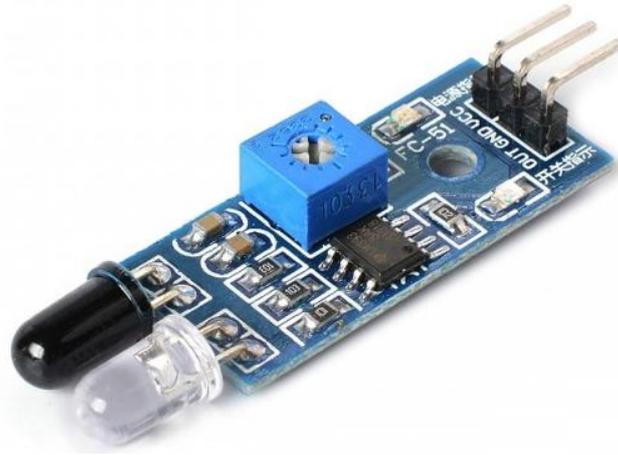


FC-51: IR Infrared Obstacle Detection Sensor Module 2 - 30cm

Giorgio De Nunzio – Giovanni Marsella



0 Premessa: i raggi infrarossi (IR)

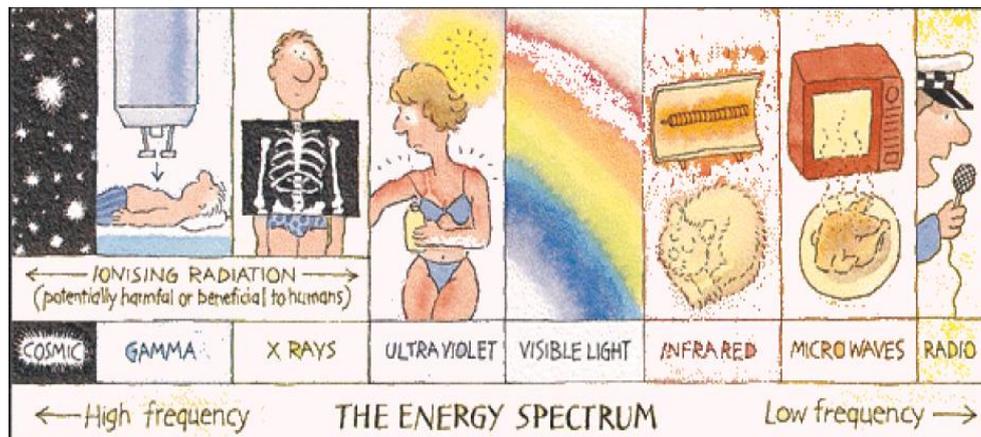


Fig. 0 – Lo spettro delle onde elettromagnetiche, in cui è presente la porzione dei raggi infrarossi (*infrared*) (da <http://galileo.phys.virginia.edu/outreach/8thgradesol/RadiationProtectionFrm.htm>, immagine proveniente dall'Uranium Information Centre, Melbourne, Australia)

1 Introduzione

La presenza di un oggetto (per esempio, un ostacolo davanti a un robot) può essere rilevata mediante un sistema a raggi infrarossi costituito da un trasmettitore e un ricevitore IR.

Più in dettaglio, un **trasmettitore IR**, o IR LED, invia un segnale a infrarossi ad una particolare lunghezza d'onda compatibile con un **ricevitore IR**, il quale ha il compito di rilevarlo.

Ci sono diversi tipi di sensori IR per diverse tipologie di applicazioni. La tecnologia IR è usata, ad esempio, nei sensori di prossimità per rilevare un oggetto nelle immediate vicinanze, nei sensori di contrasto per individuare un percorso tracciato sul pavimento, o nei sensori di conteggio per contare oggetti che passino davanti a sensore.

2 Principio di funzionamento

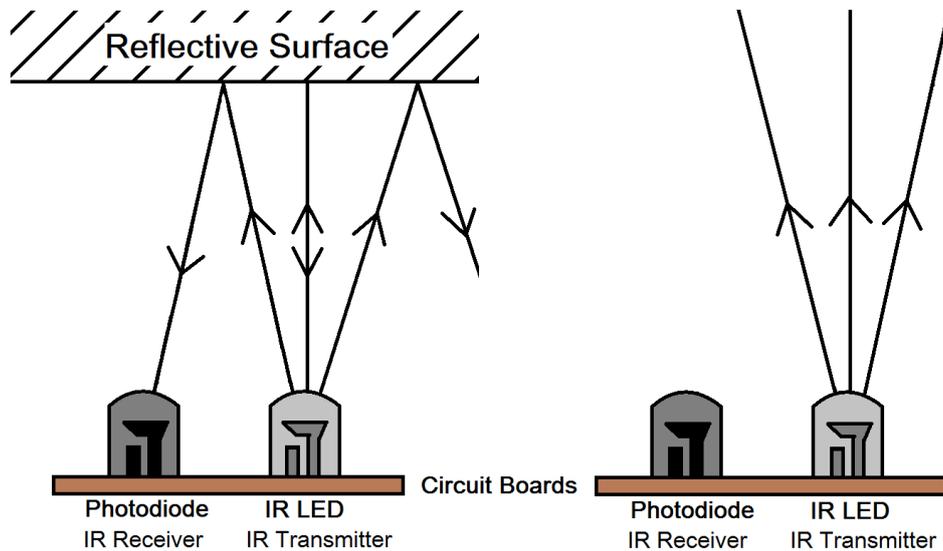


Fig. 1 – Principio di funzionamento di un sensore IR con e senza un oggetto in prossimità.

Il trasmettitore IR invia un segnale infrarosso che, in presenza di una superficie riflettente (soprattutto se di colore bianco), “rimbalza” in varie direzioni, compresa quella lungo la quale la radiazione colpisce il ricevitore IR, il quale cattura il segnale rilevando l’oggetto, e segnalandolo attraverso uno dei suoi pin.

Nel caso di una superficie assorbente (ad esempio di colore nero) il segnale IR è riflesso in minima parte, con la conseguenza che l’oggetto è difficilmente rilevato dal sensore.

2.1 Trasmettitore IR e ricevitore IR

Il trasmettitore IR è un particolare LED che emette radiazioni nel range di frequenza dell’infrarosso, invisibile a occhio nudo. Un LED infrarosso funziona esattamente come un LED nel visibile, tranne che per la lunghezza d’onda emessa.

La tensione di lavoro è di 3.3-5V DC e il consumo di corrente di circa 20mA. Il ricevitore IR, (un fotodiodo), è in grado di rilevare radiazioni infrarosse emesse da un trasmettitore IR. Esteticamente esso è simile ad un LED ma la capsula esterna può essere avvolta da un pellicola di colore scuro.

3 Sensore IR FC-51

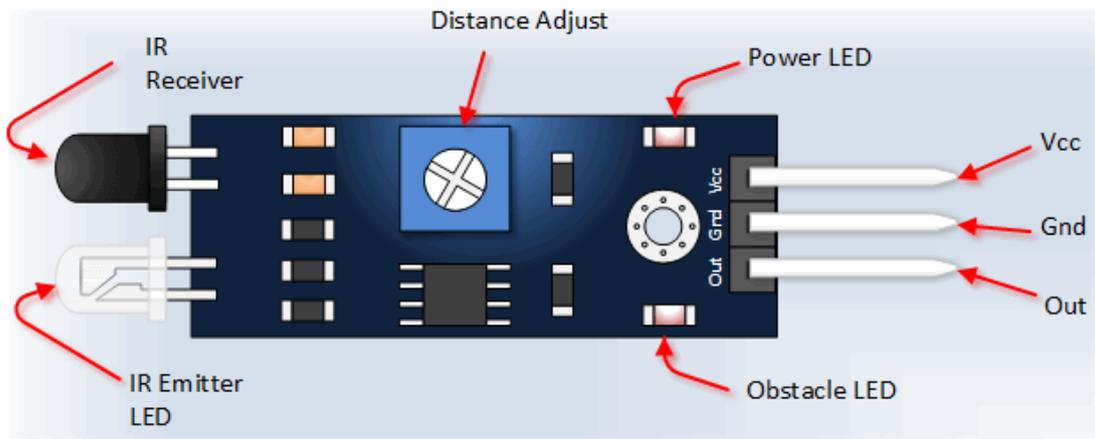


Fig.2 – Il Pin map del sensore FC-51.

Il sensore usato in queste demo è il modello **FC-51**. Si tratta di un sensore economico facilmente reperibile su internet a meno di 2€.

3.1 Configurazione dei Pin e schema elettronico

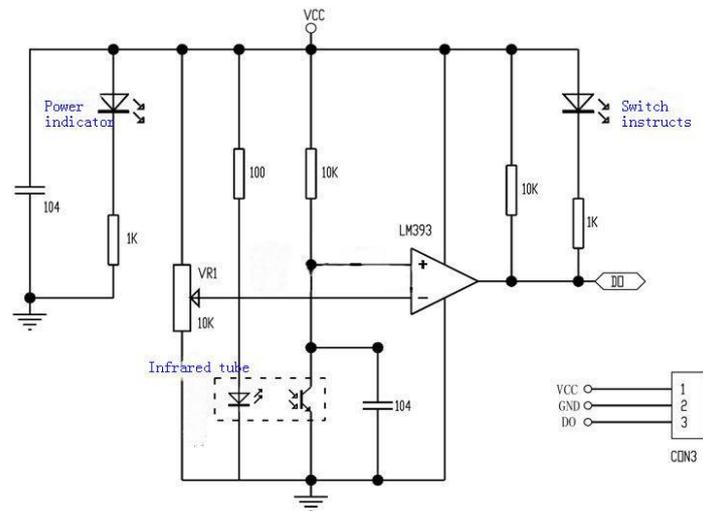


Fig.3 – Schema di un sensore IR FC-51. Il circuito integrato LM393 è un comparatore di tensione a collettore aperto (open-collector) il quale fornisce un'uscita se è presente una resistenza di pull-up tra l'uscita del IC (DO) e l'alimentazione Vcc (R=10KΩ); l'uscita DO è alta se l'oggetto non è rilevabile, bassa se l'oggetto è rilevabile.

Il componente ha tre pin di connessione:

1. Vcc per l'alimentazione 3.3-5V DC;

2. Gnd per il riferimento a massa;
3. Out per il segnale di uscita digitale del sensore.

Il Power LED sul dispositivo si illumina quando esso è alimentato. L'Obstacle LED si illumina quando si individua la presenza di un ostacolo.

L'uscita del sensore è alta (HIGH) se non è rilevato un ostacolo (il LED receiver non riceve segnali riflessi), è bassa (LOW) in caso di presenza di un ostacolo.

Questo sensore rileva **oggetti ad una distanza compresa tra 2 e 30cm**. Mediante il trimmer potenziometrico è possibile calibrare la sensibilità in base all'applicazione e alle condizioni ambientali (ad esempio la luminosità). **Ruotando il potenziometro in senso antiorario, la distanza alla quale il sensore rileva l'oggetto diminuisce, ruotandolo in senso orario tale distanza aumenta.**

E' opportuno notare che si tratta di un dispositivo piuttosto "rozzo", per cui può capitare che un ostacolo non sia rilevato se di colore scuro o se comunque assorbe le lunghezze d'onda IR, e che la precisione e la portata effettiva dipendono dalla qualità del sensore (molto variabile) e dal materiale di cui è fatto l'oggetto da rilevare.

4 Alcune demo

4.1 Test IR sensor FC-51 mediante terminale seriale (Demo 01)

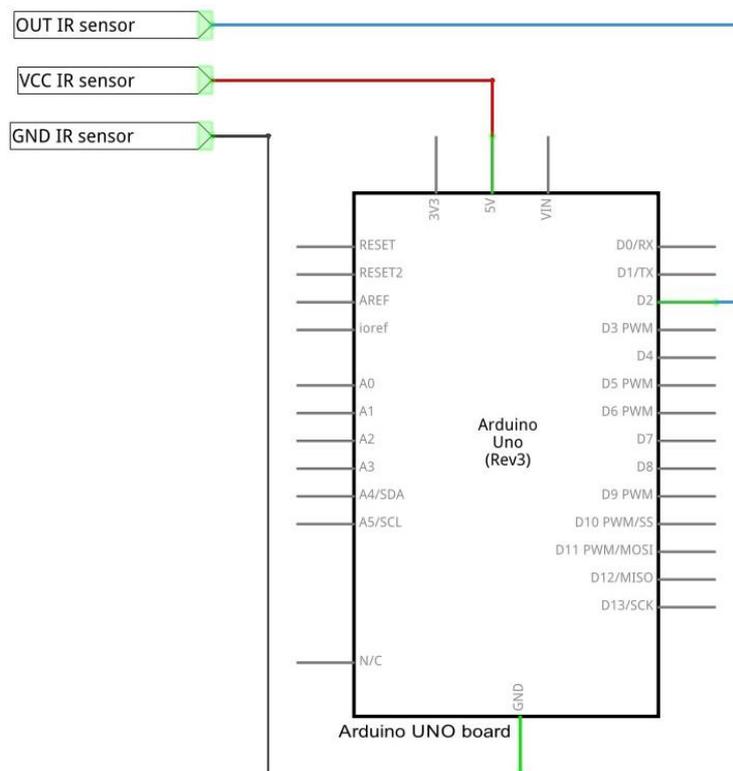


Fig.4 – Schema della demo 1.

Nella prima demo, mediante la connessione tra la porta seriale di Arduino e il PC, leggeremo l'informazione sul rilevamento dell'oggetto.

Diamo uno sguardo ai passaggi richiesti da questa demo:

1. Colleghiamo il pin d'uscita del sensore al pin digitale 2 di Arduino che chiameremo **IR**.
2. La funzione **setup()** è eseguita una sola volta prima della funzione loop. Inseriamo qui il codice di inizializzazione che abilita la porta seriale di Arduino e imposta il pin digitale 2 come un ingresso.
3. La funzione **loop()** è la funzione principale ed è ciclicamente ripetuta finché non si spegne la scheda Arduino. Convertiamo in linguaggio C il funzionamento del circuito elettronico analizzato in precedenza. Salviamo nella variabile **detection** il valore prelevato dal pin IR mediante la funzione **digitalRead**: se il valore è basso allora c'è un oggetto davanti al sensore, altrimenti non c'è (o non è rilevabile).

Ecco il codice:

```
#define IR 2
int detection;
void setup() {
  Serial.begin(9600);
  pinMode(IR, INPUT);
}
void loop() {
  detection = digitalRead(IR);
  if(detection == LOW){
    Serial.print("There is an obstacle!\n");
  }
  else{
    Serial.print("No obstacle!\n");
  }
  delay(500);    // in ms
}
```

4.2 IR sensor FC-51 e LED (Demo 02) (dal sito [2])

Questa demo aggiunge l'accensione del LED presente sulla scheda di Arduino quando è rilevato un ostacolo. Il pin di Arduino usato è diverso, ma solo per la diversa provenienza dell'esempio applicativo. Qualunque pin digitale è utilizzabile.

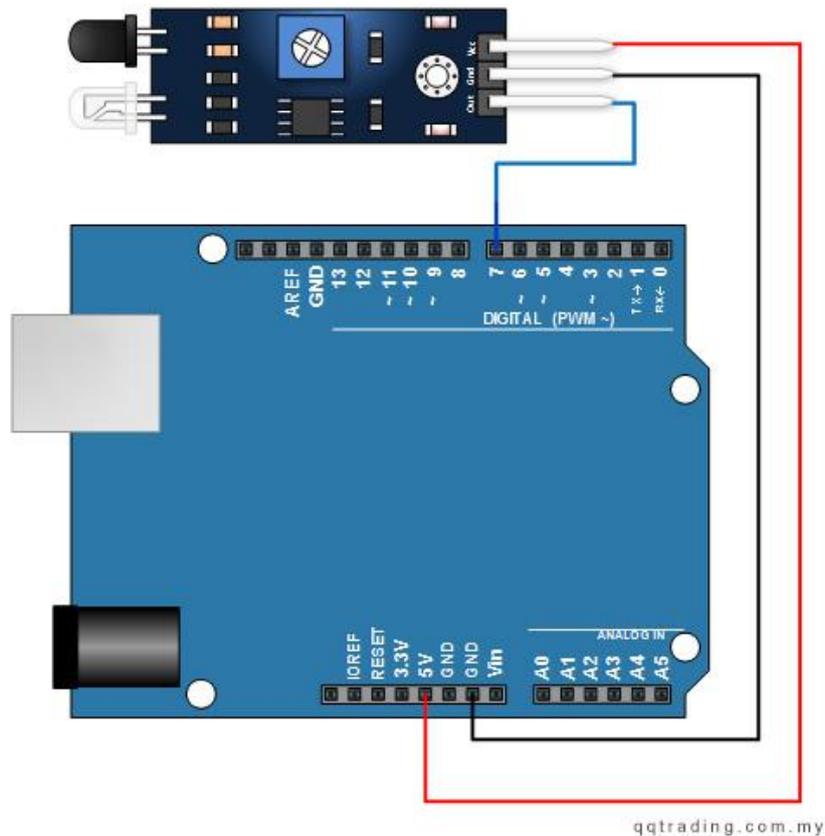


Fig.5 – Schema della demo 2.

```
// IR Obstacle Collision Detection Module

int LED = 13; // Use the onboard Uno LED
int isObstaclePin = 7; // This is our input pin
int isObstacle = HIGH; // HIGH MEANS NO OBSTACLE

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(isObstaclePin, INPUT);
  Serial.begin(9600);
}

void loop() {
  isObstacle = digitalRead(isObstaclePin);
  if (isObstacle == LOW) {
    Serial.println("OBSTACLE!!, OBSTACLE!!");
    digitalWrite(LED, HIGH);
  } else {
    Serial.println("clear");
    digitalWrite(LED, LOW);
  }
  delay(200);
}
```

4.3 IR sensor FC-51 e LED (Demo 03)

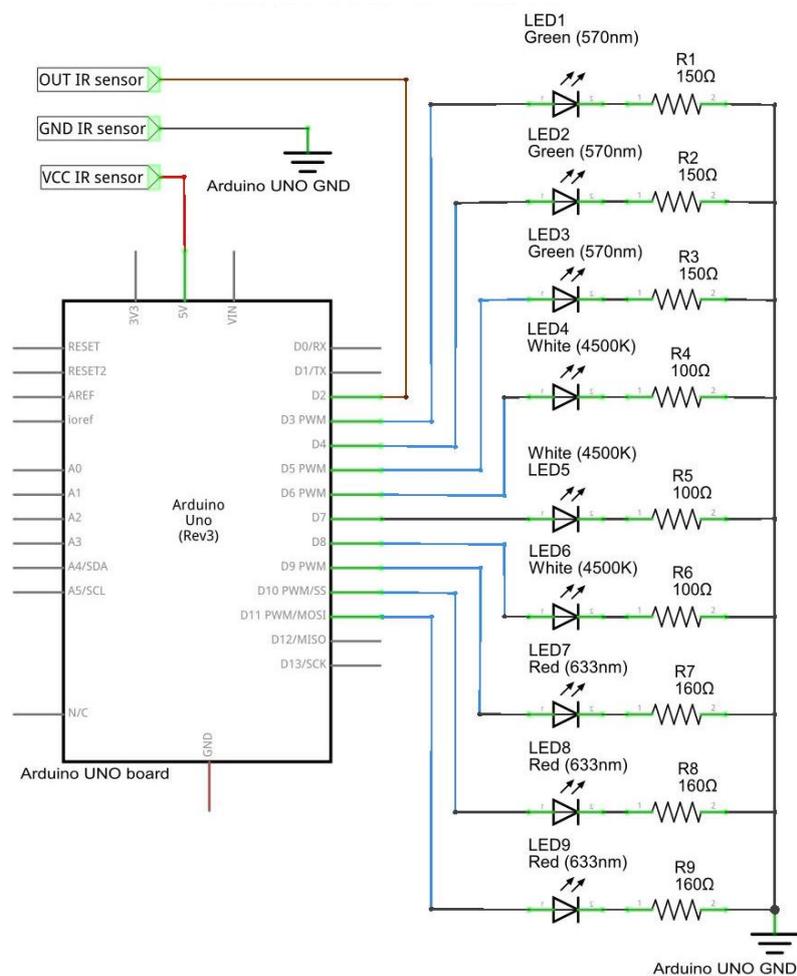


Fig.6 Schema elettrico della demo 3

In questa demo associamo un'uscita ad ogni stato di funzionamento del sensore IR. I componenti richiesti sono (vedere anche la nota alla fine del paragrafo!):

- Sensore IR FC-51;
- 3 x LEDs verde;
- 3 x R=150Ω;
- 3 x LEDs bianco;
- 3 x R=100Ω;
- 3 x LEDs rosso;
- 3 x R=160Ω.

Ricordate che i pin I/O sono in grado di assorbire/erogare 40mA max, in totale 200mA max (vedi il datasheet dell'**ATmega328P**). Diamo uno sguardo ai passaggi richiesti da questa demo:

1. Colleghiamo il pin d'uscita del sensore al pin digitale 2 di Arduino. Definiamo i pin digitali dei LEDs come un array di pin, da 3 a 11 che chiamiamo **LedPIN**.
2. La funzione **setup()** è eseguita una sola volta prima del main loop. In aggiunta al codice di inizializzazione già visto, dichiariamo come output i 9 LED mediante un ciclo for.
3. La funzione **loop()** è la funzione principale ed è ciclicamente ripetuta finché non si spegne la scheda Arduino. Salviamo nella variabile **detection** il valore prelevato dal pin **IR** mediante la specifica funzione **digitalRead()**. Questo valore può essere basso se c'è un oggetto o alto se non c'è.

Diamo uno sguardo al codice:

```
#define IR 2    // digital pin input for ir sensor
int detection;
int i;
// array digital pin for:
// green led(3,4,5) - white led (6,7,8)- red led (9,10,11)
int LedPIN[] = {3, 4, 5, 6, 7, 8, 9, 10, 11};

void setup() {
  pinMode(IR, INPUT);
  for(i = 0; i < 9; i++){
    pinMode(LedPIN[i], OUTPUT);
  }
}

void loop() {
  detection = digitalRead(IR);
  if(detection == LOW){ // there is an object!
    BlinkLED();
  }
  else{
    LedOFF();
  }
}

void BlinkLED() {
  for(i = 0; i < 9; i++) {
    digitalWrite(LedPIN[i], LOW); // turn all the LEDs off
  }
  delay(150);
  for(i = 0; i < 9; i++) {
    digitalWrite(LedPIN[i], HIGH); // turn all the LEDs on
  }
  delay(150);
}

void LedOFF() {
  for(i = 6; i < 9; i++) {
    digitalWrite(LedPIN[i], LOW);
  }
  delay(150);
  for(i = 3; i < 6; i++) {
    digitalWrite(LedPIN[i], LOW);
  }
  delay(150);
  for(i = 0; i < 3; i++) {
```

```
    digitalWrite(LedPIN[i], LOW);  
  }  
  delay(150);  
}
```

Nota: il codice è solo una delle possibili varianti: quando è presente un ostacolo, fa lampeggiare i nove LED, quando non vi è ostacolo, se sono accesi li spegne in sequenza, in gruppi di tre. Naturalmente, avendo a disposizione meno LED o volendo modificare il codice, è possibile realizzare un circuito che usi solo un LED (che lampeggi se c'è ostacolo, sia spento altrimenti) o con un numero a piacere.

Sitografia

[1] <http://www.playembedded.org/blog/it/2016/01/08/rilevare-un-ostacolo-con-il-sensore-ir-e-arduino/>

[2] <http://qqtrading.com.my/ir-infrared-obstacle-detection-sensor-module-fc-5>

[3] <http://www.playembedded.org/blog/it/2016/02/29/un-contatore-di-oggetti-utilizzando-un-sensore-ir-e-arduino/>