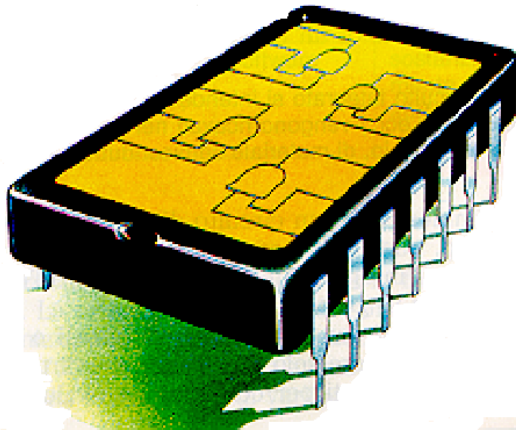


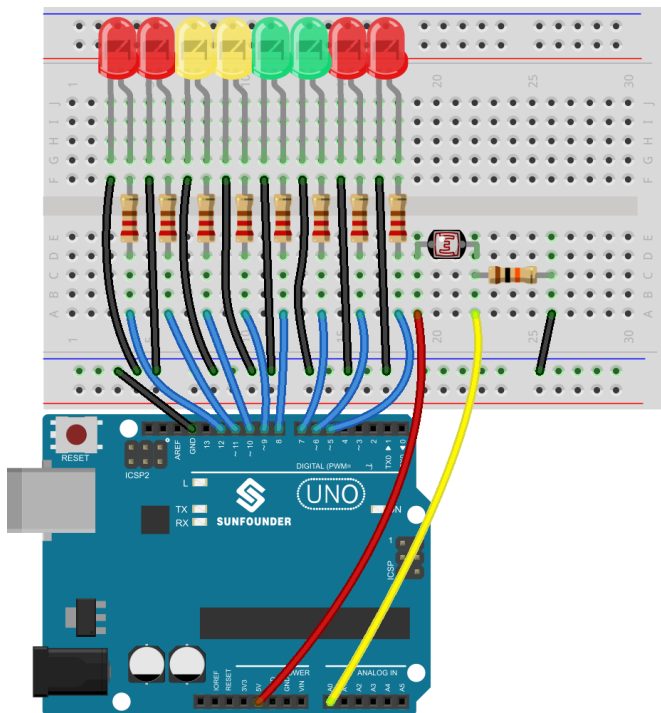
Lez. N. 14 Fotoresistenza



Materiali

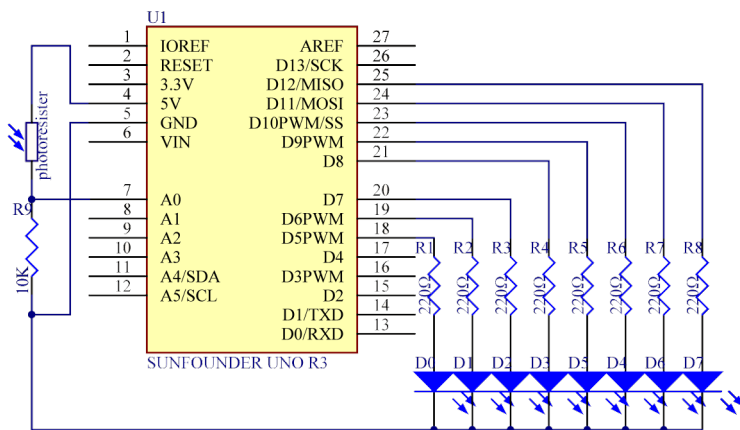
Arduino
8 Led
8 resistenze da $220\ \Omega$ a 5 bande
1 resistenza da $10\text{ K}\ \Omega$ a 5 bande
1 fotoresistore
Jumper vari

Tanta pazienza!



fritzing

schema elettrico



Introduzione

Una fotoresistenza o fotocellula è una resistenza variabile controllata dalla luce. La resistenza di una fotoresistenza diminuisce all'aumentare dell'intensità della luce incidente; in altre parole, esibisce la fotoconduttività. Una fotoresistenza può essere applicata a circuiti rivelatori sensibili alla luce e circuiti di commutazione attivati da luce e buio.

Principio sperimentale

La resistenza della fotoresistenza cambia con l'intensità della luce incidente. Se l'intensità della luce incidente è elevata, la resistenza si riduce; se bassa, aumenta.

In questo esperimento, utilizzeremo otto LED per indicare l'intensità della luce. Maggiore è l'intensità della luce, più il LED è acceso. Quando l'intensità della luce è abbastanza alta, tutti i LED saranno accesi. Quando non c'è luce, tutti i LED si spengono.

Procedure sperimentali

- Passaggio 1: Costruisci il circuito
- Passaggio 2: Programma (fare riferimento al codice di esempio in IMPARARE -> Ottieni tutorial sul nostro sito Web)
- Passaggio 3: compila il codice
- Passaggio 4: caricare lo schizzo sulla scheda Arduino Uno

Ora, se fai brillare la fotoresistenza con una certa intensità di luce, vedrai diversi LED accesi. Se aumenti l'intensità della luce, vedrai più LED accesi. Quando lo metti in un ambiente buio, tutti i LED si spengono.

Codice

```
//Phoresistor
const int NbrLEDs = 8;//number of LEDs
const int ledPins[] = {5, 6, 7, 8, 9, 10, 11, 12};
const int photocellPin = A0;
int sensorValue = 0; // value read from the sensor
int ledLevel = 0; // sensor value converted into LED 'bars'

void setup() {
for (int led = 0; led < NbrLEDs; led++)
{
pinMode(ledPins[led], OUTPUT);// make all the LED pins outputs
}
}

void loop() {
sensorValue = analogRead(photocellPin);
ledLevel = map(sensorValue, 300, 1023, 0, NbrLEDs); // map to the
number of LEDs

for (int led = 0; led < NbrLEDs; led++)
{
if (led < ledLevel ) {
digitalWrite(ledPins[led], HIGH); // turn on (attiva) pins less
than the level
}
else {
digitalWrite(ledPins[led],LOW); // turn off (disattiva) pins
higher than the level
}
}
}
```

const int ledPins[] = {5, 6, 7, 8, 9, 10, 11, 12};	Le due parentesi quadre indicano la presenza di un array, in sostanza indicano tutti gli 8 pin
---	---