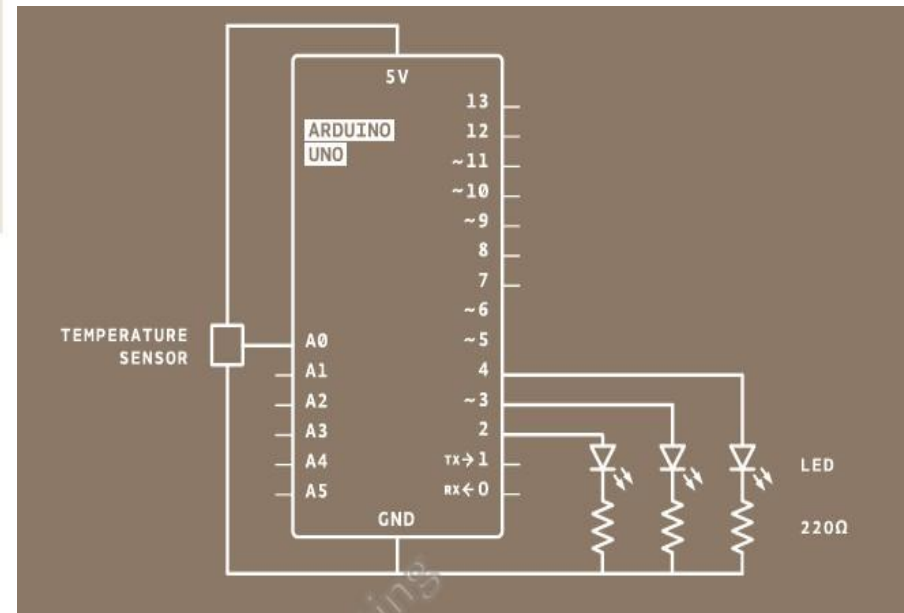
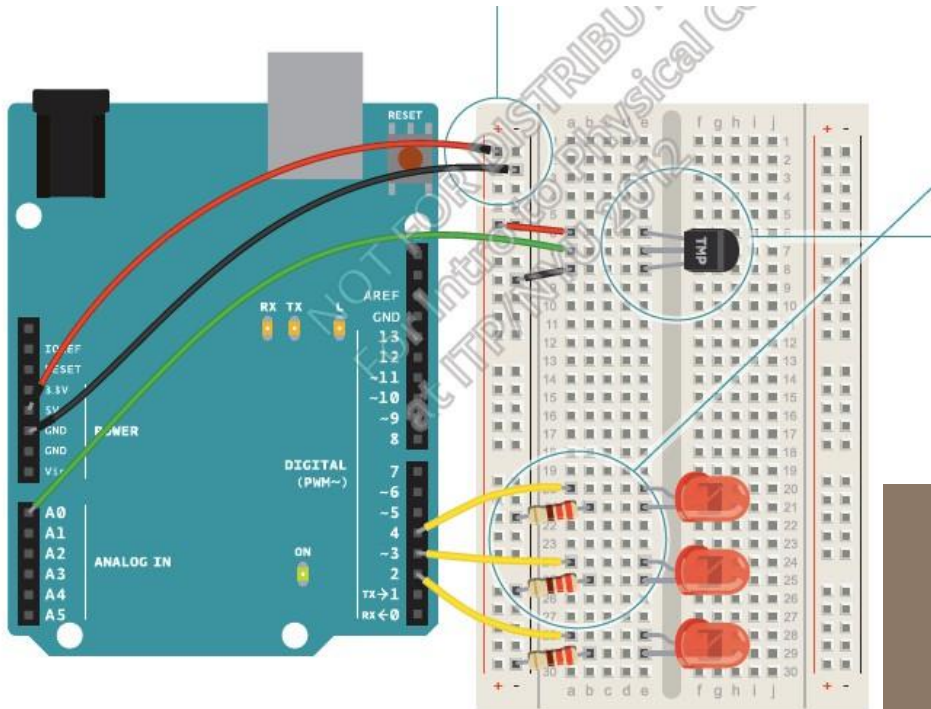


Prime applicazioni hardware

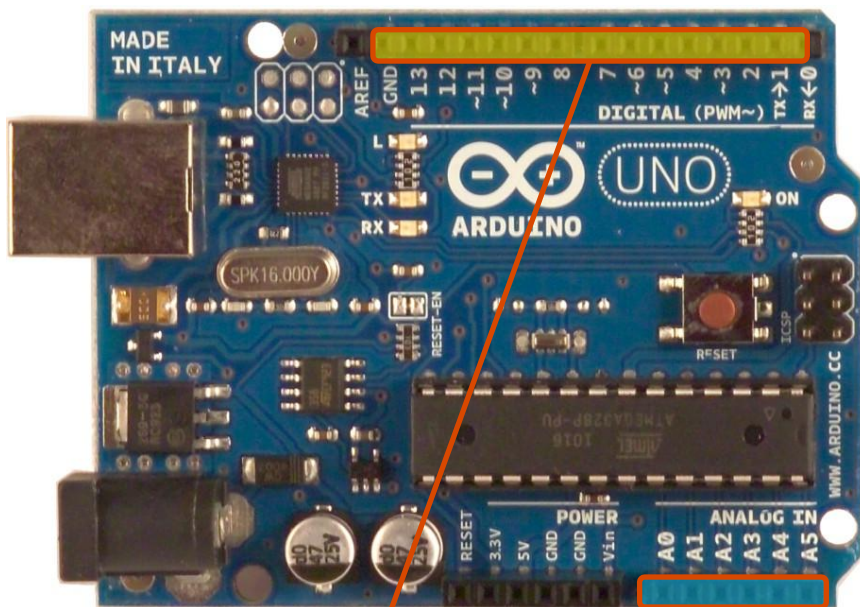
G. De Nunzio

Esercizio 3:

Lettura sensore di temperatura e LED



Com'è fatto Arduino

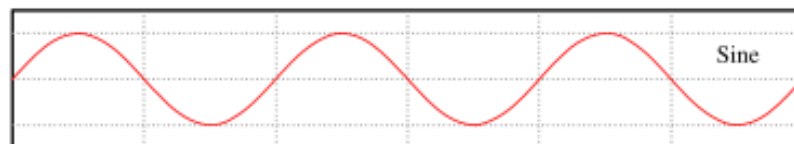


14 Ingressi/Uscite digitali

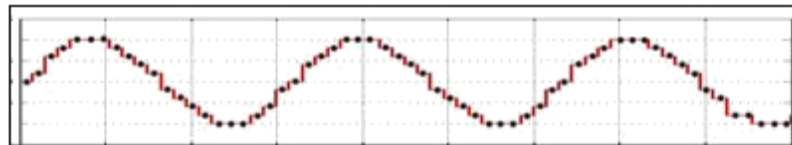
6 Ingressi analogici

Nota:

Segnale analogico: è un segnale a **tempo ed ampiezza continua**.



Segnale digitale o numerico: è un segnale a **tempo discreto e ad ampiezza quantizzata**.



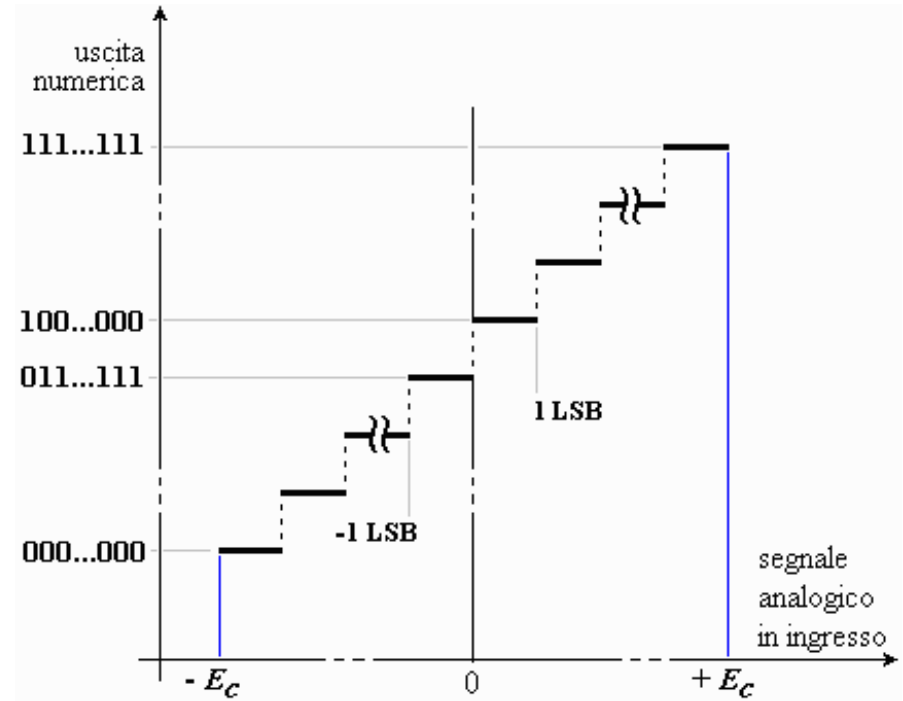
Lettura canali analogici

Arduino ha 6 ingressi analogici (A0... A5) ognuno dei quali ha una risoluzione a 10bit (cioè riconosce $2^{10} = 1024$ intervalli di tensione differenti).

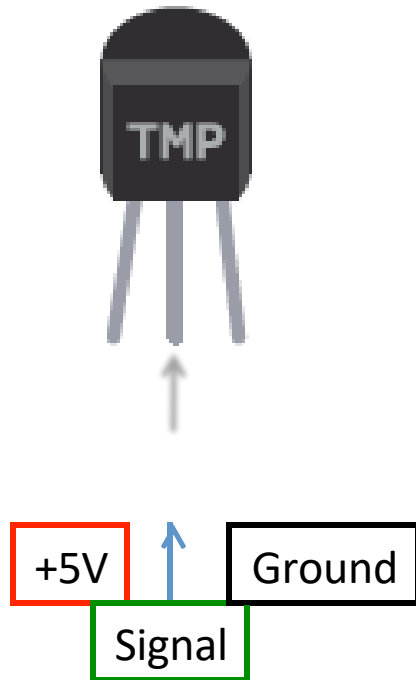
Il convertitore analogico-digitale (ADC) interno di Arduino è settato di default per acquisire valori tra 0 e 5V. Questo vuol dire che l'intervallo di 5V sarà diviso in 1024 intervalli.

Per ottenere il valore in volt:

$\text{Voltaggio} = (\text{valore letto} / 1024) \times 5V$



Sensore di temperatura: TMP36



- Il dispositivo dà un segnale in tensione (**voltage**), legato alla temperatura dalla seguente relazione:

$$\text{temperature} = (\text{voltage} - 0.5) \times 100$$

- Il canale analogico di Arduino converte il segnale in tensione in un numero digitale tra 0 e 1024 (10 bit) e la tensione si ottiene come segue:

$$\text{voltage} = (\text{valore letto}/1024) \times 5V$$

Codice per esercizio 3

```
const int sensorPin = A0;

const float baselineTemp = 20.0;

void setup(){
  Serial.begin(9600); // open a serial port
  for(int pinNumber = 2; pinNumber<5; pinNumber++){
    pinMode(pinNumber,OUTPUT);
    digitalWrite(pinNumber, LOW);
  }
}

void loop(){
  int sensorVal = analogRead(sensorPin);
  Serial.print("Sensor Value: ");
  Serial.print(sensorVal);

  // convert the ADC reading to voltage float
  voltage = (sensorVal/1024.0) * 5.0;
  Serial.print(", Volt: ");
  Serial.print(voltage);
  Serial.print(", degrees C: ");
  // convert the voltage to temperature in degrees
  float temperature = (voltage - .5) * 100;
  Serial.println(temperature);
  if(temperature < baselineTemp)
  {
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }
  else if(temperature >= baselineTemp+2 &&
    temperature < baselineTemp+4) {
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  } else if(temperature >= baselineTemp+4 &&
    temperature < baselineTemp+6){
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
  }else if(temperature >= baselineTemp+6){
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
  }
  delay(1);
}
```

Esercizio 4:

Utilizzo dei canali PWM

La PWM – Pulse Width Modulation è una tecnica utilizzata per la generazione di un segnale analogico utilizzando un uscita digitale.

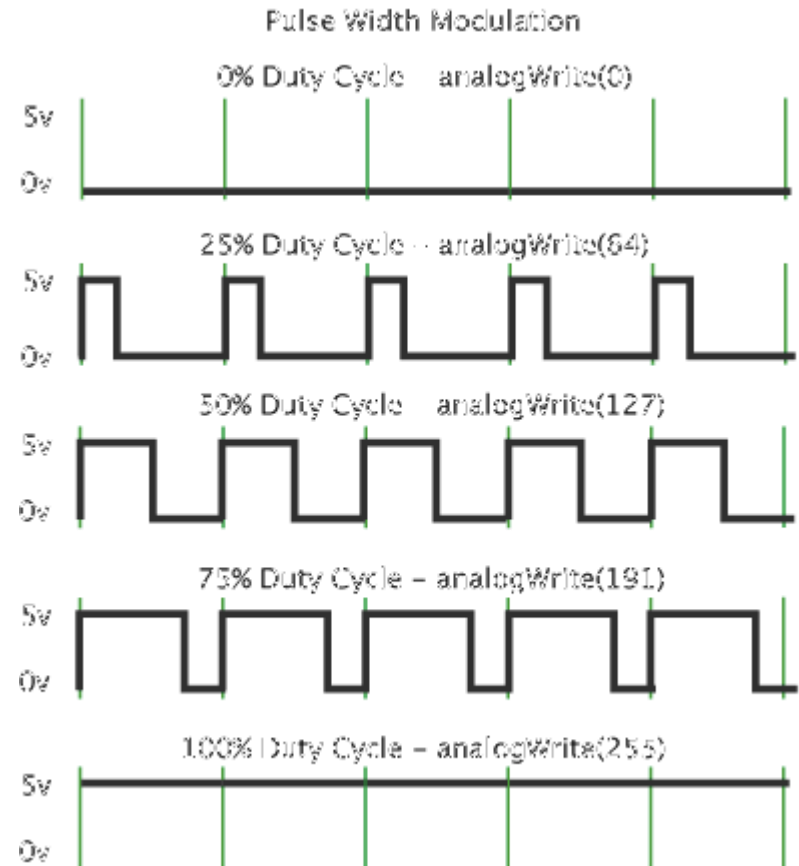
Variando la lunghezza dell'impulso posso generare dei valori analogici da 0 a Vcc (5V per Arduino).

Il duty-cycle è il rapporto tra il periodo dell'impulso al valore logico alto sul periodo in percentuale.

La frequenza di lavoro del PWM di Arduino è circa 470Hz.

Pilotando un led con questa tecnica posso far assumere diverse gradazioni di luminosità, l'occhio non percepisce il continuo on/off ma un livello differente di luminosità.

PWM 0-256 (divide periodo in 256) usa 7 bit



Esercizio 4:

Utilizzo dei canali PWM



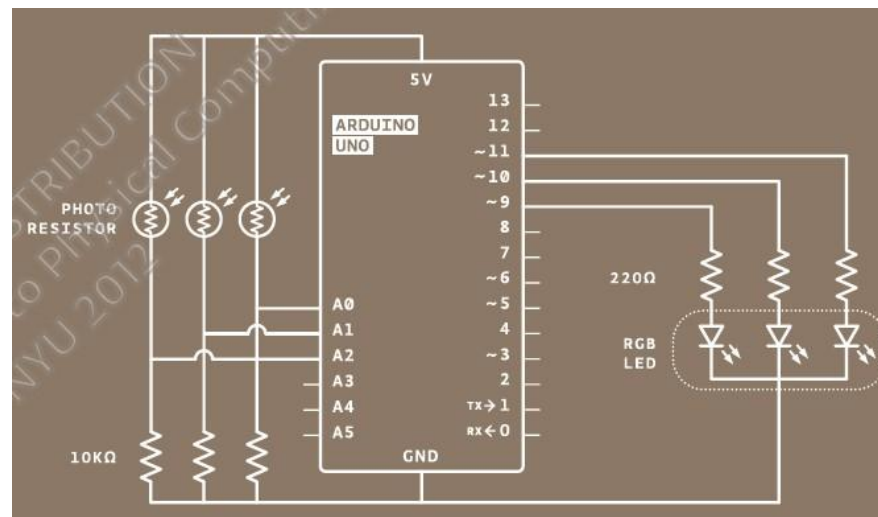
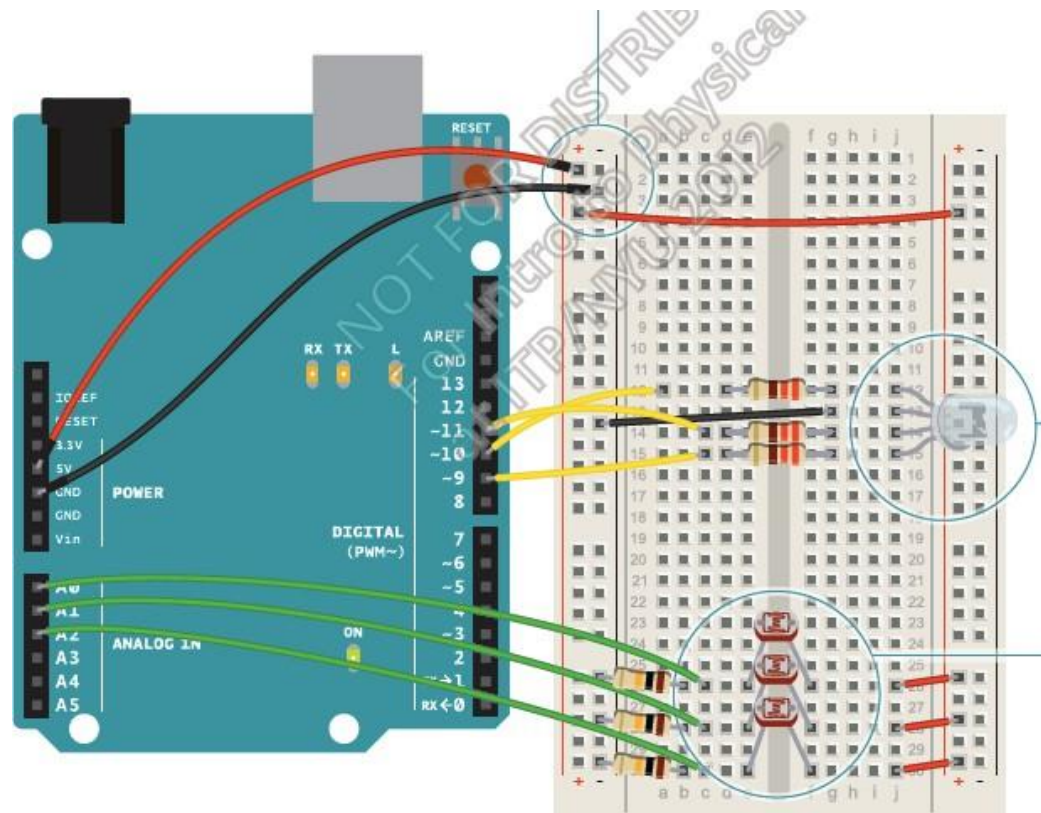
Fotoresistore
(fotosensore):
Cambia resistenza al
Variare dell'intensità
luminosa

- Con i fotosensori
modulare l'intensità
dei led
- PWD proporzionale al
segnale dei fotosensori
- Lettura analogica dei
segnali del fotosensore
(0---1024)
- Dividere per 4 il valore del
segnale per PWM (0---256)



LED RGB:
Ha 3 ingressi + ground
R (Led Rosso)
B (Led Blu)
G (Led Verde)

In base a intensità dei vari
Segnali cambia colore



The code

```
const int greenLEDPin = 9;
const int redLEDPin = 11;
const int blueLEDPin = 10;
const int redSensorPin = A0;
const int greenSensorPin = A1;
const int blueSensorPin = A2;
int redValue = 0;
int greenValue = 0;
int blueValue = 0;
int redSensorValue = 0;
int greenSensorValue = 0;
int blueSensorValue = 0;
```

```
void setup() {
  Serial.begin(9600);
  pinMode(greenLEDPin,OUTPUT);
  pinMode(redLEDPin,OUTPUT);
  pinMode(blueLEDPin,OUTPUT);
}
```

```
void loop() {
  redSensorValue = analogRead(redSensorPin);
  delay(5);
  greenSensorValue = analogRead(greenSensorPin);
  delay(5);
  blueSensorValue = analogRead(blueSensorPin);
  Serial.print("Raw Sensor Values \t Red: ");
  Serial.print(redSensorValue);
  Serial.print("\t Green: ");
  Serial.print(greenSensorValue);
  Serial.print("\t Blue: ");
  Serial.println(blueSensorValue);
```

```
  redValue = redSensorValue/4;
  greenValue = greenSensorValue/4;
  blueValue = blueSensorValue/4;
  Serial.print("Mapped Sensor Values \t Red: ");
  Serial.print(redValue);
  Serial.print("\t Green: ");
  Serial.print(greenValue);
  Serial.print("\t Blue: ");
  Serial.println(blueValue);
  analogWrite(redLEDPin, redValue);
  analogWrite(greenLEDPin, greenValue);
  analogWrite(blueLEDPin, blueValue);
}
```