Input e Output in Arduino

Programmazione di sistemi multicore

Michele Martinelli
Michele.martinelli@uniroma1.it

Marco Bernardi
m.bernardi@uniroma1.it

Comunicazione Seriale in Arduino

- **Serial.begin(speed)** – Imposta la velocità dei dati in bit al secondo (baud) per la trasmissione seriale dei dati

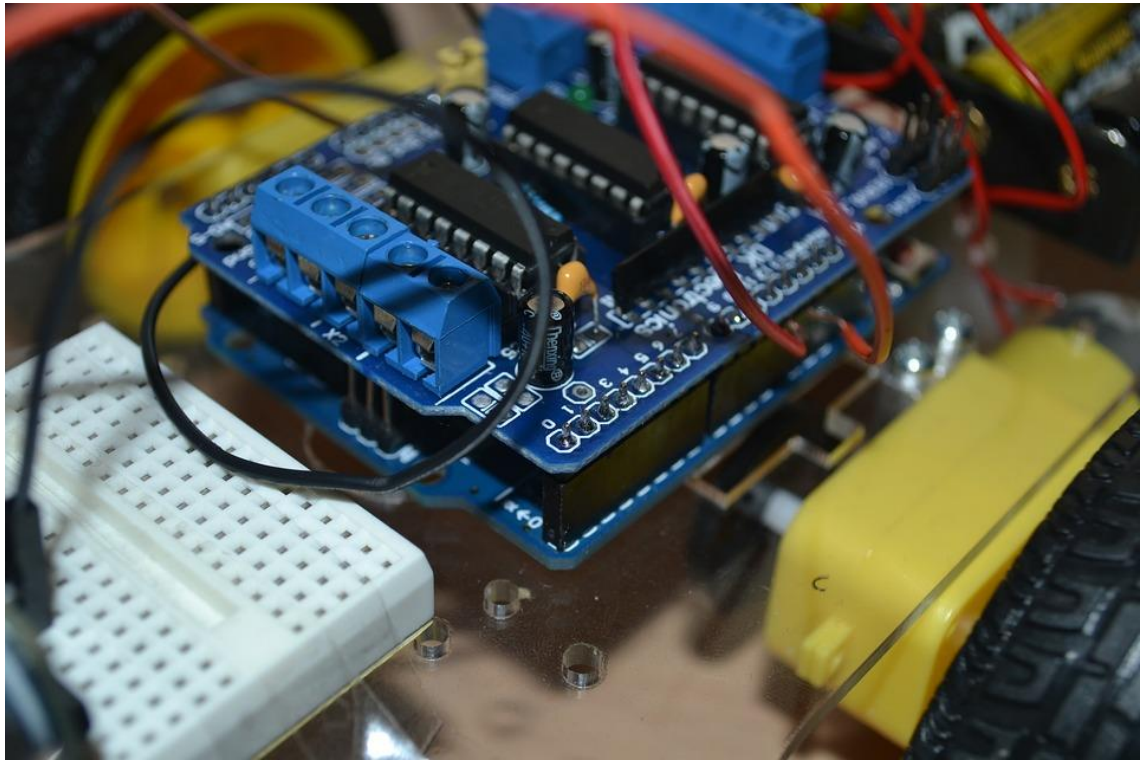
  Ex: Serial.begin(9600); // imposta la comunicazione a 9600 bit per secondo

- **Serial.println(sensorValue)** – Stampa in output (sul monitor seriale) il valore *sensorValue*

  Ex: value = analogRead(A0);

  Serial.println(value);

Analogic Read Serial

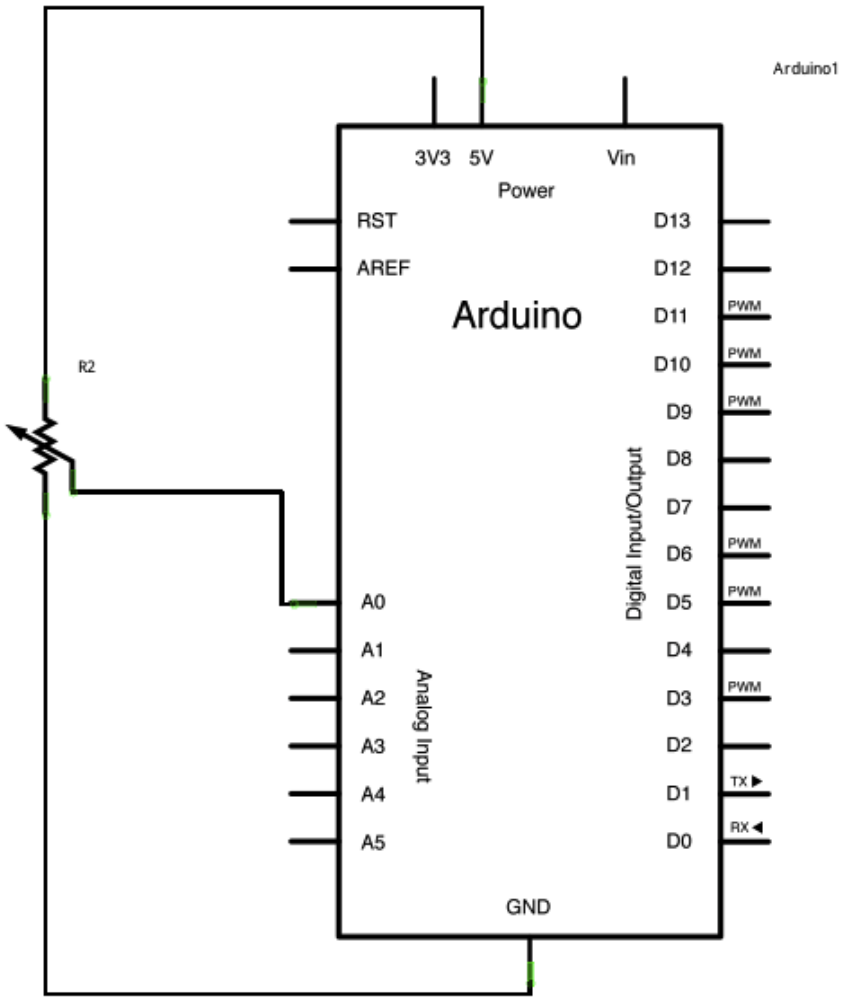


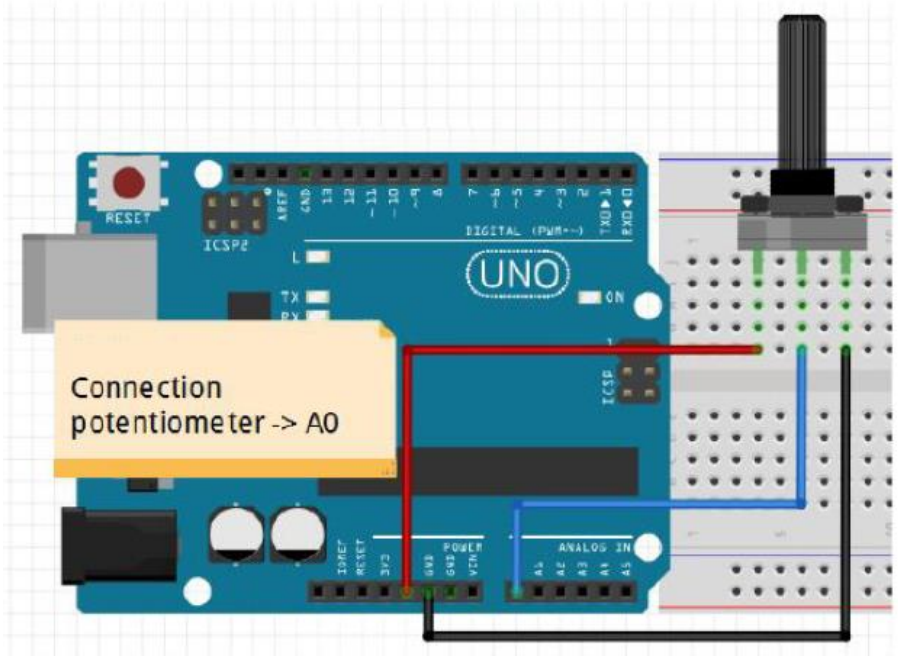
In questo esempio utilizziamo un resistore variabile (un potenziometro), ne leggeremo il valore usando un ingresso analogico di una scheda Arduino e cambieremo la frequenza di lampeggiamento del LED incorporato.

Hardware Richiesto:

- Potenziometro
- BreadBoard e cavi
- Arduino
- Cavo USB

Analogic Read Serial

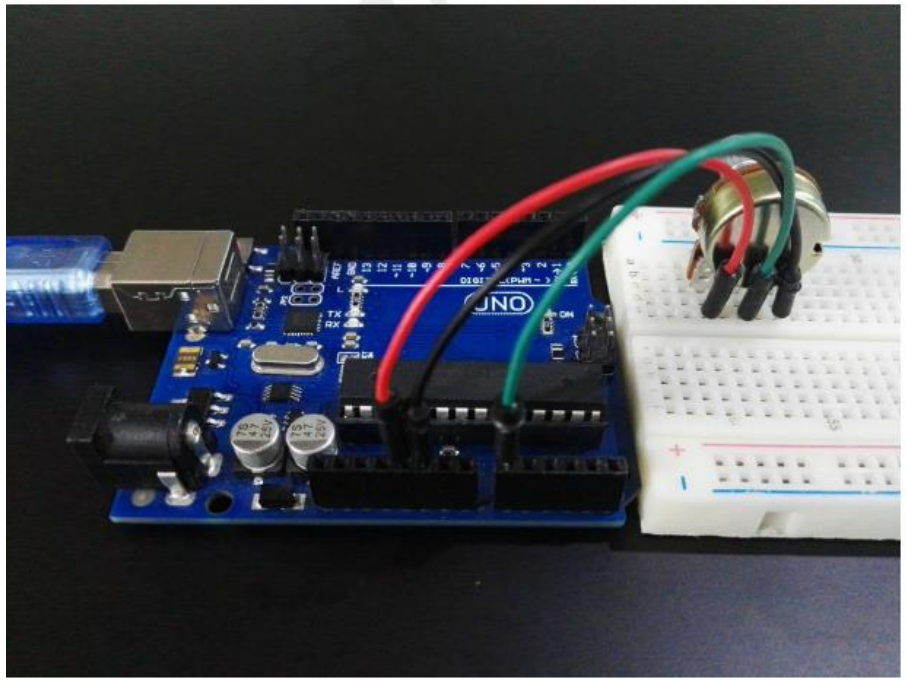




Step:

Collegare tre fili alla scheda Arduino. Il primo va dal Ground ad uno dei pin esterni del potenziometro. Il secondo collega il pin 5V con il pin esterno del potenziometro. Il terzo va dall'ingresso analogico 0 al pin centrale del potenziometro.

Circuito e Programma



```
// the setup routine runs once when you press reset:
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);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

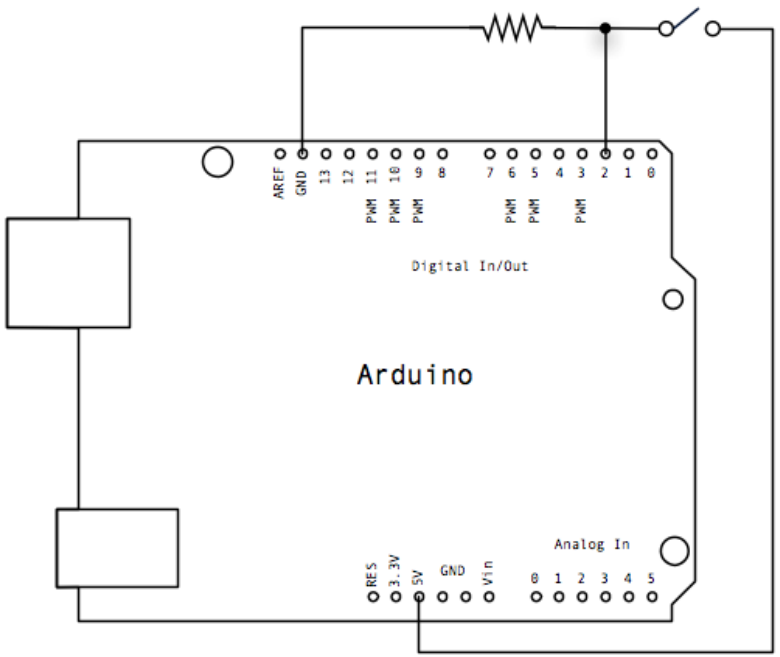
Push the button

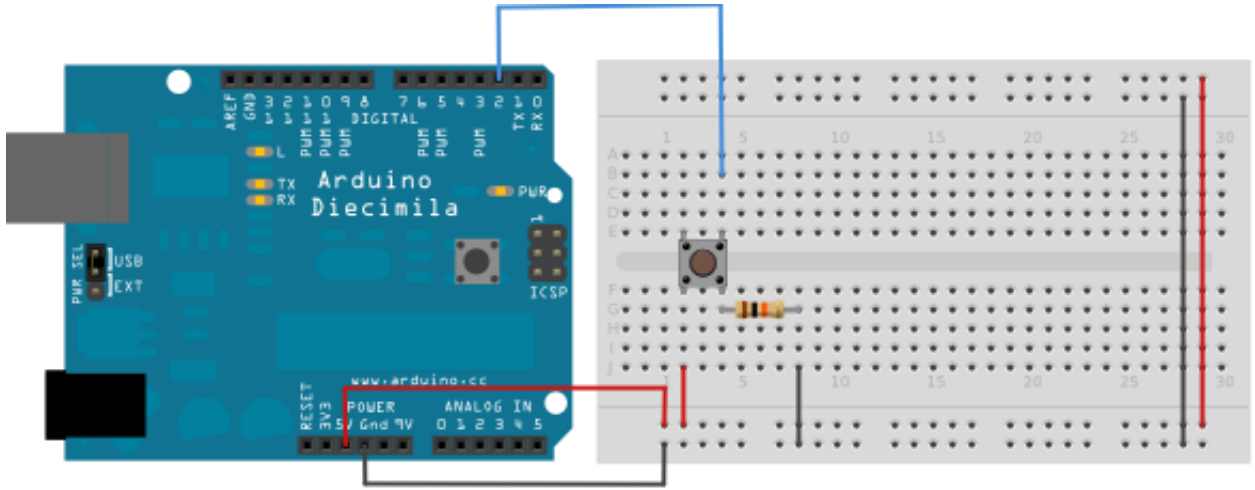
Scopo di questo esercizio è la progettazione di un circuito per l'illuminazione di un led a seguito della pressione di un pulsante utilizzando ARDUINO.

Hardware Richiesto:

- 1 Switch o pulsante
- Resistenza da 10KΩ
- BreadBoard e cavi
- Arduino
- Cavo USB

Push the button: Costruire il circuito

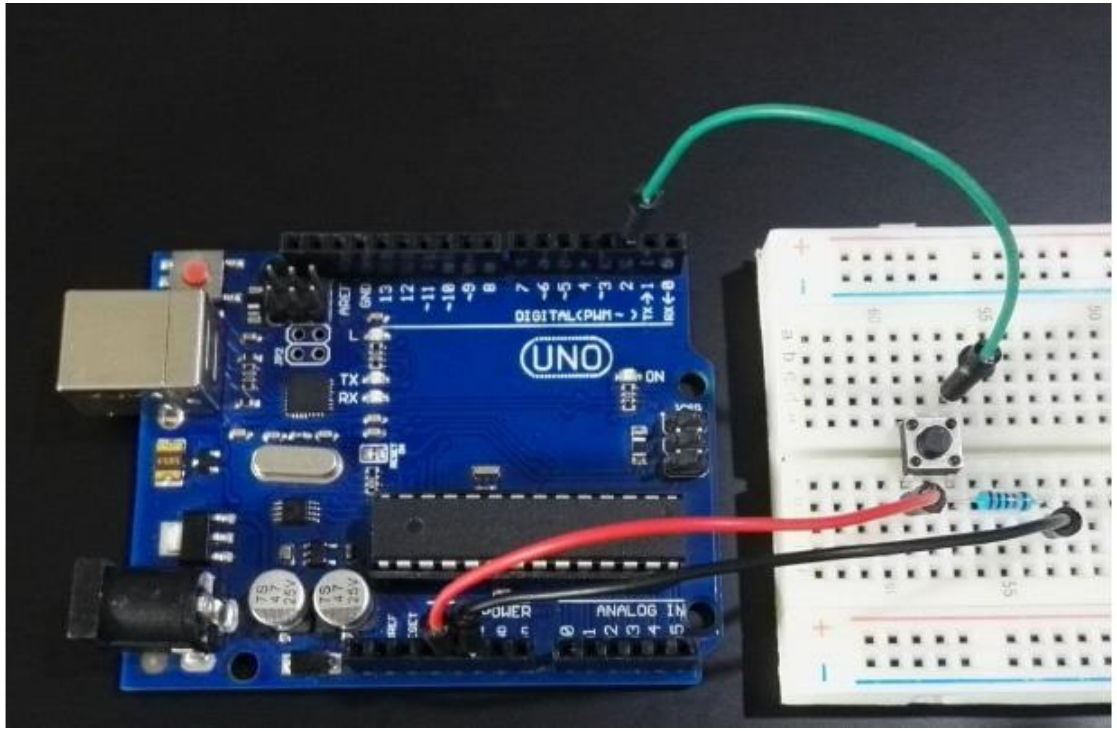




Step:

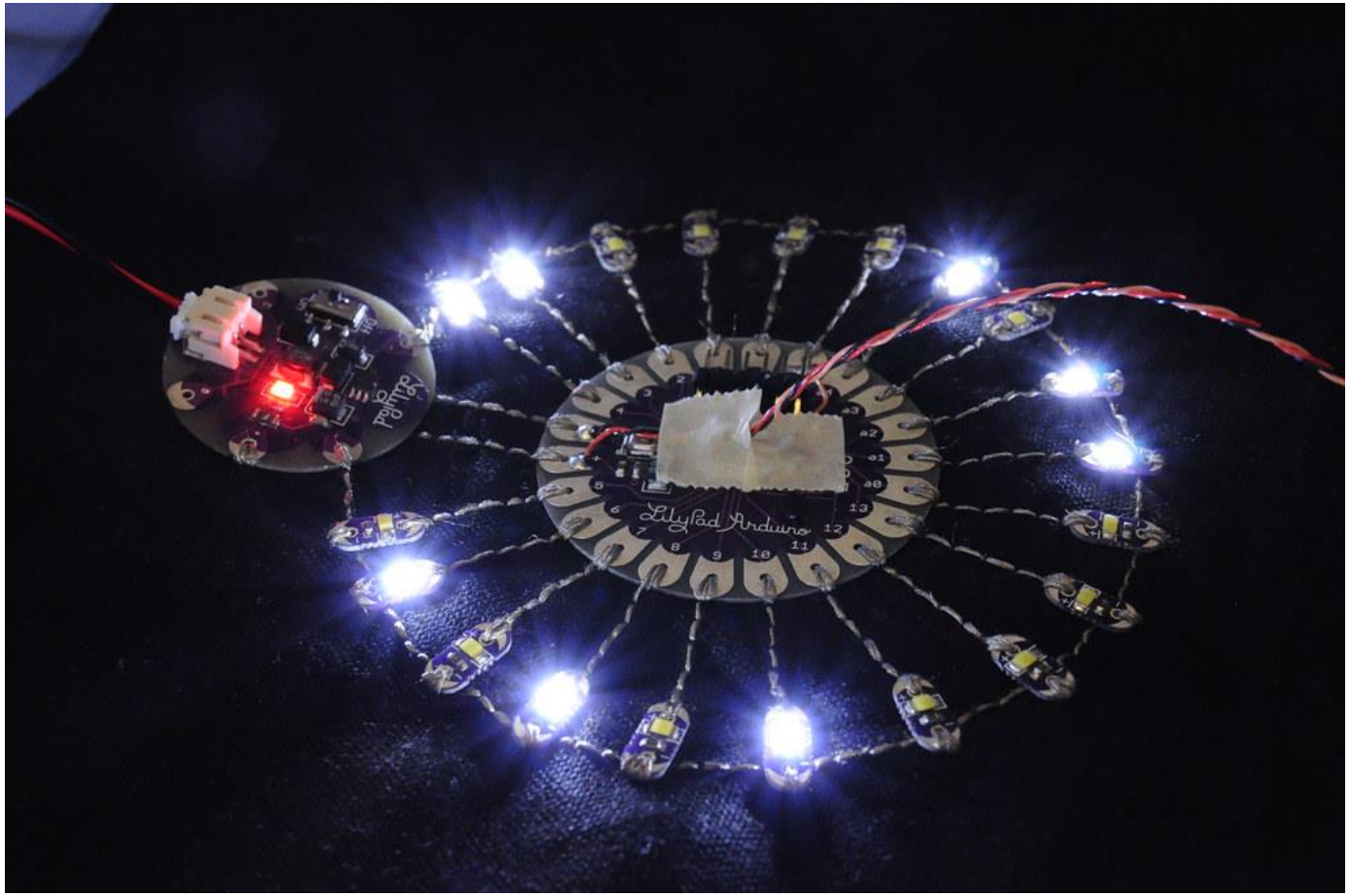
1. Collegare tre fili alla scheda. I primi due si collegano alle due lunghe file verticali sul lato della breadboard per fornire accesso all'alimentazione a 5V e al Ground. Il terzo filo passa dal pin digitale 2 a una gamba del pulsante.
2. Una gamba del pulsante si collega al resistore da 10KΩ. L'altra gamba del pulsante si collega all'alimentazione a 5V.

Circuito e Programma




```
const int buttonPin = 2;
const int ledPin = 13;

int buttonState = 0;
void setup() {

    pinMode(ledPin, OUTPUT);

    pinMode(buttonPin, INPUT);
}
void loop() {

    buttonState = digitalRead(buttonPin);

    if (buttonState == HIGH)
    {
        digitalWrite(ledPin, HIGH);
    }
    else
    {
        digitalWrite(ledPin, LOW);
    }
}
```

Fading

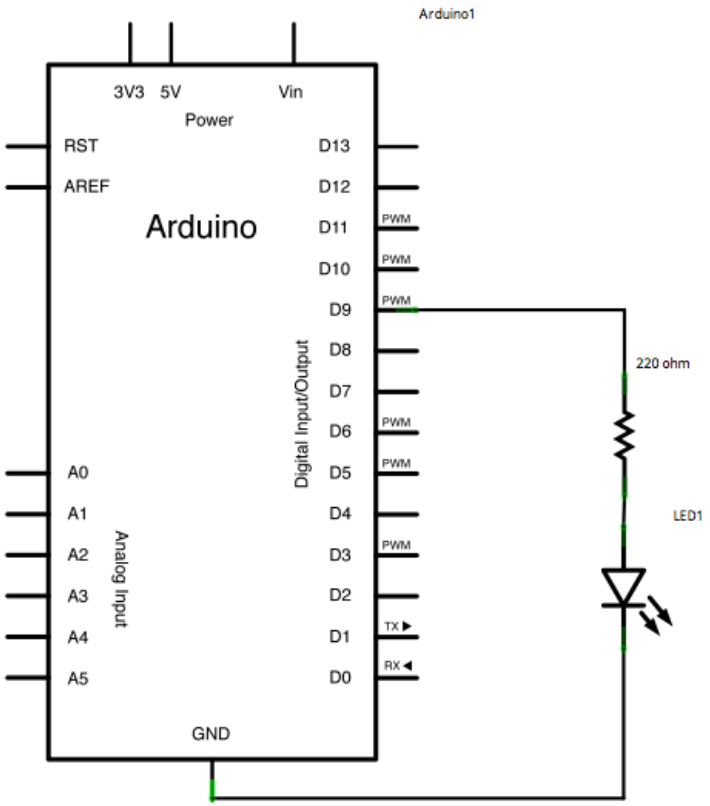


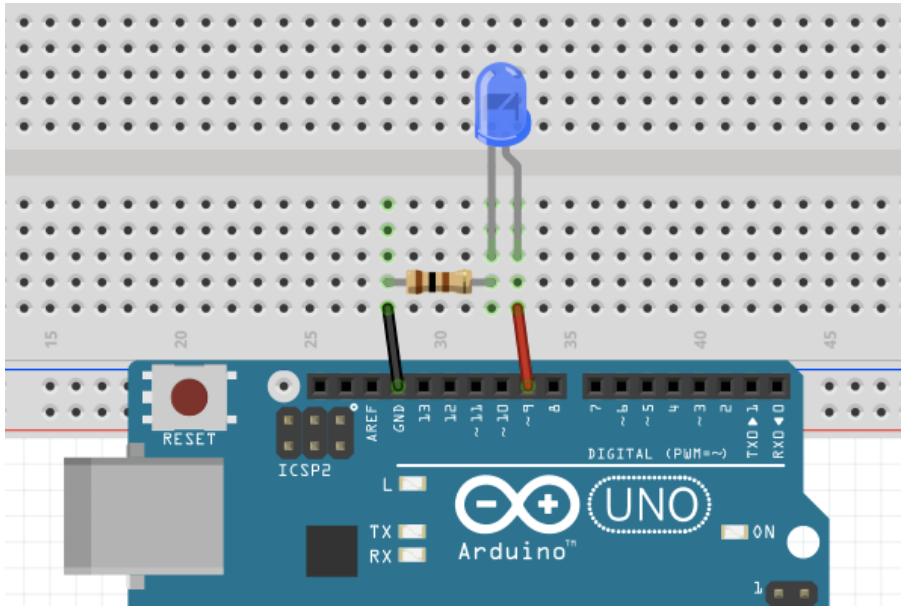
In questo esercizio vedremo l'utilizzo di come si possa utilizzare l'input e output analogico (Pulse Width Modulation (PWM)) per controllare l'illuminazione di un LED.

Hardware Richiesto:

- 1 LED
- 1 Resistenze da 220/330 Ω
- BreadBoard e cavi
- Arduino
- Cavo USB

Fading: Costruire il circuito

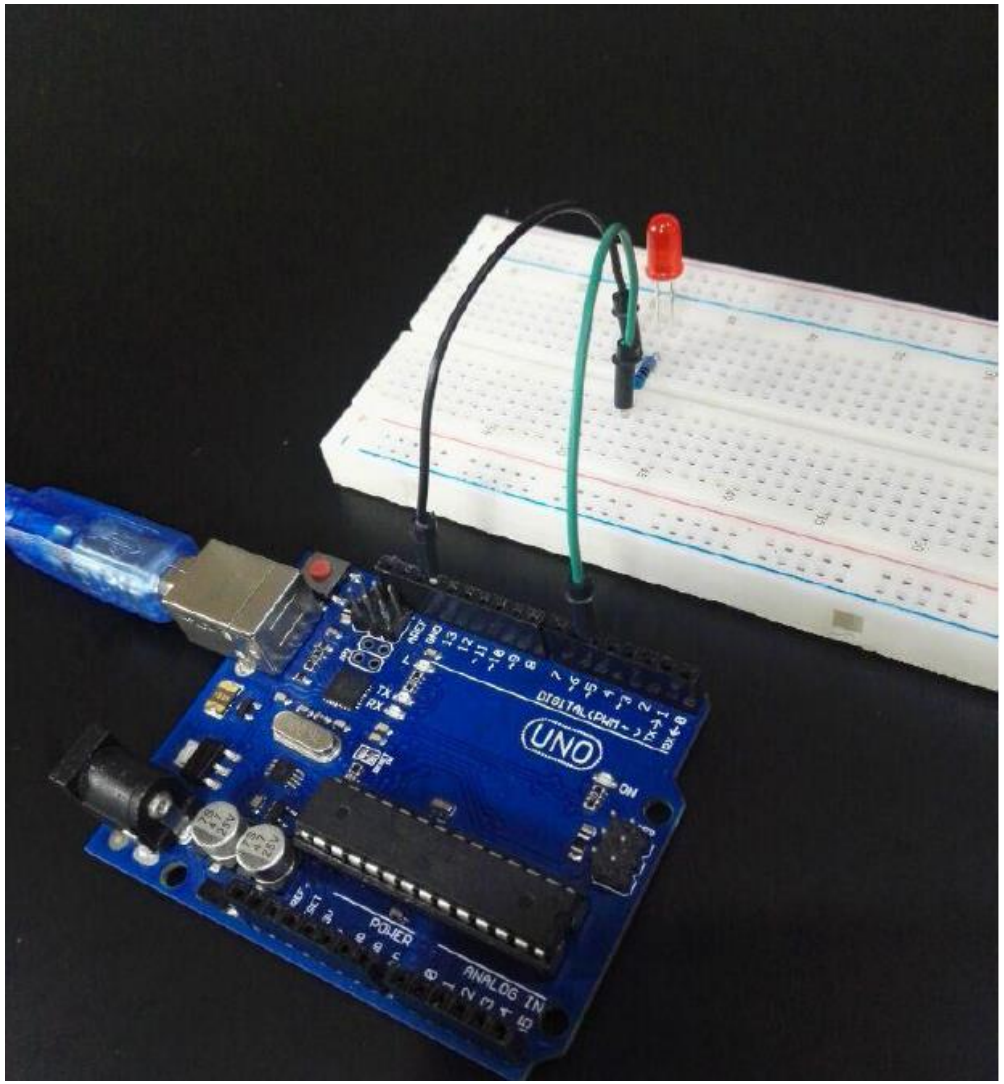




Step:

1. Collegare il pin 9 di Arduino alla resistenza.
2. Collegare alla resistenza l'anodo del LED.
3. Collegare l'altra estremita della resistenza al pin GND (ground) di Arduino

Il programma in Arduino




```
int ledPin = 5;
void setup() {
}

void loop() {

  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {

    analogWrite(ledPin, fadeValue);

    delay(30);
  }

  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {

    analogWrite(ledPin, fadeValue);

    delay(30);
  }
}
```