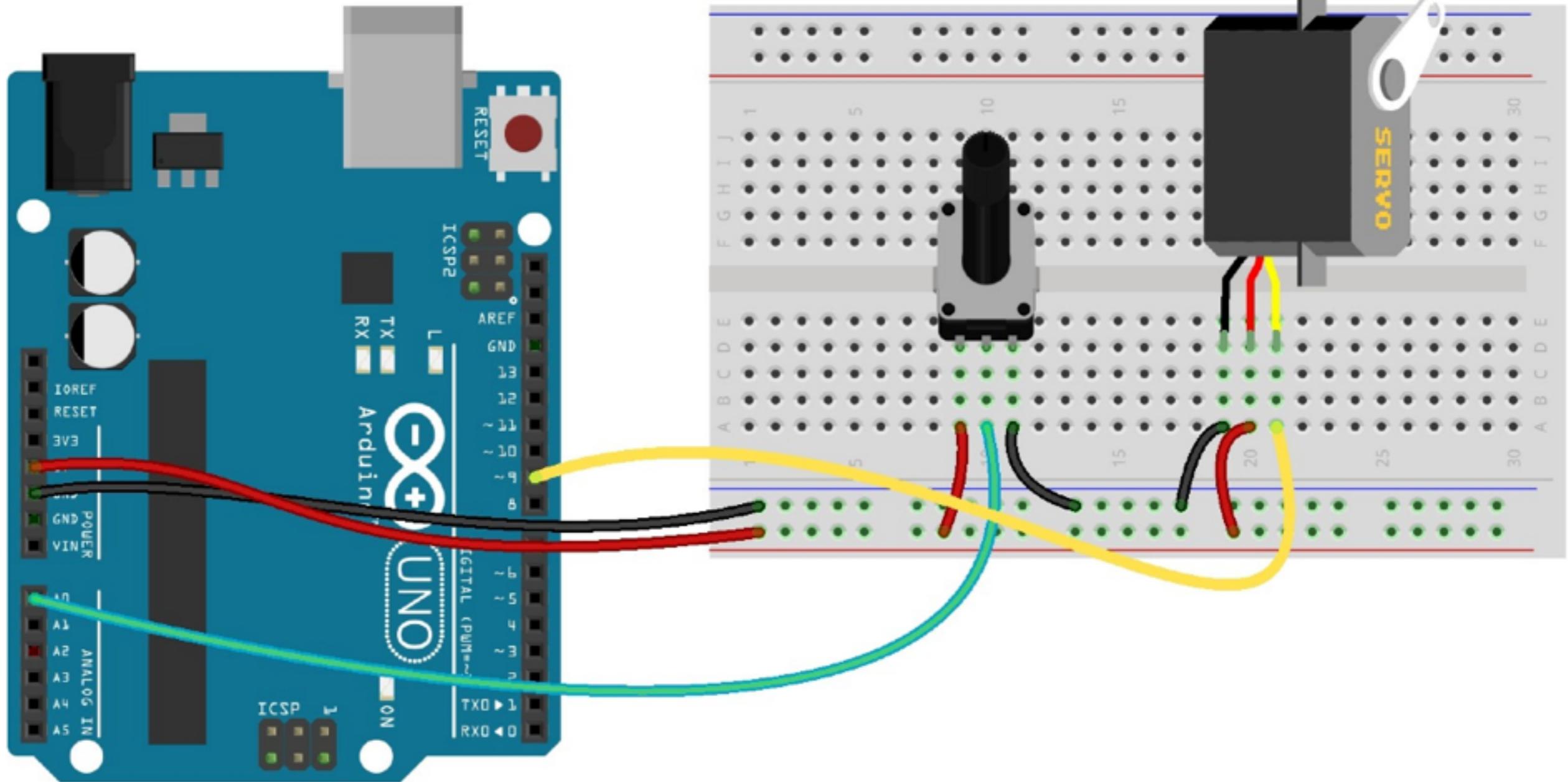
```
int sensorPin = A0; // POT
int ledPin = 13; // LED
int sensorValue = 0; // VALUE

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

void loop() {
    sensorValue = analogRead(sensorPin);

    // input 0-1024 output 0-255
    analogWrite(ledPin, sensorValue/4);

}
```



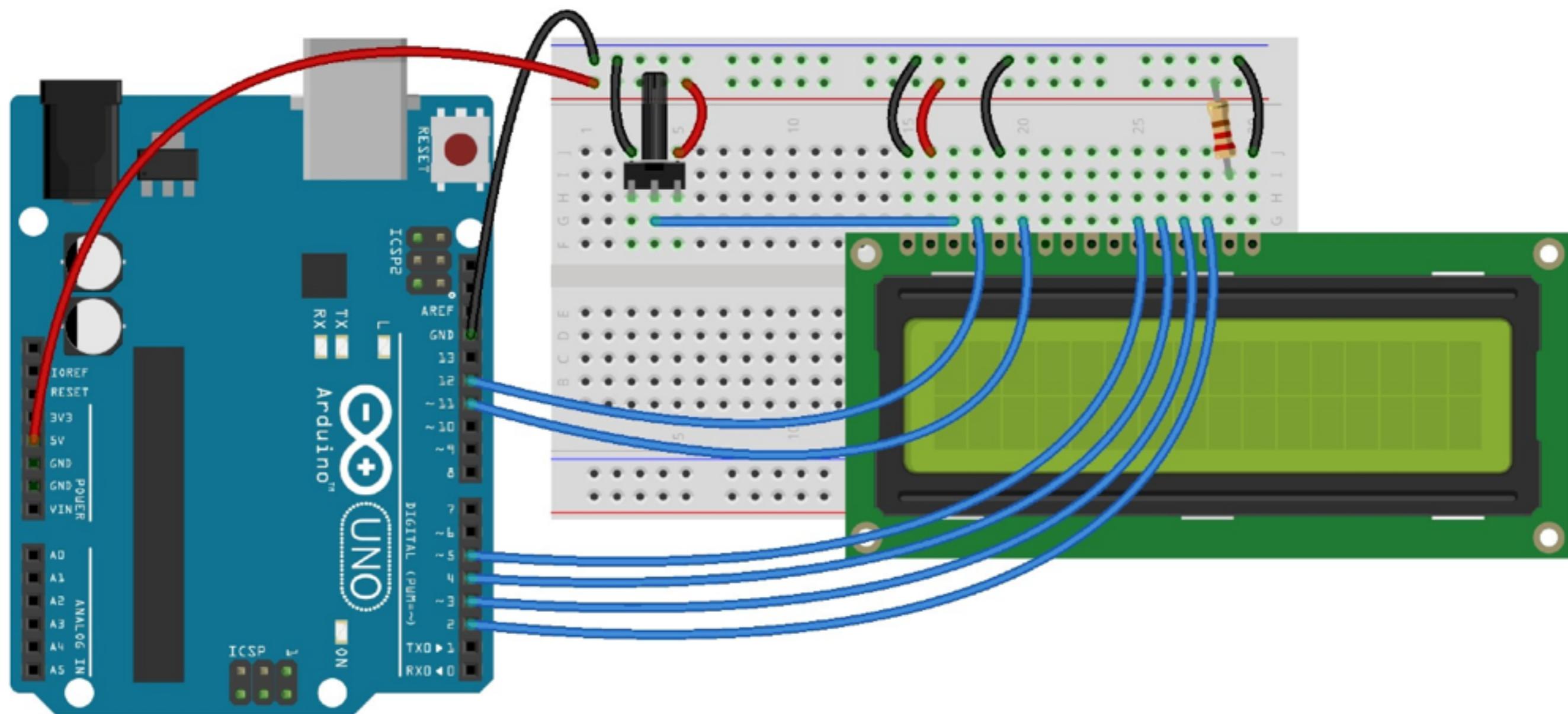
Servo Motor

```
#include <Servo.h>          // Libreria per i servo
#define POTPIN A0             // pin analogico del potenziometro
Servo myservo;              // crea un oggetto Servo (myservo)

int val;                     // variabile del valore letto

void setup()
{
  myservo.attach(9); // setta il pin 9 al servo
}

void loop()
{
  val = analogRead(POTPIN);      // Legge il potenziometro
  val = map(val, 0, 1023, 0, 179); // Scala il valore letto in un
                                  // range 0 ~ 179
  myservo.write(val);           // Muove il servo ad un angolo val
  delay(15);                   // aspetta che il servo si muova
}
```



```
// inserisce la libreria
#include <LiquidCrystal.h>

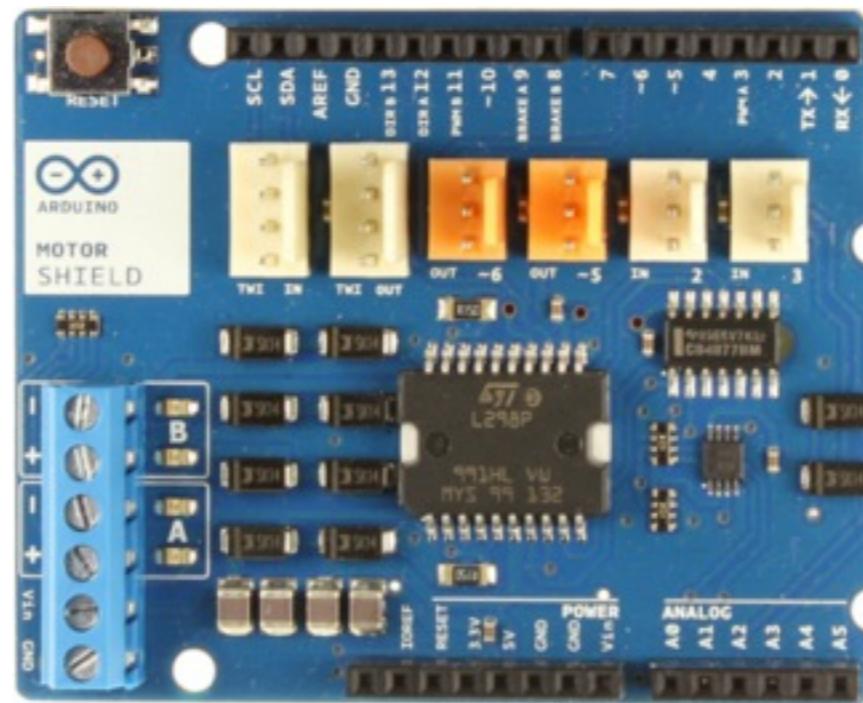
// crea un oggetto lcd inizializzandolo con i pin relativi
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
    // imposta righe e colonne del display
    lcd.begin(16, 2);
    // stampa un messaggio
    lcd.print("hello, world!");
}

void loop() {
    // sposta il cursore alla linea 1, colonna 0
    lcd.setCursor(0, 1);
    // stampa il numero di secondi dall'accensione
    lcd.print(millis()/1000);
}
```



WIFI



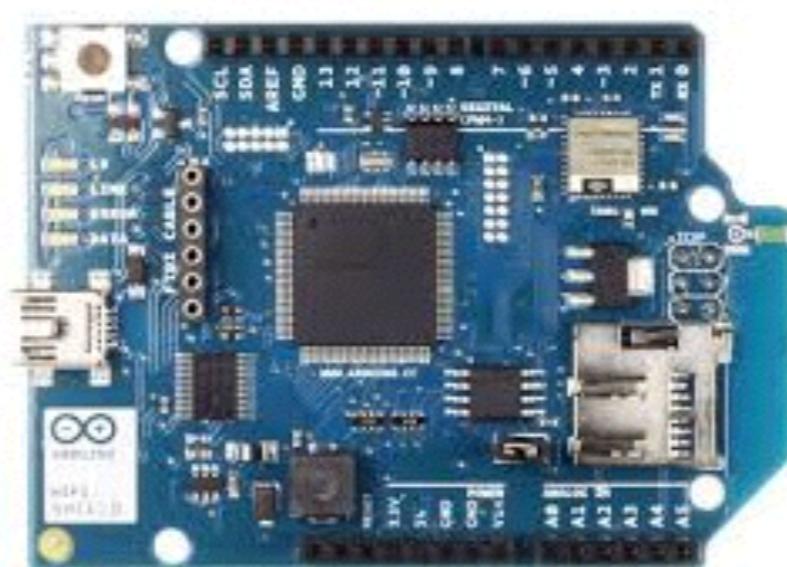
MOTOR



TFT



ETHERNET

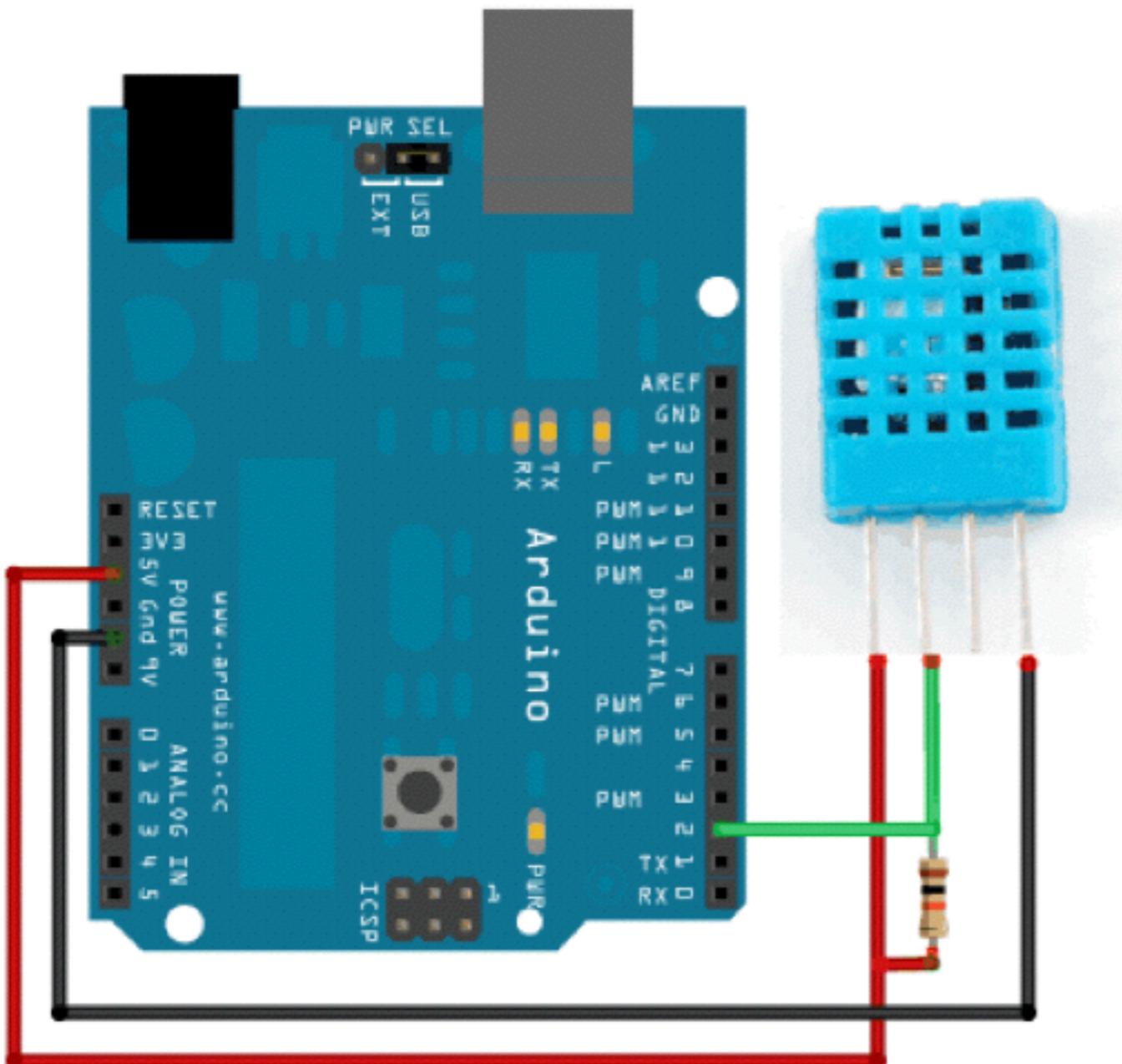


GSM

GPS,
Midi,
Relé
etc

Protocols

- One Wire
- I2C SDA/SCK
- SPI MISO/MOSI/CLK/SS
- Serial TX-RX



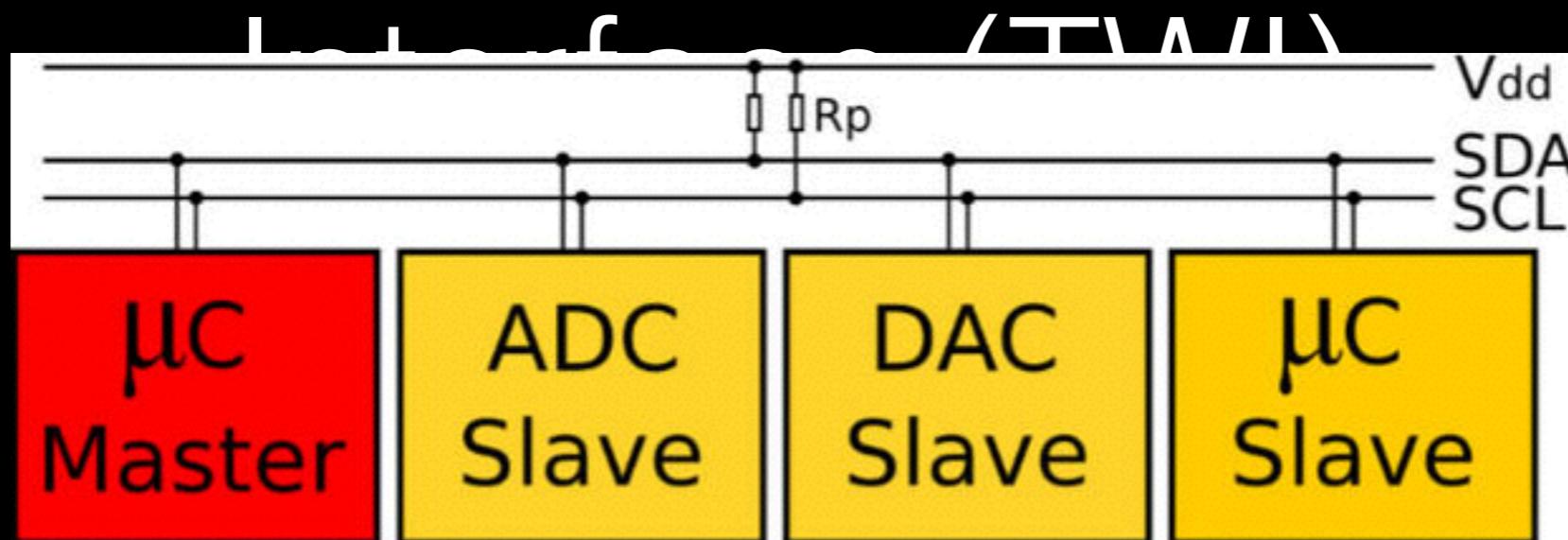
```
#include <dht11.h>

dht11 DHT11;
#define DHT11PIN 2
int chk = DHT11.read(DHT11PIN);

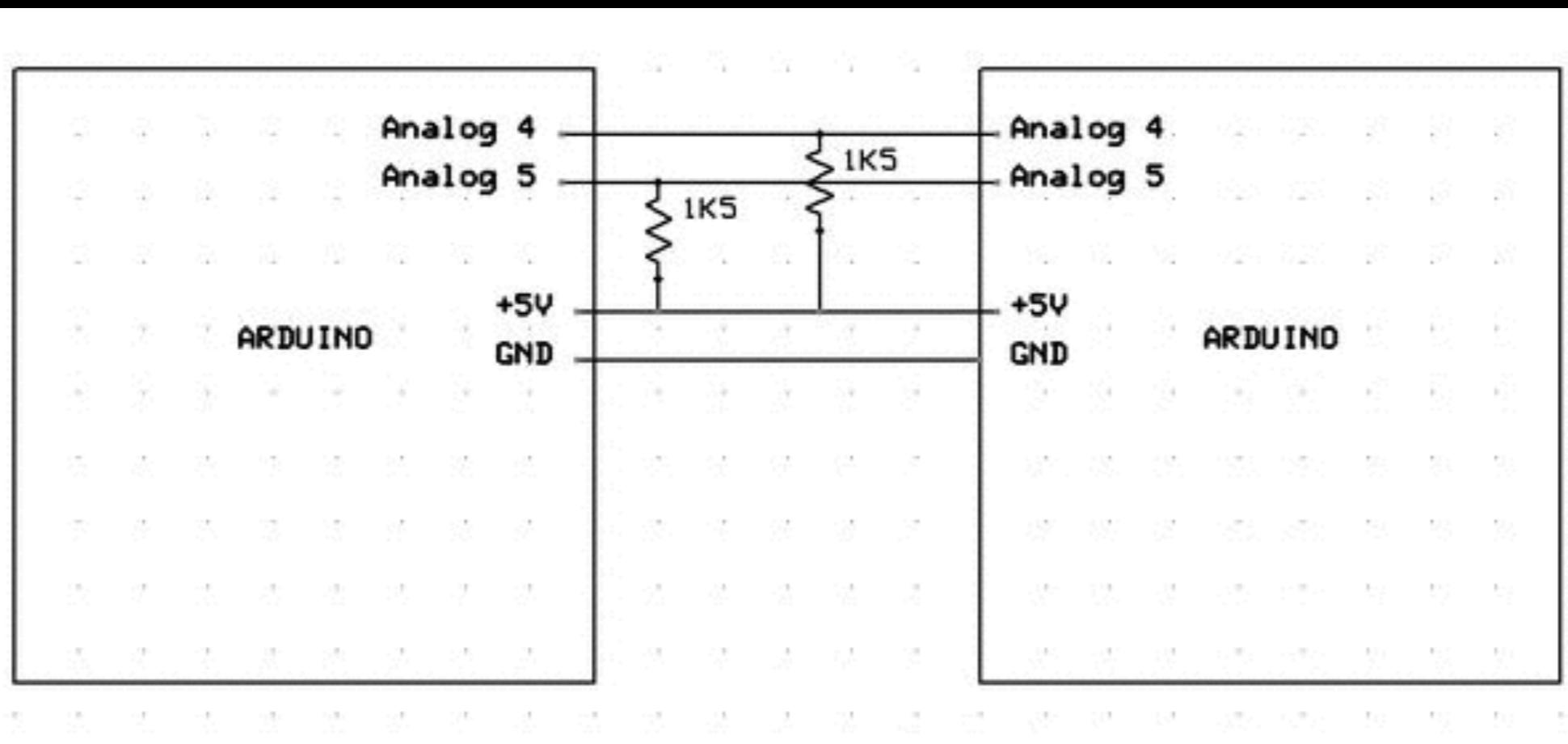
Serial.print("Read sensor: ");
switch (chk)
{
    case DHTLIB_OK:
        Serial.println("OK");
        break;
    case DHTLIB_ERROR_CHECKSUM:
        Serial.println("Checksum error");
        break;
    case DHTLIB_ERROR_TIMEOUT:
        Serial.println("Time out error");
        break;
    default:
        Serial.println("Unknown error");
        break;
}
```

<http://playground.arduino.cc/main/DHT11Lib>

Protocollo Two Wire



- **hardware**



I2C Master

```
#include <Wire.h>

#define LED_PIN 13
byte x = 0;

void setup()
{
    Wire.begin(); // Start I2C Bus as Master
    pinMode(LED_PIN, OUTPUT);
    digitalWrite(LED_PIN, LOW);

}

void loop()
{
    Wire.beginTransmission(9); // transmit to device #9
    Wire.send(x);           // sends x
    Wire.endTransmission(); // stop transmitting
    x++;
    if (x > 5) x=0;
    delay(450);
}
```

I2C Slave

```
#include <Wire.h>

#define LED_PIN 13

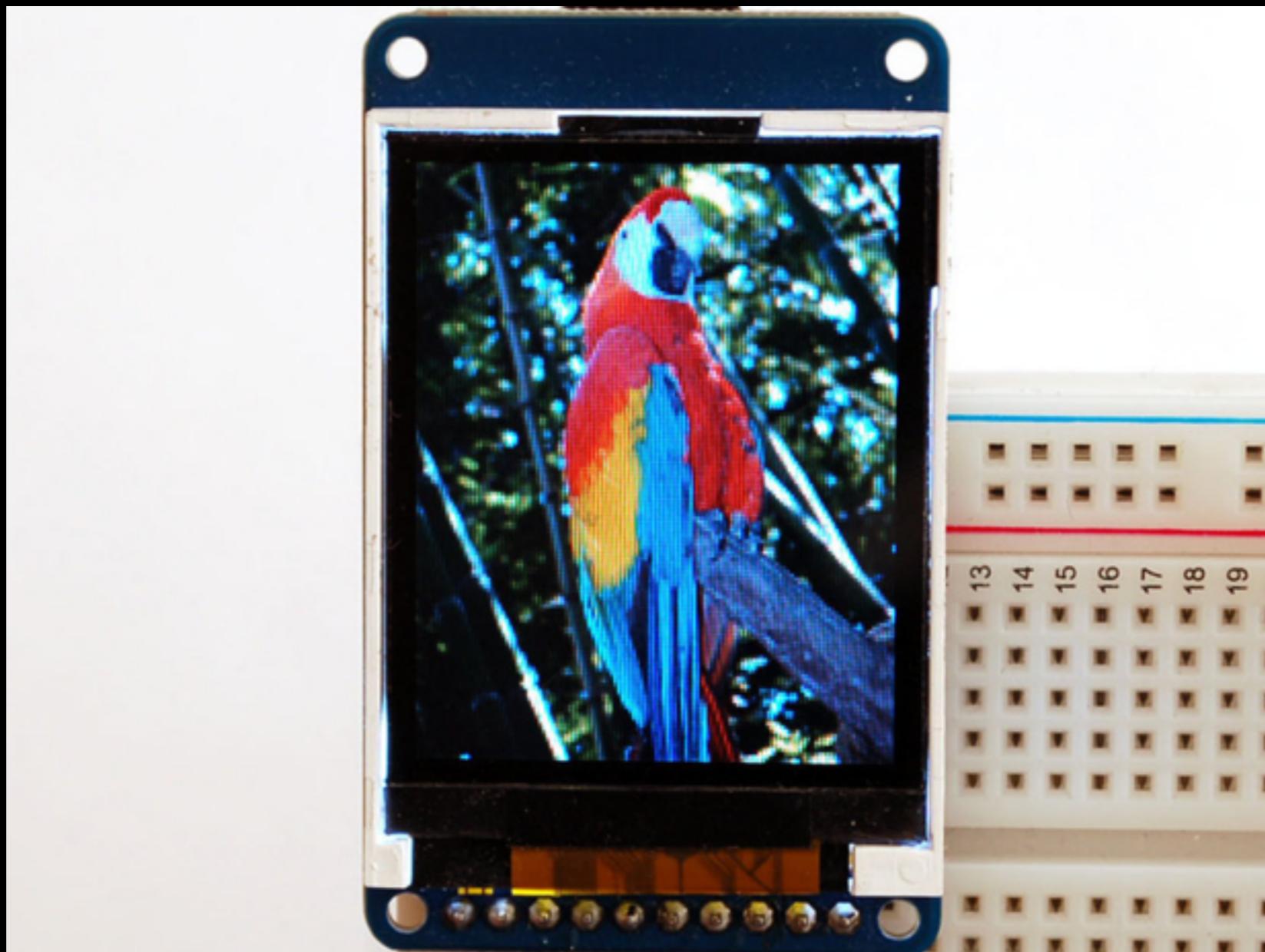
int x;

void setup() {
    Wire.begin(9);          // Start I2C Bus as a Slave (Device Number 9)
    Wire.onReceive(receiveEvent); // register event
    x = 0;
}

void loop() {
    digitalWrite(LED_PIN, HIGH);
    delay(x);
    digitalWrite(LED_PIN, LOW);
    delay(x);
}

void receiveEvent(int howMany) {
    x = Wire.receive(); // receive byte as an integer
}
```

SPI



1.8" SPI TFT display, 160x128 18-bit color

<https://github.com/adafruit/Adafruit-ST7735-Library>

```
#include <Adafruit_GFX.h>      // Core graphics library
#include <Adafruit_ST7735.h> // Hardware-specific library
#include <SPI.h>

#define TFT_CS      10
#define TFT_RST    9
#define TFT_DC     8

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS,  TFT_DC,  TFT_RST);

void setup(){
  tft.initR();
}

void loop(){

  tft.fillScreen(ST7735_BLACK);

  tft.fillRoundRect(25, 10, 78, 60, 8, ST7735_WHITE);

  tft.fillTriangle(42, 20, 42, 60, 90, 40, ST7735_RED);

}
```

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

<http://github.com/adafruit>