



Fondazione ITS Academy per la Mobilità Sostenibile Aerospazio/Meccatronica e Servizi alle Imprese - Piemonte

Corso ITS 15^a Edizione

A.F. 2025-2026 - Torino

Tecnico Superiore per l'Automazione e la Robotica Industriale

Embedded System e AI per l'Automazione – EMBT03

INDICE

1 OBIETTIVO.....	3
2 COMPONENTI E STRUMENTI.....	3
3 PROCEDIMENTO.....	3
3.1 Ideazione.....	3
3.2 Assemblaggio.....	3
3.3 Realizzazione delle animazioni.....	3
3.3.1 <i>Sviluppo dei frame</i>	3
3.3.2 <i>Implementazione nel codice</i>	3
3.4 Realizzazione del programma.....	3
3.4.1 <i>Campionamento</i>	3
3.4.2 <i>Definizione dei livelli del volume</i>	3
3.4.3 <i>Selezione e visualizzazione frame</i>	3
4 CONCLUSIONE.....	3
GLOSSARIO.....	4

Sensore Distanza

Individuazione distanza e output visivo

1 Obiettivo

Utilizzando una scheda Arduino Uno R4 ed un sensore ad ultrasuoni SRF05, realizzare un sistema che:

- Illumini a piena potenza un led rosso quando nessun oggetto si trova entro 500 mm dal sensore
- Riduca l'illuminazione del led in modo proporzionale al diminuire della distanza fino a spegnersi del tutto quando la distanza risulta minore di 100mm
- Stampa ogni secondo sul serial monitor la distanza in mm dell'oggetto posto di fronte al sensore e il livello di intensità del segnale

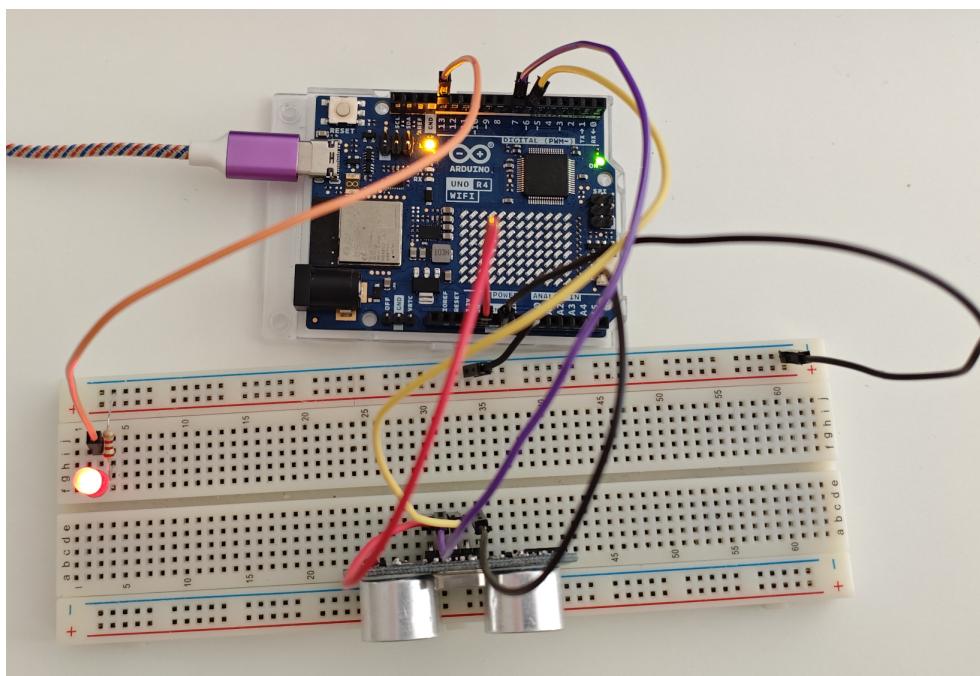
2 Componenti e Strumenti

Hardware:

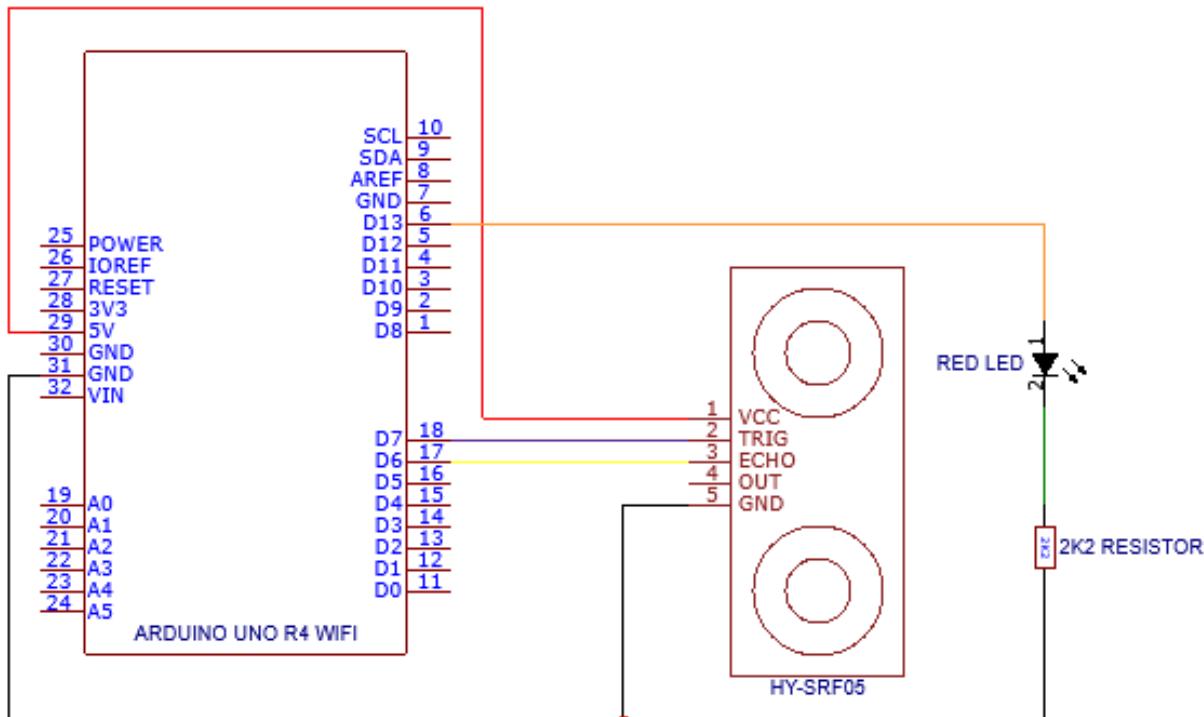
- Arduino Uno R4 WiFi
- Sensore HY-SRF05
- LED rosso
- resistenza 2K2Ω
- cavi jumper

Software:

- Arduino IDE
- libreria SRF05.h di Rob Tillaart



3 Procedimento



3.1 Assemblaggio

HY-SRF05:

- VCC a 5V
- GND a GND
- Trig al PIN digitale 7
- Echo al PIN digitale 6

LED:

- Anodo al PIN digitale 13
- Catodo alla resistenza e poi a GND

3.2 Realizzazione del programma

3.2.1 Libreria

È stata utilizzata la libreria SRF05.h di Rob Tillaart, per comunicare con il sensore, viene importata con il comando:

```
#include "SRF05.h" // Aggiunta della libreria per gestire il sensore
```

e utilizzata con i comandi:

```
SRF05 SRF(TRIG, ECHO); // Configurazione necessaria per il richiamo della libreria
distance = SRF.getMillimeter(); // acquisizione della distanza tramite la funzione
fornita dalla libreria
```

3.2.2 Definizione variabili

Si definiscono le seguenti variabili:

```
// PIN necessari per il funzionamento del sensore, vengono gestiti dalla libreria
const byte TRIG = 7;
const byte ECHO      = 6;

const byte RED_LED = 13; // PIN a cui è collegato il led
int distance;           // variabile che contiene la distanza
int intensity;          // variabile per stabilire l'intensità variabile del led nel
range stabilito

// variabili per gestire la stampa sulla seriale ogni secondo mantenendo la lettura
del sensore costante
long int start_time;
long int end_time;
```

3.2.3 Loop

```
void loop() {  
  
    delay(10);  
  
    distance = SRF.getMillimeter(); // acquisizione della distanza tramite la funzione fornita dalla libreria  
  
    if (distance>500){ // se distanza maggiore di 500mm led massima intensità  
        intensity = 255;  
        analogWrite(RED_LED, intensity);  
    } else if (distance<100){ // se distanza minore di 100mm led spento  
        intensity = 0;  
        analogWrite(RED_LED, intensity);  
    } else if (distance >= 100 && distance <= 500){ // se distanza tra 100 e 500 intensità del led variabile  
        intensity = map(distance, 100, 500, 0, 255);  
        analogWrite(RED_LED, intensity);  
    }  
  
    // stampa sulla seriale dopo 1sec la distanza letta dal sensore  
    end_time = start_time + 1000;  
    if (millis()>end_time){  
        Serial.print("Distanza: ");  
        Serial.print(distance);  
        Serial.print(" - Intensità: ");  
        Serial.println(intensity);  
        start_time = end_time;  
    }  
}
```

4 Conclusione

Per concludere l'approccio con l'utilizzo della libreria si è dimostrato ottimale permettendo di ottimizzare i tempi e di utilizzare il sensore senza difficoltà