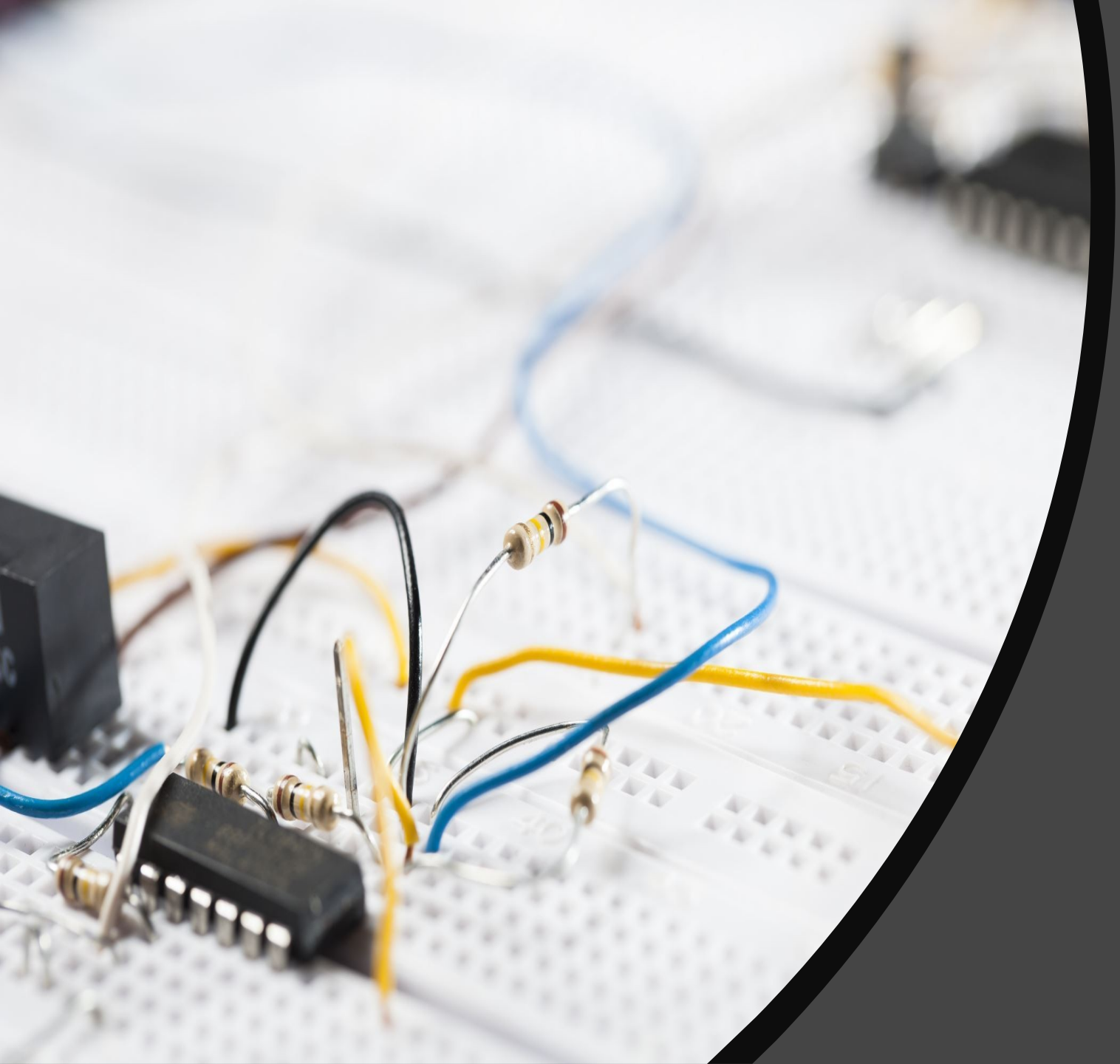


Arduino

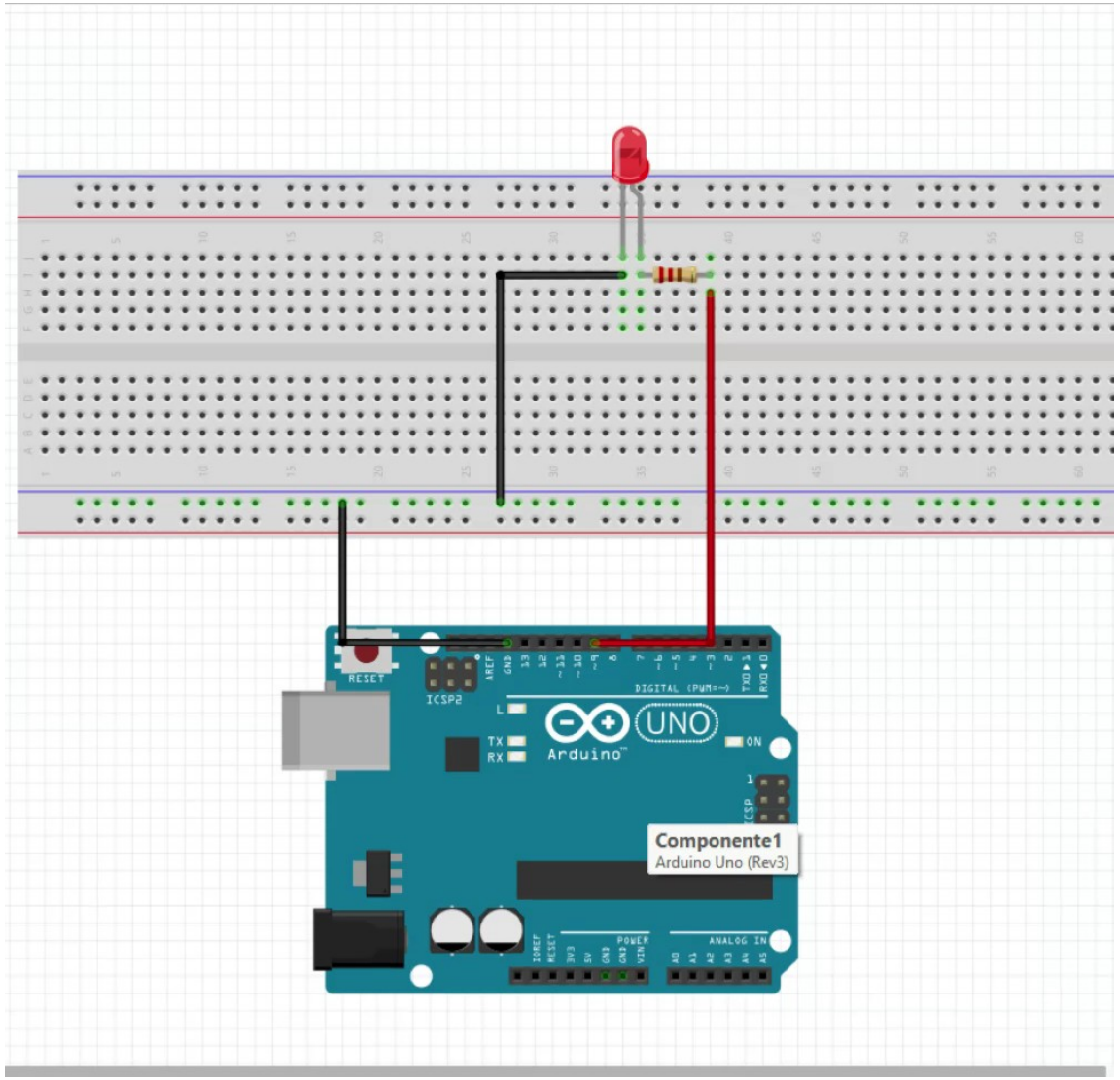
Fatto da Valentino Hu

classe 1A

Anno 2021/2022

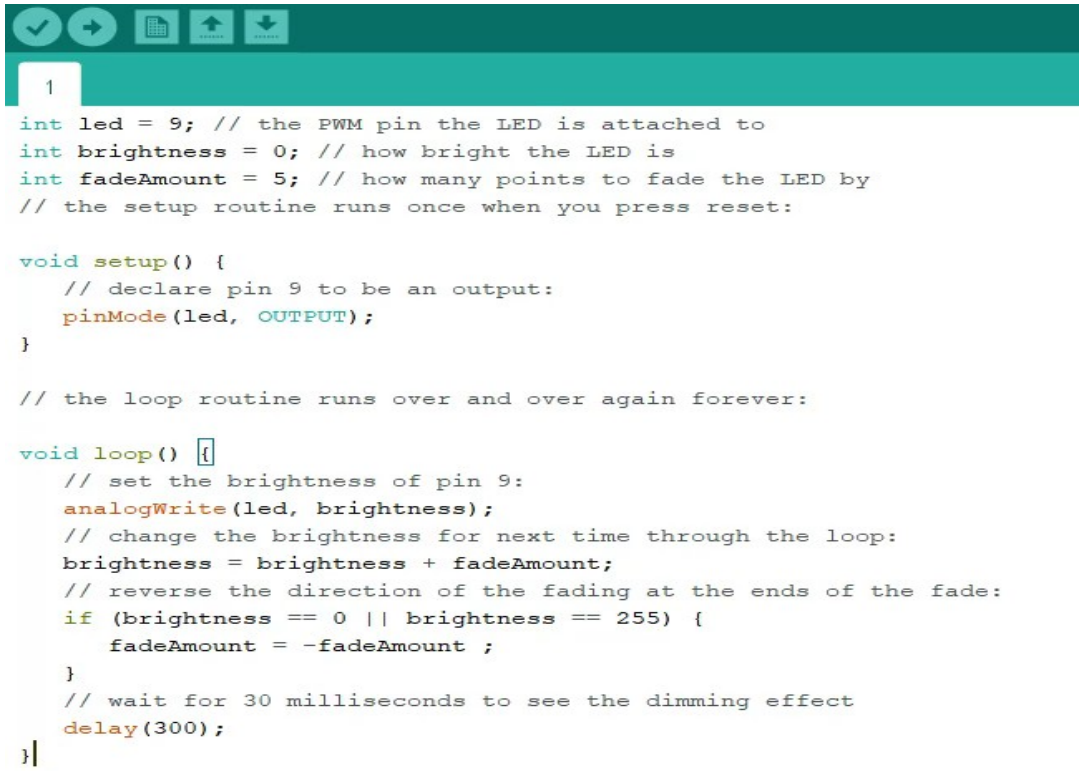
Primo laboratorio: primo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un led rosso
- Una resistenza da 330 Ohm



Codice primo circuito

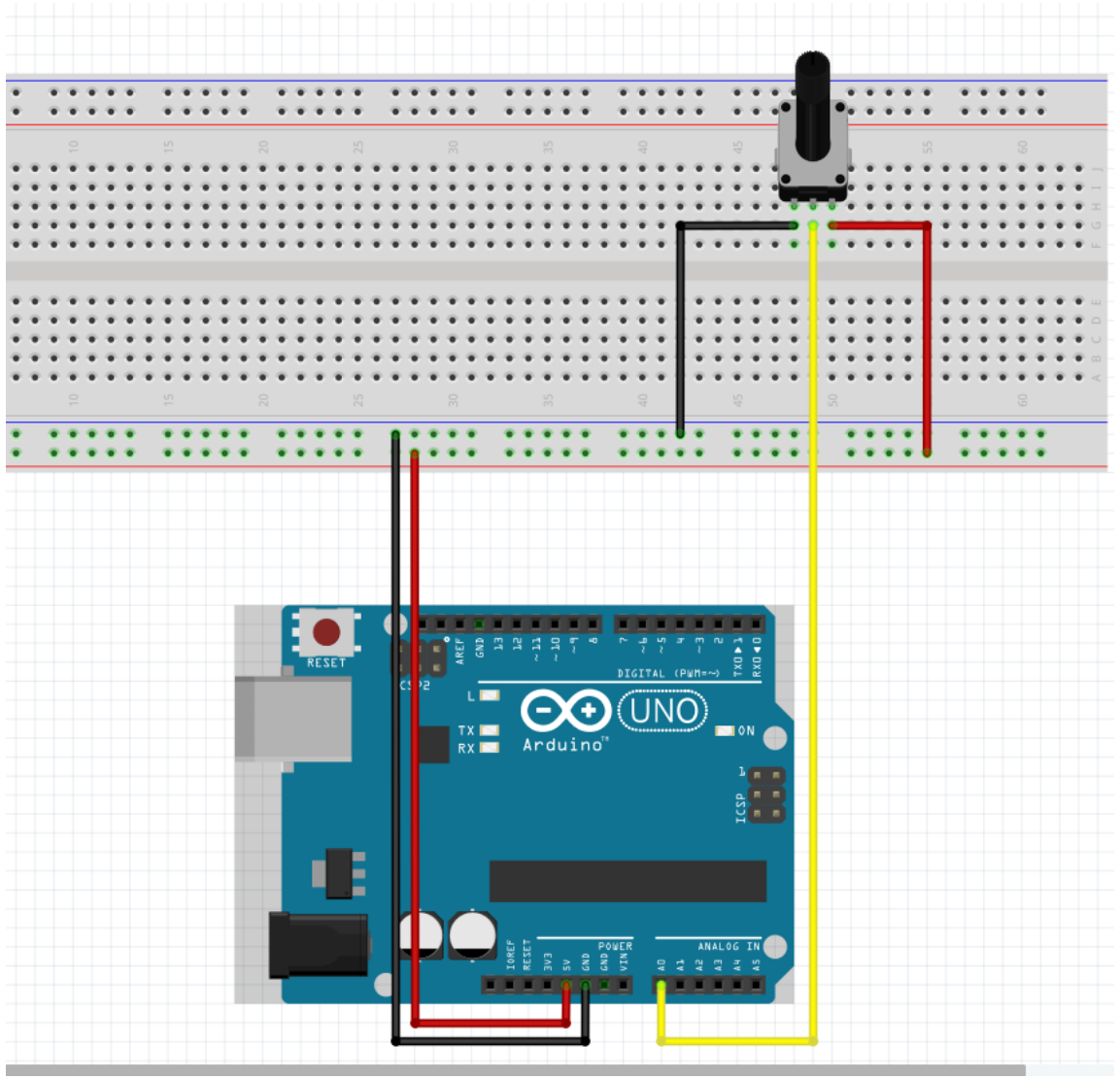
Con questo codice il led dovrebbe man mano aumentare di luminosità, arrivare al massimo e poi diminuire sfruttando un ciclo for



```
int led = 9; // the PWM pin the LED is attached to
int brightness = 0; // how bright the LED is
int fadeAmount = 5; // how many points to fade the LED by
// the setup routine runs once when you press reset:

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(300);
}
```

Primo laboratorio: secondo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un resistore variabile 5k (potenziometro)



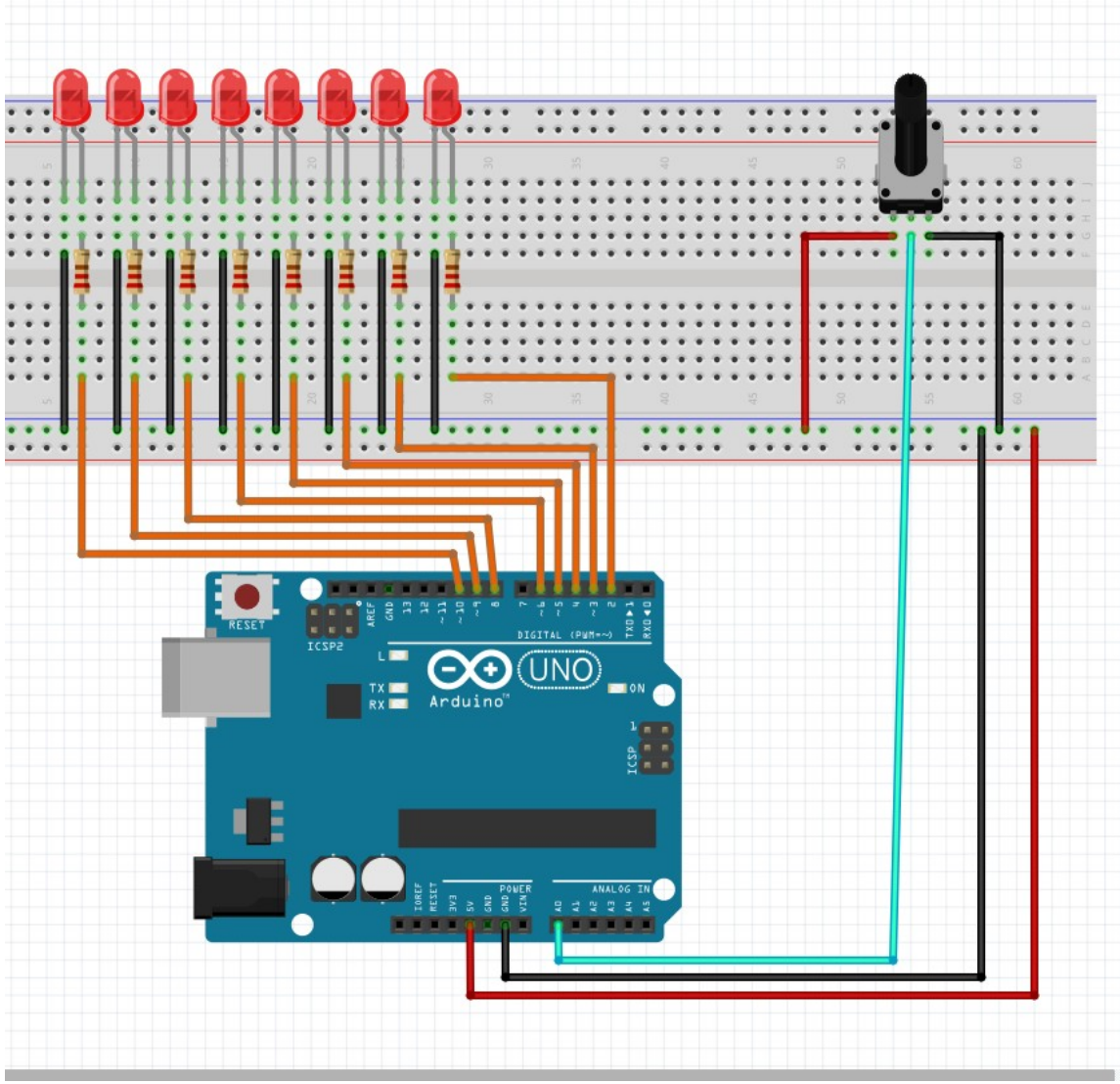
Codice secondo circuito

Con questo codice andando su IDE e cliccando in alto a destra su monitor seriale, uscirà uno schermo dove vi farà vedere il voltaggio letto dal PIN. La tensione aumenta e diminuisce girando il potenziometro

```
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);
}
```

Primo laboratorio: terzo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un resistore variabile 5k (potenziometro)
- Otto led rossi
- Otto resistenze da 330 ohm



Codice terzo circuito

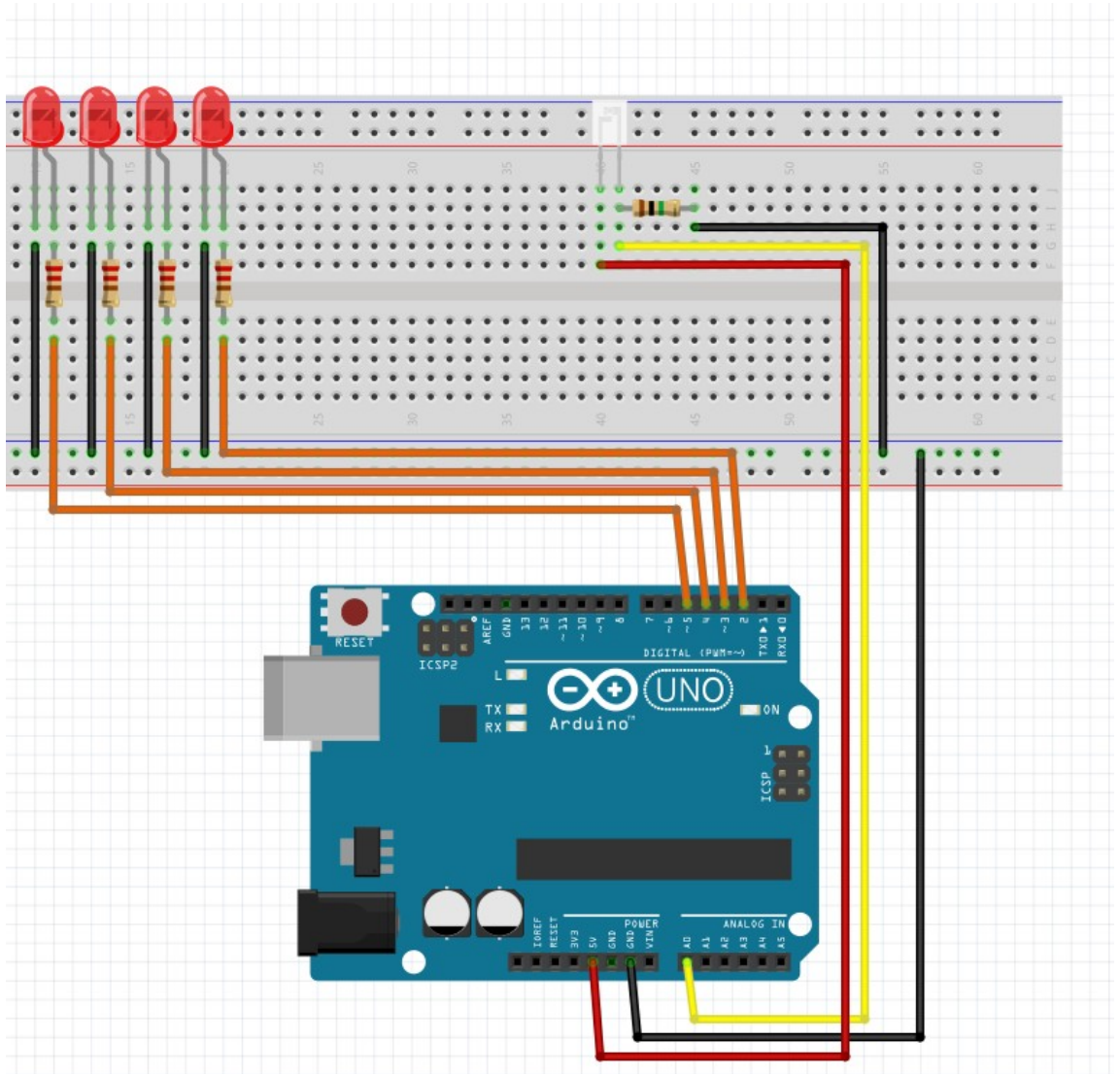
Con questo codice girando il potenziometro i led si accenderanno uno dopo l'altro in proportzione alla tensione sentita dal PIN

sketch_nov18a §

```
const int analogPin = A0; // the pin that the potentiometer is attached tc
const int ledCount = 8; // the number of LEDs in the bar graph
int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9}; // an array of pin numbers to which LEDs are attached

void setup() {
   // loop over the pin array and set them all to output:
   for (int thisLed = 0; thisLed < ledCount; thisLed++) {
      pinMode(ledPins[thisLed], OUTPUT);
   }
}

void loop() {
   // read the potentiometer:
   int sensorReading = analogRead(analogPin);
   // map the result to a range from 0 to the number of LEDs:
   int ledLevel = map(sensorReading, 0, 1023, 0, ledCount);
   // loop over the LED array:
   for (int thisLed = 0; thisLed < ledCount; thisLed++) {
      // if the array element's index is less than ledLevel,
      // turn the pin for this element on:
      if (thisLed < ledLevel) {
         digitalWrite(ledPins[thisLed], HIGH);
      }else { // turn off all pins higher than the ledLevel:
         digitalWrite(ledPins[thisLed], LCW);
      }
   }
}
```

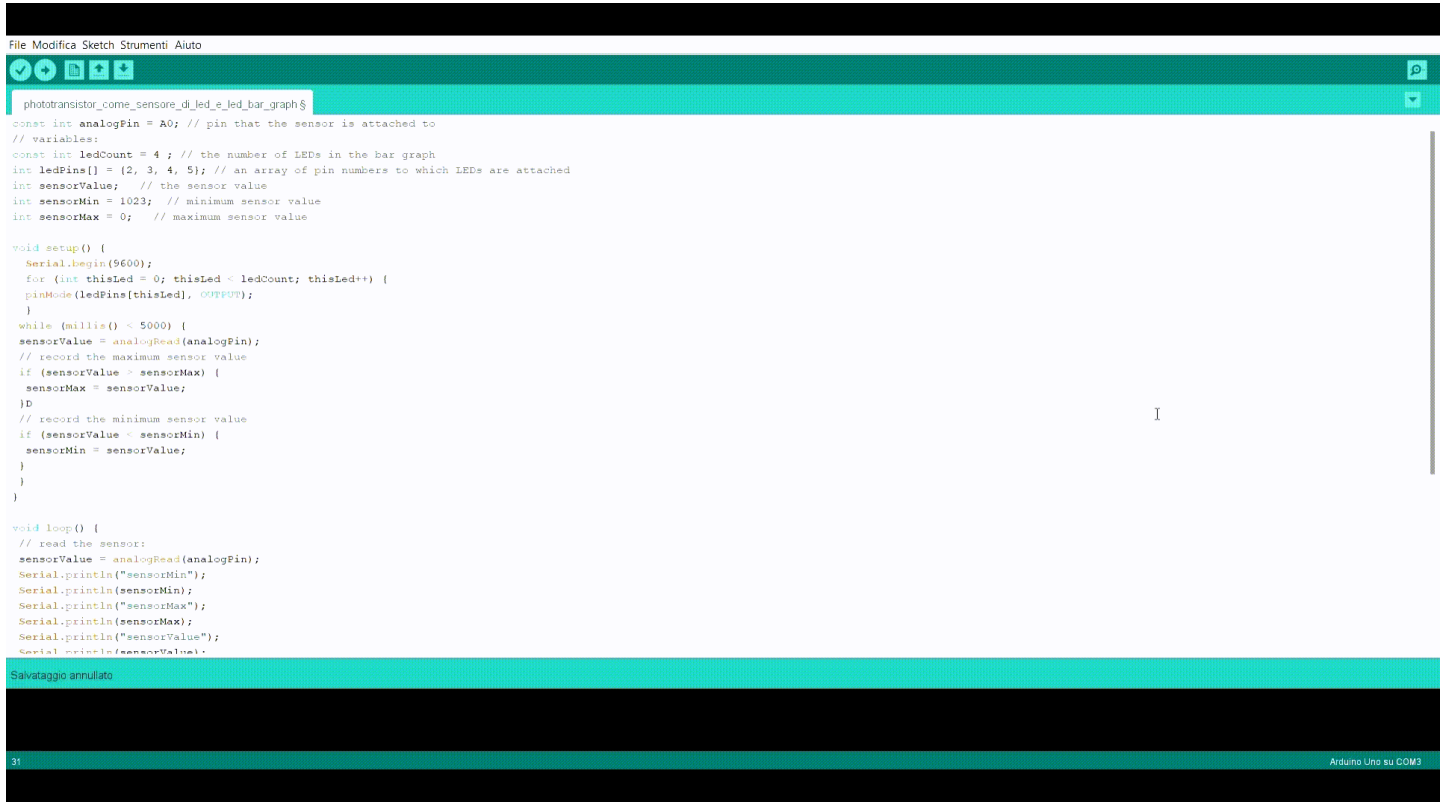
secondo laboratorio: primo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Resistenza da 1M ohm
- Quattro led rossi
- Quattro resistenze da 330 ohm
- phototransistor



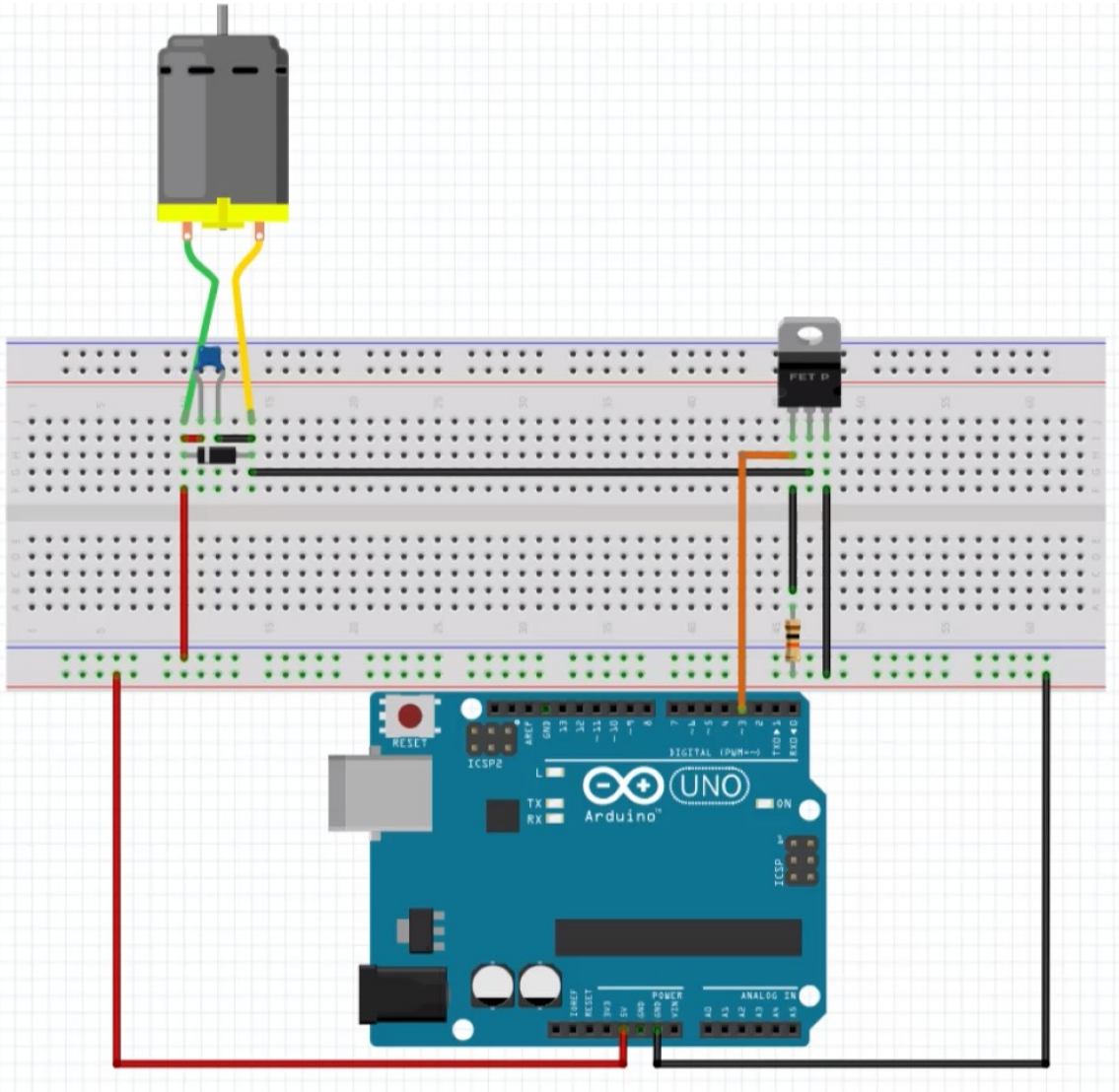
Codice primo circuito

Con questo codice dopo la fase di calibrazione e la scelta della resistenza adatta, più si fa buio e più LED si accendono



Terzo laboratorio: primo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un DC motor
- Un diodo
- Un mosfet
- Un condensatore
- Resistenza



Codice primo circuito

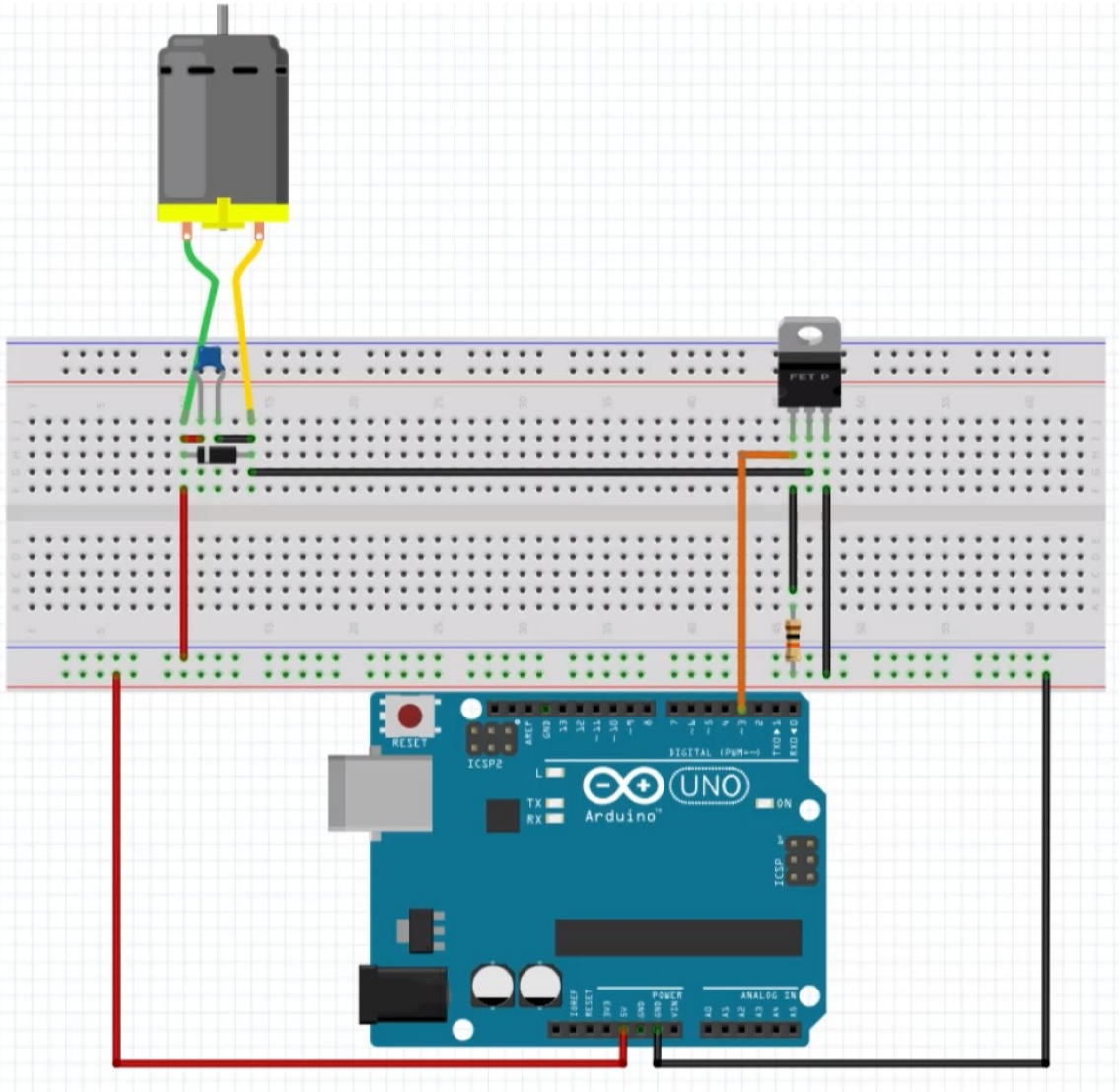
Con questo codice il motere si azionerà e si fermerà a intermittenza per intervalli di un secondo, ripetendosi all'infinito

motore_a_secondi

```
int pinMot = 3;


void setup() {
// put your setup code here, to run once:
pinMode(pinMot,OUTPUT);

}

void loop() {
// put your main code here, to run repeatedly:
//analogWrite(pinMot,0);
//delay(1000);
digitalWrite(pinMot,HIGH);
delay(1000);
digitalWrite(pinMot,LOW);
delay(1000);



}
```

Terzo laboratorio: secondo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un DC motor
- Un diodo
- Un mosfet
- Un condensatore
- Una resistenza



Codice secondo circuito

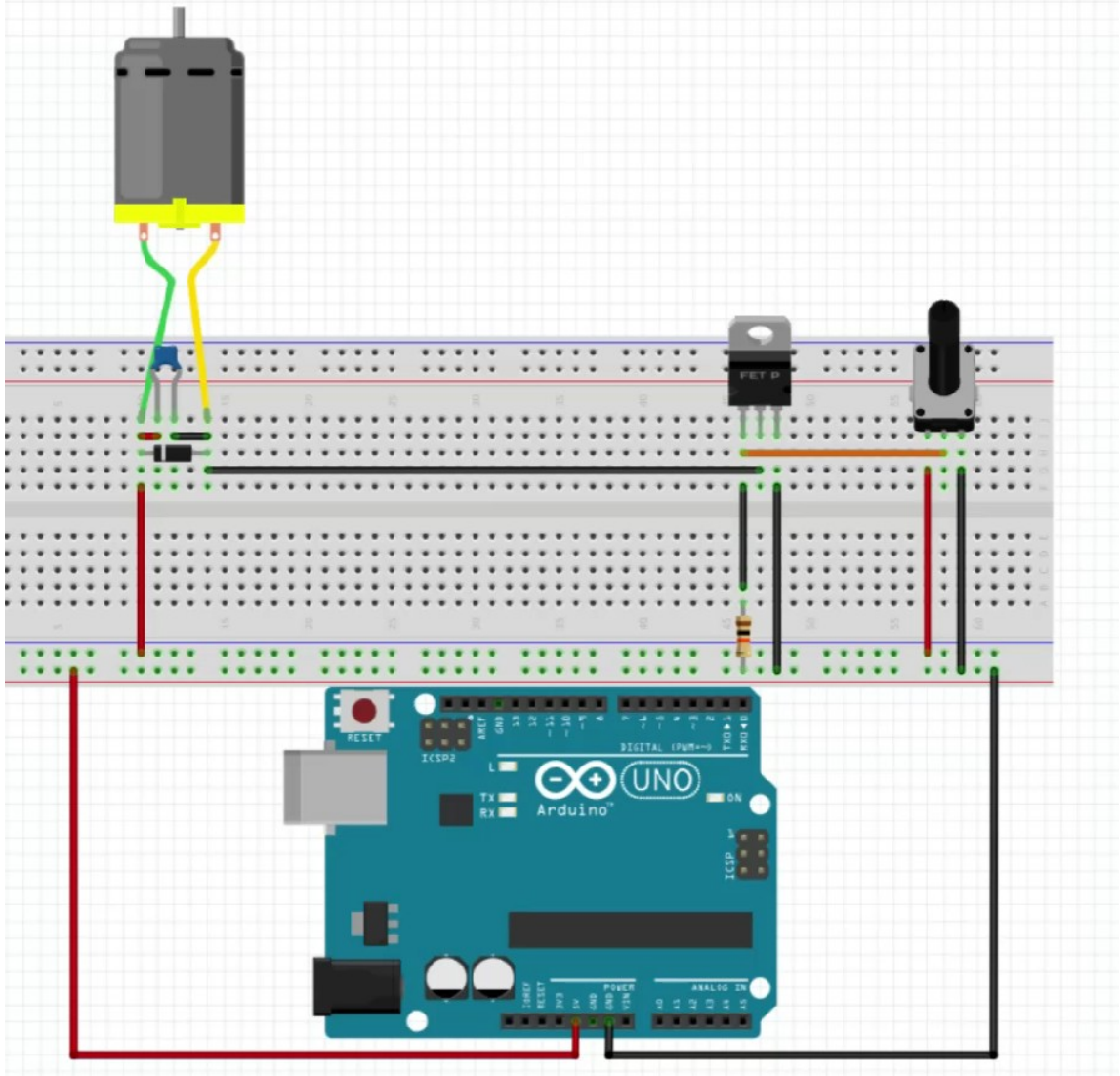
Con questo codice entrando su monitor seriale possiamo decidere la velocità del motere mettendo un numero tra 0 e 255

motore_cambio_veloct_

```
int motorPin = 3;
void setup()
{
pinMode(motorPin, OUTPUT);
Serial.begin(9600);
while (! Serial);
Serial.println("Speed 0 to 255");
}

void loop()
{
if (Serial.available())
{
int speed = Serial.parseInt();
if (speed >= 0 && speed <= 255)
{
analogWrite(motorPin, speed);
}
delay(5000);
}
}
```

Terzo laboratorio: terzo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un DC motor
- Un diodo
- Un mosfet
- Un condensatore
- Un resistore variabile 5k (potenziometro)
- Una resistenza

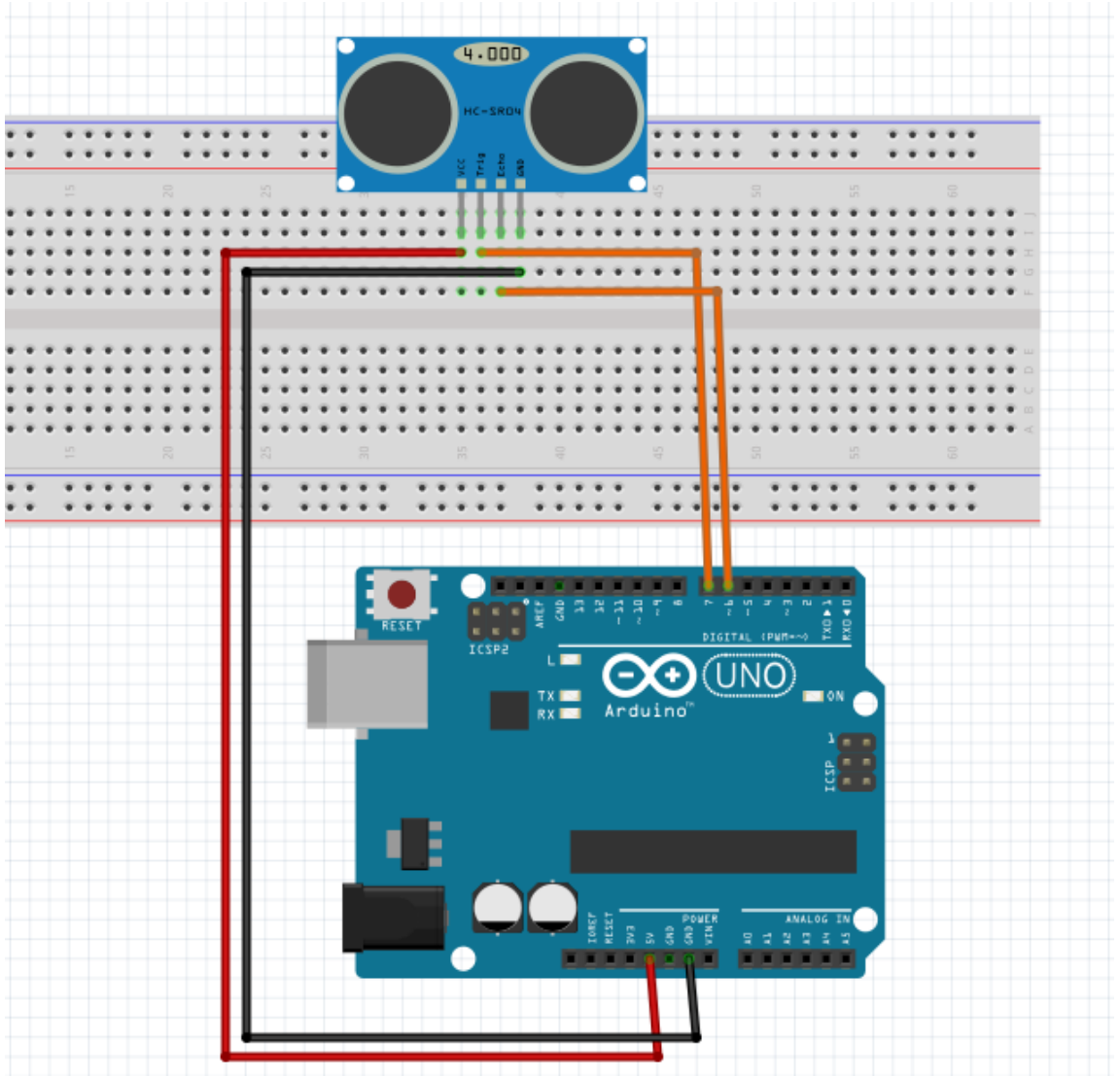


Nel terzo circuito non serve Il codice

Questo circuito girando il potenziometro il motore comincierà ad aumentare la velocità

Quarto laboratorio: primo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un sensore di distanza(ultrasonic sensor)



Codice primo circuito

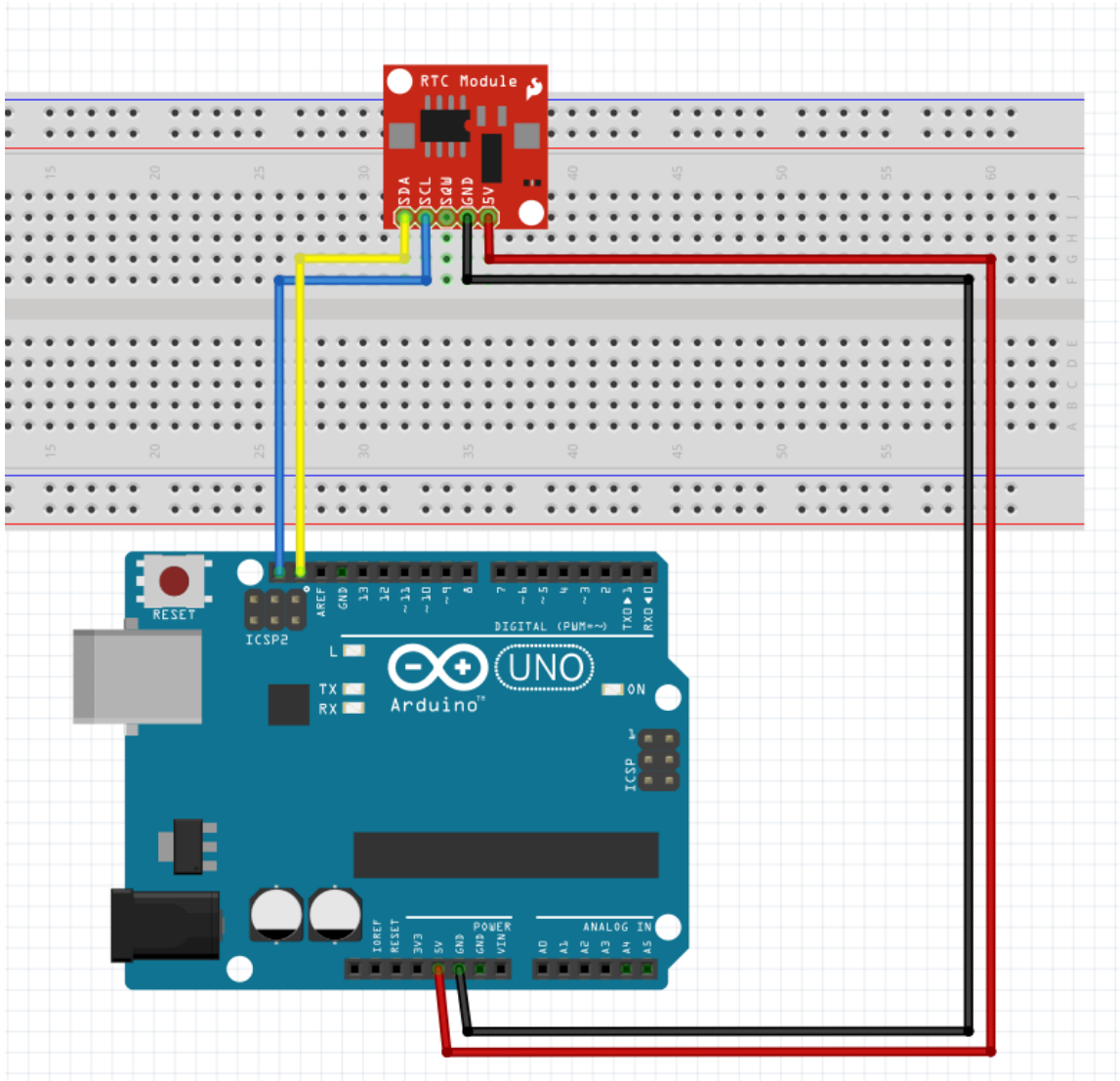
Andando su monitor seriale ci farà vedere le distanze in cm

ultrasonic_sensor

```
const int pingPin = 7; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 6; // Echo Pin of Ultrasonic Sensor

void setup() {
   Serial.begin(9600); // Starting Serial Terminal
}

void loop() {
   long duration, cm;
   pinMode(pingPin, OUTPUT);
   digitalWrite(pingPin, LOW);
   delayMicroseconds(2);
   digitalWrite(pingPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(pingPin, LOW);
   pinMode(echoPin, INPUT);
   duration = pulseIn(echoPin, HIGH);
   cm = microsecondsToCentimeters(duration);
   Serial.print(cm);
   Serial.print("cm");
   Serial.println();
   delay(100);
}

long microsecondsToCentimeters(long microseconds) {
   return microseconds / 29 / 2;
}
```

Quarto laboratorio: secondo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un rtc(real time clock) module



Codice secondo circuito

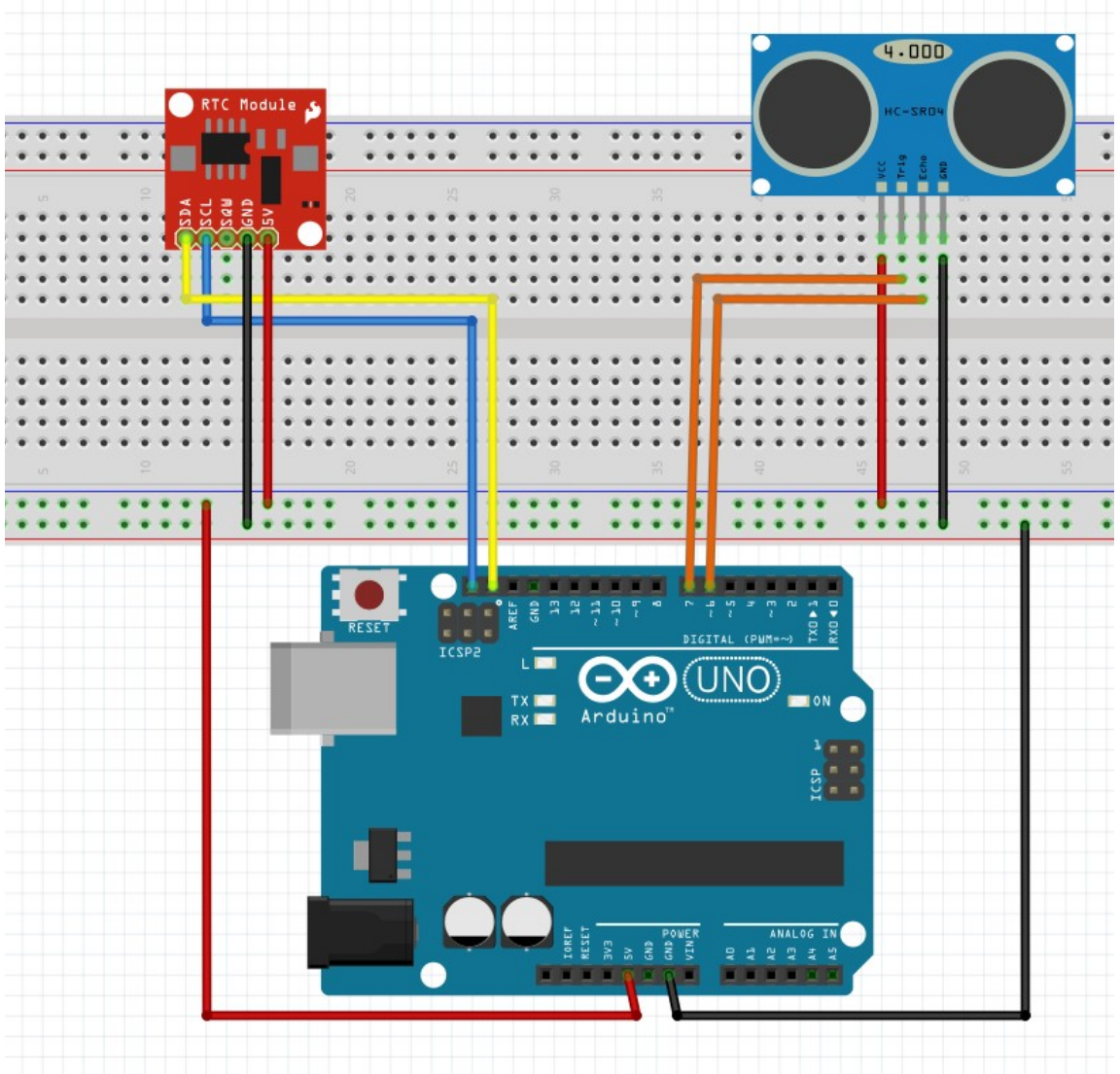
Mettendo questo codice impostiamo data e ora e successivamente li si potrà vedere sul monitor seriale.

Per usare questo codice bisogna scaricare la libreria "ds3231FS"

```
#include <Wire.h>
#include <ds3231.h>
struct ts t;
void setup() {
Serial.begin(9600);
Wire.begin();
t.hour=11;
t.min=12;
t.sec=0;
t.mday=06;
t.mon=03;
t.year=2022;
DS3231_set(t);
}
void loop() {
DS3231_get(&t);
Serial.print("");
Serial.print(t.mday);
Serial.print(",");
Serial.print(t.mon);
Serial.print(",");
Serial.print(t.year);
Serial.print(",");
Serial.print(t.hour);
Serial.print(",");
Serial.print(t.min);
Serial.print(",");
Serial.println(t.sec);
delay(1000);
```

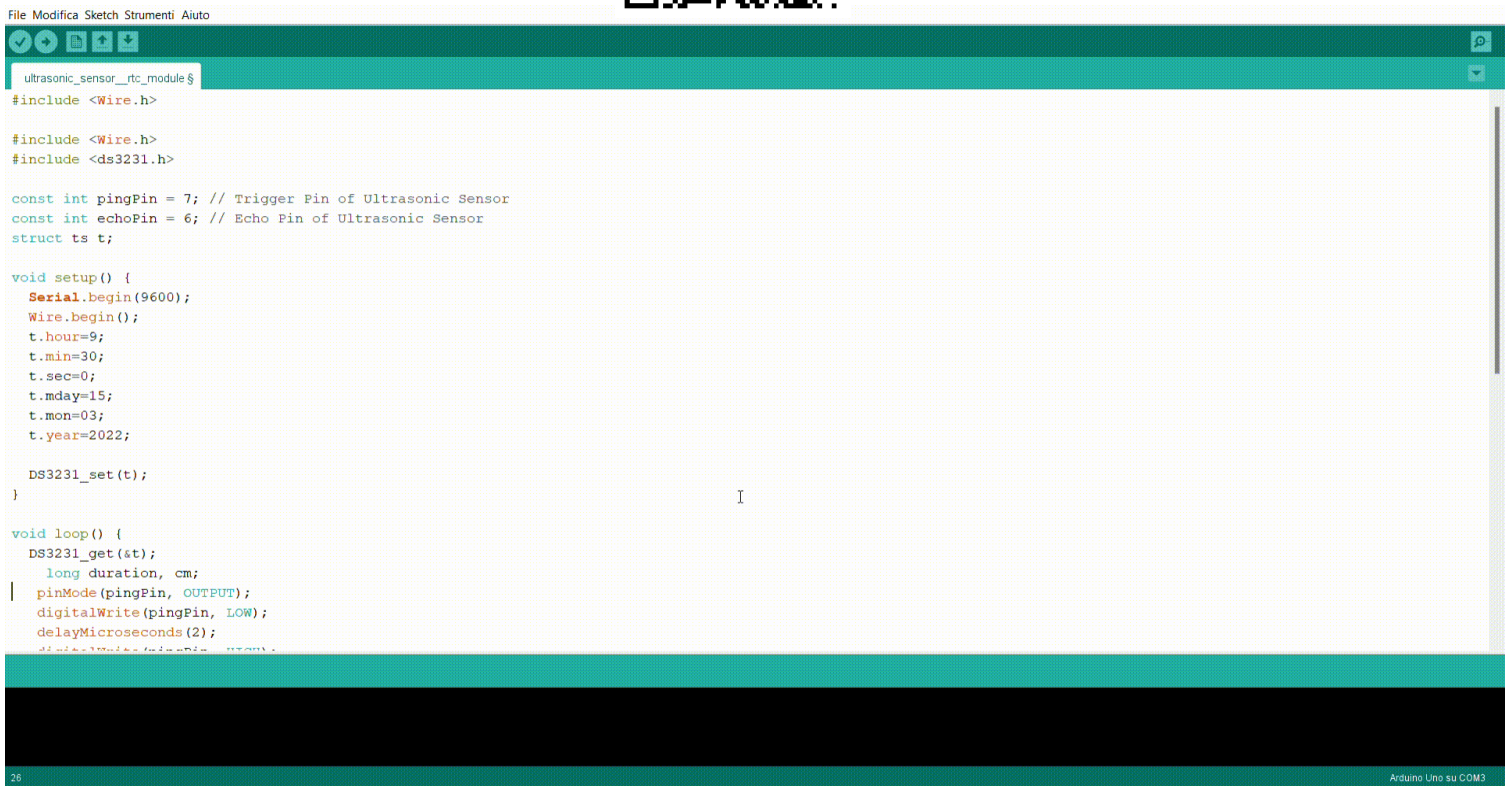
Quarto laboratorio: terzo circuito

- Componenti:
- Una Breadboard
- Un Arduino Uno
- Un sensore di distanza(ultrasonic sensor)
- Un rtc(real time clock) module



Codice terzo circuito

Questo codice è l'unione dei due codici precedenti



Grazie per la visione

Fine