

Documentazione Tecnica

Microcontrollore

Progetto Smart Office - IoT and 3D Intelligent Systems

Puzone Felicia

Arfaoui Bilel

Macellaro Vincenzo





Panoramica

Questa documentazione ha lo scopo di descrivere in dettaglio le schematiche, i componenti e le routine del dispositivo IoT di basso livello.

Il codice sorgente è presente nella cartella `/arduino`.

Il file `main.ino` contiene il codice MCU

il file `bridge_receiver.py` contiene il codice python del bridge.

Struttura

Per la costruzione del dispositivo IoT si è scelto di utilizzare una board *ELEGOO UNO R3 BOARD*, equivalente ad Arduino Uno. L'alimentazione è ottenuta parzialmente da seriale e parzialmente da un battery case. Il wiring delle componenti avviene tramite una breadboard.

Data la mancanza di connettività di rete in microcontrollore, la comunicazione con l'esterno avviene con uno **script Python** che funge da **bridge**.

Il bridge assume il compito di sottoscrivere i canali MQTT per la ricezione dei dati degli attuatori, e pubblicare sui canali dedicati ai sensori.

La gestione della comunicazione MCU-Bridge avviene attraverso un **protocollo embedded** custom.

Componenti

Componenti utilizzate per il prototipo e rispettivi wiring.

ELEGOO UNO R3 BOARD

The ELEGOO UNO is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains

everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The UNO differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

La **motivazione della scelta** è stata prettamente pratica, essendo già in possesso del componente.



ULTRASONIC SENSOR HC-SR04

This is the HC-SR04 ultrasonic distance sensor. This economical sensor provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm. Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit.



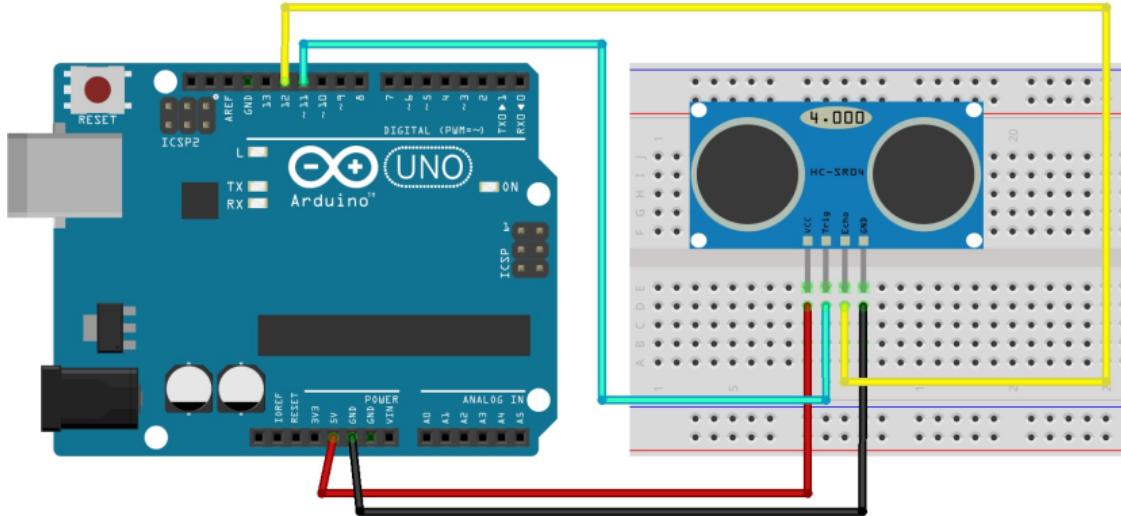
Wiring

VCC Pin: Linea 5V

GND: Linea Ground

TRIG: Digital Pin 6

ECHO: Digital Pin 7



E' stato scelto con lo scopo di simulare il "passaggio" di una persona attraverso la porta del prototipo. Funge da sensore di prossimità.

ADAFRUIT NEOPIXEL DIGITAL RGB LED STRIP

NeoPixels are 5050-sized LEDs with an embedded microcontroller **inside the LED**. You can set the brightness and color of each R/G/B/W with 8-bit PWM precision (so 32-bit color per pixel). The LEDs are controlled by shift-registers that are chained up down the strip so you can shorten or lengthen the strip. Only 1 digital output pin is required to send data down. The PWM is built into each LED-chip so once you set the color you can stop talking to the strip and it will continue to PWM all the LEDs for you.

Structure of the led

Technical Specs:

- **12.5mm (0.5") wide**, 4mm (0.16") thick with casing on, 16.7mm (0.65") long per segment
- 60 LEDs per meter
- Weight of 1 meter (w/ silicone jacket and 6" leads): 39g
- Removable IP65 weatherproof casing



- Maximum 5V @ 60mA draw per 0.65" strip segment (all LEDs on full brightness)
- 5VDC power requirement (do not exceed 6VDC) - no polarity protection
- 1 integrated RGB LEDs per segment, individually controllable
- LED wavelengths: 630nm/530nm/475nm
- Connector: [2-pin JST SM](#)
- [WS2812 Datasheet](#)
- [SK6812 Datasheet](#)
- May ship with either WS2812B or SK6812-based LEDs. They are the same brightness, color and protocol

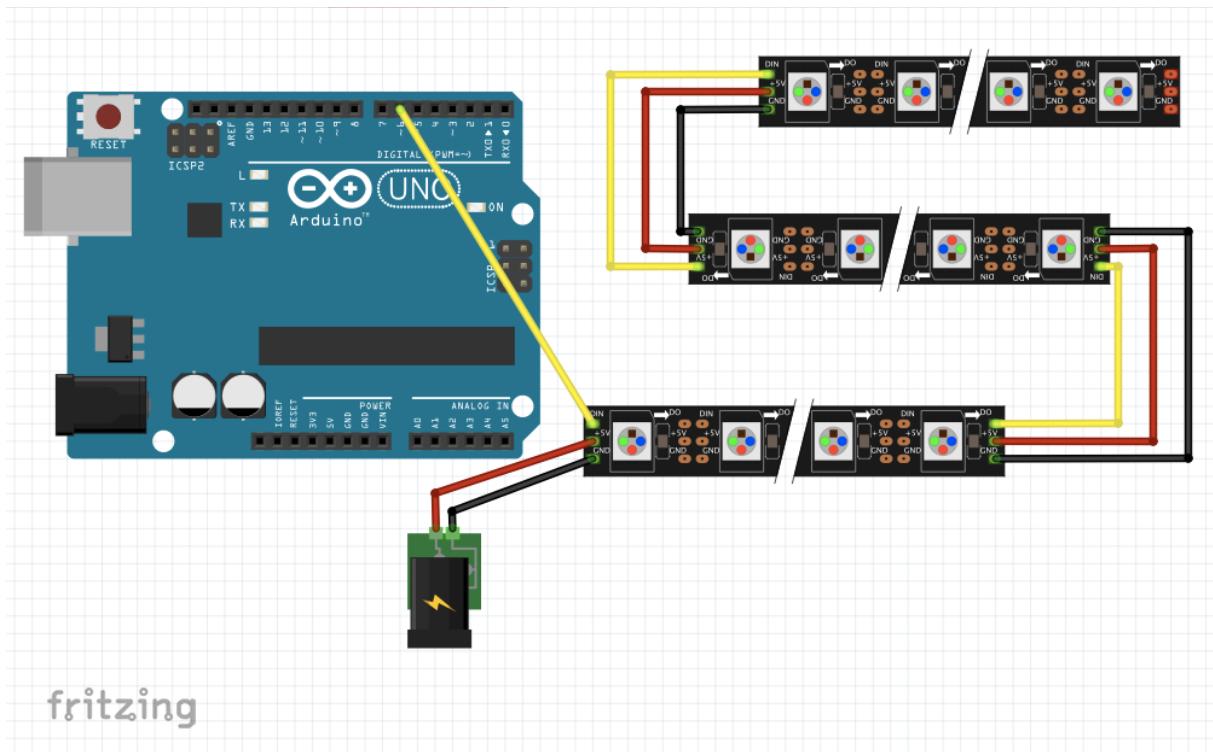
Wiring

VCC Pin: Linea 5V

GND: Linea Ground

DATA: Digital Pin 8

Numero di NeoPixel: 62



Questi LED consumano circa 9,5 Watt max (~2 Amp a 5 V) per metro. L'alimentazione ricevuta in seriale dal MCU raggiunge un'intensità di corrente pari a: 500 mA. Un supporto contenente 3 batterie AAA da 4.5V (circa 540 mA ciascuna) è stato sufficiente a garantire sufficiente potenza.



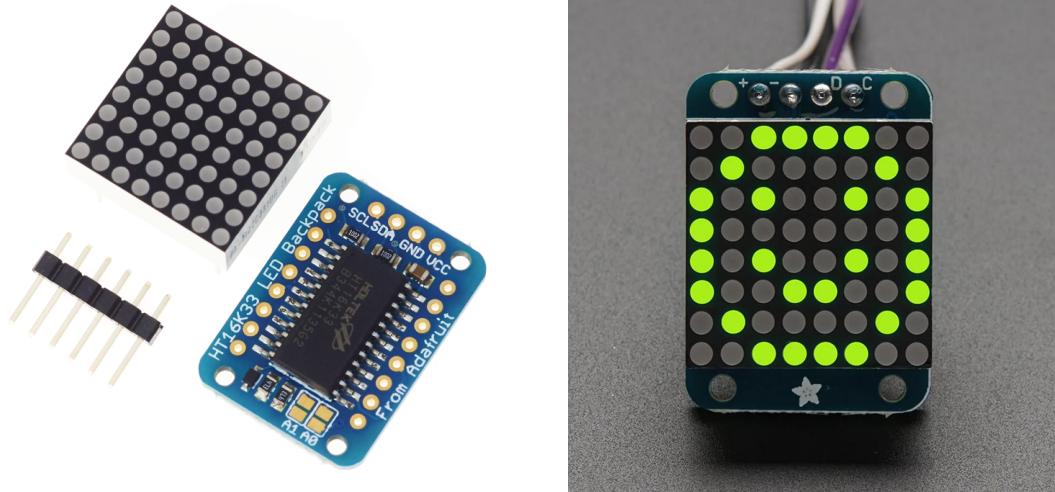
shutterstock.com • 693713785

Per collegare la restante parte del circuito alimentata da seriale, è stato necessario collegare il VCC del battery case alla linea VCC, e il ground alla linea ground.

ADAFRUIT 8x8 LED MATRIX with I2C BACKPACK

This product is composed by:

- Ultra-bright 1.2" 8x8 yellow/orange matrix
- A fully tested and assembled 1.2" LED backpack



The matrices use a driver chip that commands: They have a built-in clock so they multiplex the display. They use constant-current drivers for ultra-bright, consistent color, 1/16 step display dimming, all via a simple I₂C interface. These 1.2" matrix backpacks come with three address-selection jumpers so you can connect up to eight 1.2" 8x8's together (or a combination, such as four 1.2" 8x8's and four 7-segments, etc) on a single I₂C bus.

This board/chip uses **I₂C** 7-bit addresses between 0x70-0x77, selectable with jumpers.

Il principale scopo dell'elemento è quello di segnalare un diverso tipo di stato che l'oggetto assume. Il **motivo** per cui è stata acquistata è la sua semplicità di programmazione, nonché l'idea carina di poter visualizzare icone e **piccole animazioni** per indicare stati diversi.

Le **bitmap** della matrice sono molto semplici da strutturare, e nel nostro caso sono state inserite all'interno di un file *anim.h*.

Datasheet: <https://learn.adafruit.com/adafruit-led-backpack/downloads>

Wiring

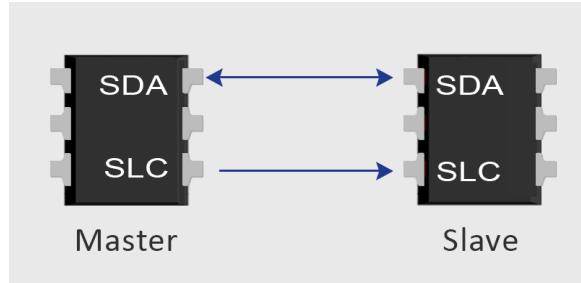
VCC Pin: Linea 5V

GND: Linea Ground

SDA: A4

SCL: A5

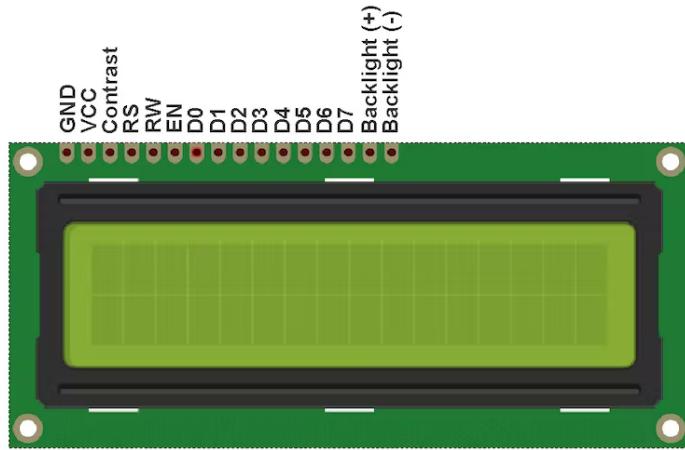
I pin SDA e SCL sono utilizzati da Arduino per implementare il protocollo i2C



I2C Protocol

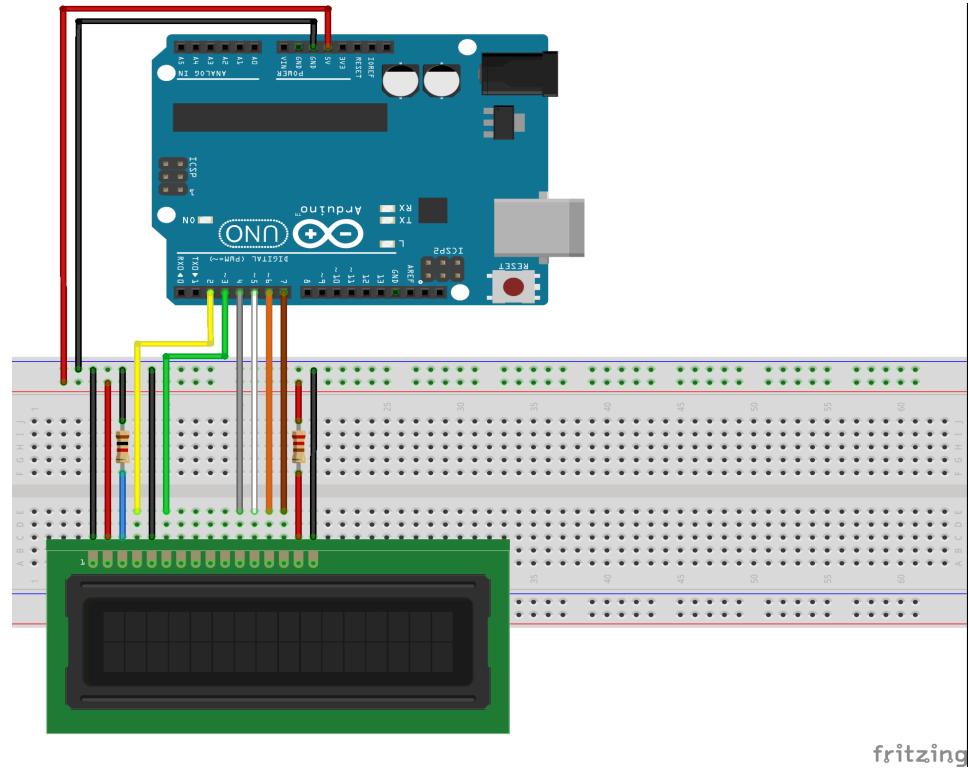
LCD DISPLAY HD44780

Il display LCD permette di visualizzare dati presenti all'interno del programma agli utilizzatori.



all'interno del progetto, lo scopo di questo componente è quello di simulare l'output di un termostato. Componente molto versatile, può segnalare piccole stringhe come codici di errore o avvisi. Si è rivelato utile anche in fase di debug.

Libreria utilizzata: [LiquidCrystal](#)



Wiring

Le resistenze utilizzate sono:

- 1000 Ohm
- 2 resistori da 2 KOhm

VCC Pin: Linea 5V

GND: Linea Ground

Pin Arduino: 11, 12, 5, 4, 3, 2, 1

BATTERY HOLDER 3xAAA 4.5V

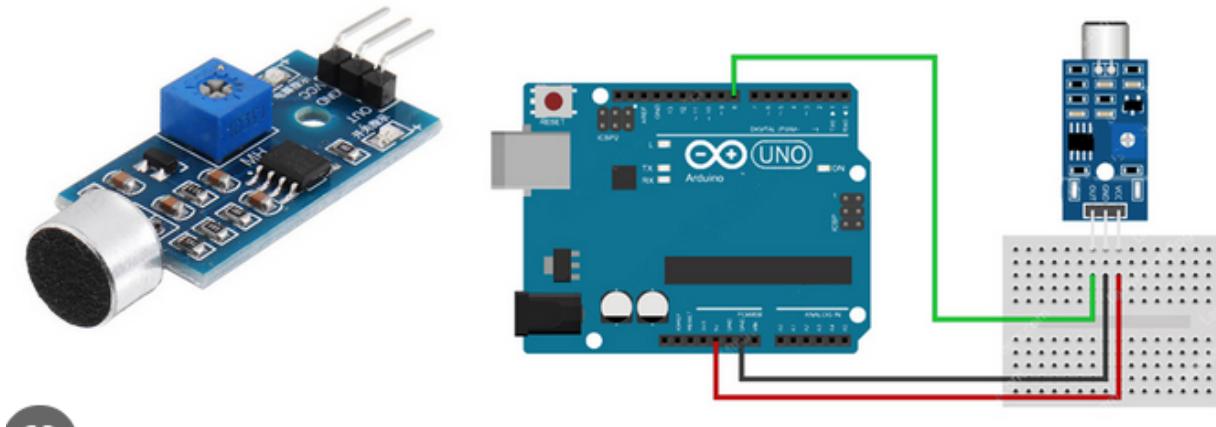
Utilizzato per estendere la capacità di erogare corrente, indispensabile al funzionamento dei Led



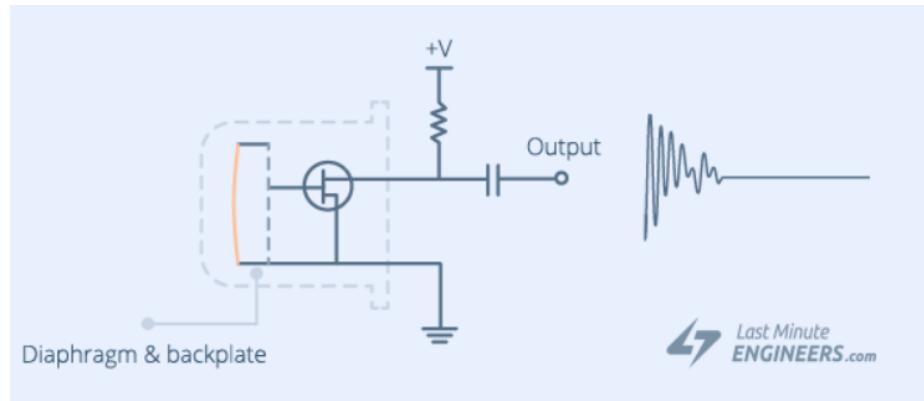
[Link](#) e specifiche.

SOUND SENSOR

A sound sensor is a simple, easy-to-use, and low-cost device that is used to detect sound waves traveling through the air. Not only this but it can also measure its intensity and most importantly it can convert it to an electrical signal.



Funzionamento



When you speak into the microphone, your voice generates sound waves that strike the diaphragm, causing it to vibrate.

When the diaphragm vibrates in response to sound, the plates move closer or farther apart, causing the capacitance to change. As a result, a voltage is generated across the plates, which we can measure to determine the amplitude of the sound.

E' uno strumento utile per il rilevamento di suoni. Esso possiede una **threshold** regolabile attraverso un *sensitivity adjustment*. Se la rilevazione di un suono supera una certa soglia, viene emesso un digital output ALTO, altrimenti BASSO.

Non si presta a misurare il livello di rumore, ma se c'è stata una variazione di intensità di suono.

Wiring

VCC Pin: Linea 5V

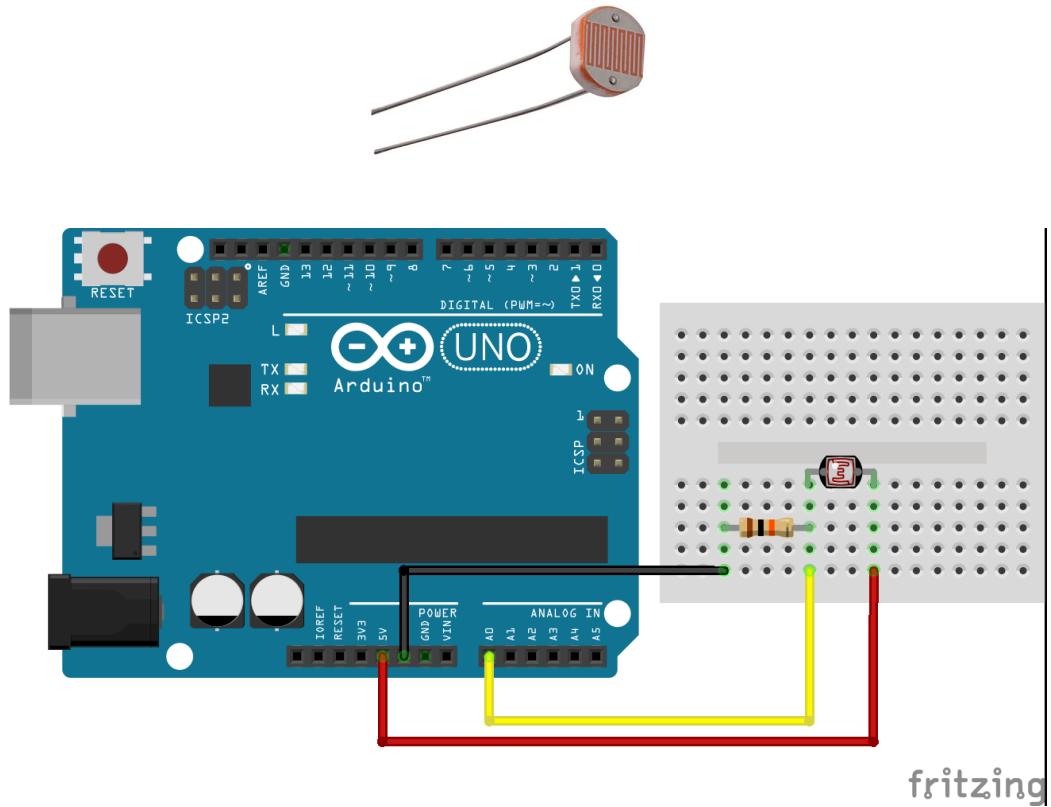
GND: Linea Ground

DATA: Pin 13

PHOTORESISTOR

A photoresistor or photocell is a light-controlled variable resistor. The resistance of a photoresistor decreases with increasing incident light intensity. A photoresistor can be applied in light-sensitive detector circuits, and light- and dark-activated switching circuits. It's also called light-dependent resistor (LDR).

La funzione all'interno del dispositivo è quella del sensore di luminosità. La lettura dei dati avviene in modo analogico.



Wiring

VCC Pin: Linea 5V

GND: Linea Ground

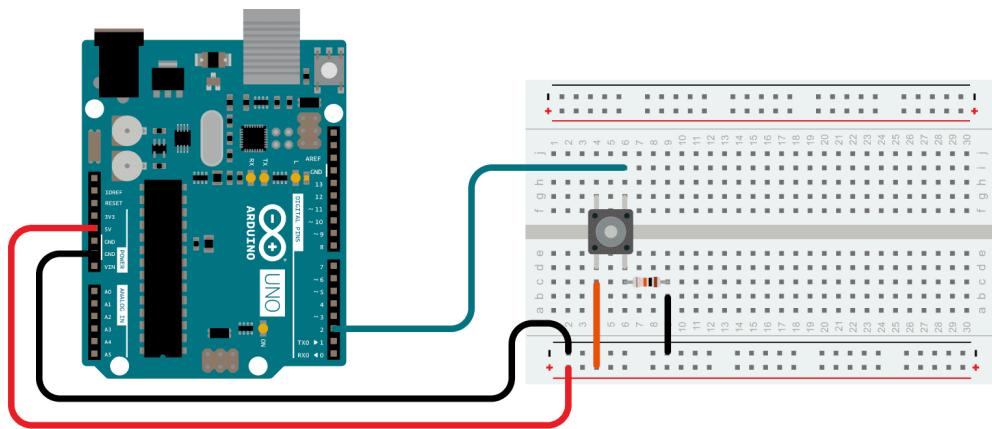
DATA: Pin A0

Resistenza: 100 Ohm

BUTTON

Il bottone è stato scelto per dare prova di poter pilotare la board anche senza l'utilizzo del remoto.

Esso è utilizzato per cambiare lo stato della stanza, e quindi lo stato di precisi altri attuatori [come da FSA]



Wiring

VCC Pin: Linea 5V

GND: Linea Ground

DATA: Pin 9

Resistenza: 10K Ohm

BREADBOARDS

Sono state utilizzate due breadboards:

1. Una principale grande
2. Una piccola per sostenere il sensore ultrasonico.

Diagramma degli stati del dispositivo

Il dispositivo a regime prevede tre stati:

- **S1_OFF**

Il dispositivo è in modalità IDLE poiché la stanza non risulta occupata.

- **S2_WAITING**



Il dispositivo si attiva poiché un utente ha occupato la stanza dall'app, ma non ha ancora attraversato la soglia.

- **S3_ENTERED**

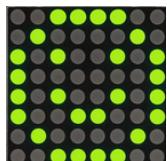
L'utente è entrato in stanza, ed essa può attivarsi secondo i dati pervenuti dal cloud.

Ogni passaggio di stato avviene secondo determinati segnali di trigger, ed ogni stato possiede una configurazione di uscita degli attuatori.

USCITE

USCITE LED MATRIX

U_MX0: *MATRIX_OFF*



U_MX1: *BITMAP_SMILING*



U_MX2: *BITMAP_ALIEN_ANIMATION*

USCITE LED COLOR

Le uscite in questo caso rappresentano le 9 tipologie di colori e l'uscita SPENTO.

U_LC1: NONE

U_LC6: TEAL

U_LC2: RED

U_LC7: BLU

U_LC3: ORANGE

U_LC8: INDIGO

U_LC4: YELLOW

U_LC9: VIOLET

U_LC5: GREEN

U_LC10: RAINBOW

USCITE LED BRIGHTNESS

Le uscite sono 3: LOW, MEDIUM, HIGH, a significare l'intensità luminosa

U_LB1: LOW



U_LB2: MEDIUM

U_LB3: HIGH

USCITE LCD TEMPERATURE

Le uscite dell'LCD sono tante quanti sono i gradi di temperatura impostabili. Raggrupperò quindi in un'unica uscita standard.

U_LCD1#: "Temperatura: # C°"

Vi sono inoltre altre due uscite

U_LCD2: "OFF"

U_LCD3 : "WAITING"

INGRESSI

Gli ingressi che possono triggerare un cambiamento di stato sono i seguenti:

I1: Ricezione da seriale il comando \x0\x0\xff [Segnale di occupazione stanza]

I2: Ricezione da seriale del comando \x0\x01\xff [Segnale di rilascio stanza]

I3: Segnale rilascio bottone

I4: Segnale trigger sensore ultrasonico

I5: Ricezione da seriale comando modifica colore led con parametro #NEW_COLOR

I6: Ricezione da seriale comando modifica luminosità led con parametro #NEW_BRIGHTNES

I7: Ricezione da seriale comando modifica temperatura con parametro
#NEW_TEMPERATURE

USCITE ASSOCIATE AGLI STATI

S1_OFF

→ U_MX0

→ U_LCD2

→ U_LC1

→ U_LB1

S2_WAITING

→ U_MX2

→ U_LCD3

→ U_LC1

→ U_LB1

S3_ENTERED

→ U_MX2

→ U_LCD1#

→ U_LC#

→ U_LB#

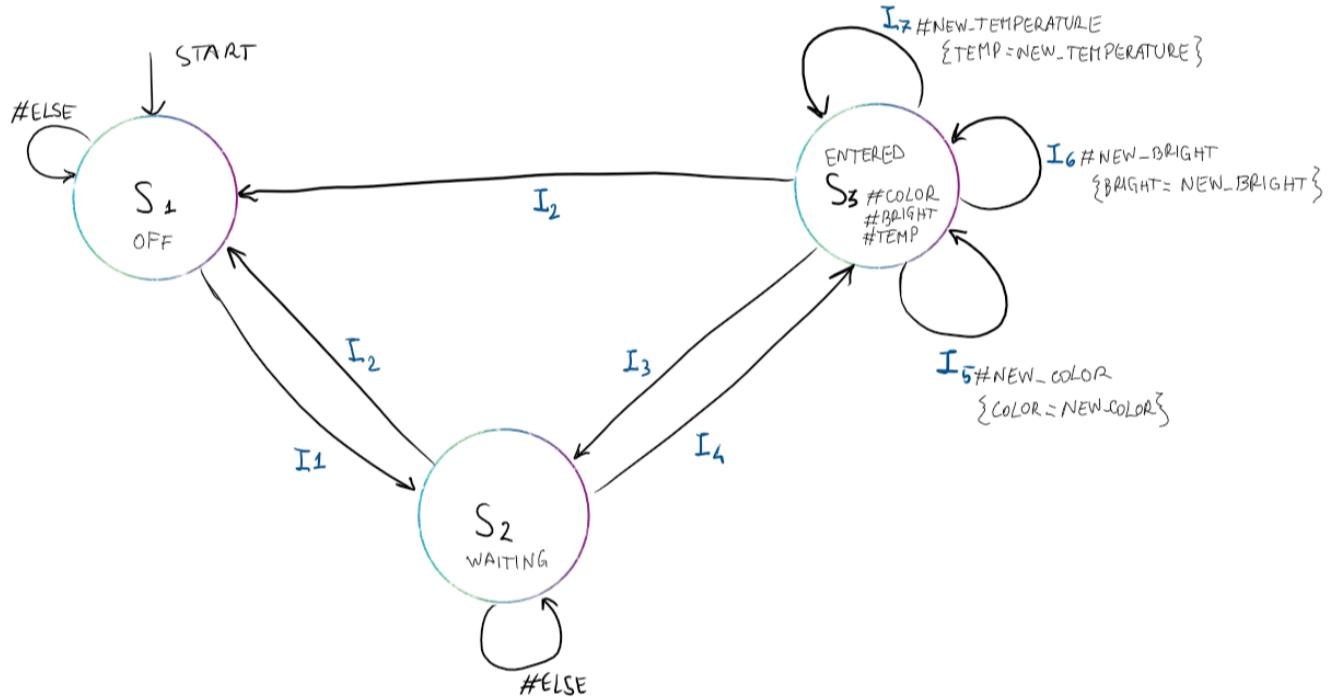
La wildcard # indica il valore CURRENT selezionato dall'utente per ogni attuatore.

FSA DIAGRAM

Per semplicità di rappresentazione, si è scelto di rappresentare lo stato S3 come insieme di tutte le possibili combinazioni COLOR/BRIGHTNESS/TEMPERATURE. Ogni ingresso di modifica di ciascun parametro riporta l'automa nello stesso stato, ma con i suoi parametri interni modificati.

Esempio: CurrentColor = NewColor

Concettualmente dovrebbe essere uno stato distinto per ogni combinazione di uscite, ma siccome le uscite sono indipendenti tra loro, non fa molta differenza in termini di significato.



Protocollo comunicazione seriale

SENSORS DATA SENDING PROTOCOL (MCU to BRIDGE)

START	ID_SENSOR	DATA_SIZE	DATA	END
1 Byte \\xFF	1 Byte	1 Byte	N Bytes as specified in DATA_SIZE	1 Byte \\xFE

SENSORS LIST

[ID1#0] LIGHT SENSOR
 [ID1#1] NOISE SENSOR

COMMANDS DATA SENDING PROTOCOL for ACTUATORS (BRIDGE to MCU)

ID_ACTUATOR	COMMAND	END
1 Byte	1 Byte	1 Byte \\FF

COMMANDS LIST

[ID0#0] LED MATRIX SIGNAL

[ID0#0]	BITMAP_SMILE
[ID0#1]	BITMAP_ALIEN_ANIMATION

[ID0#1] LED STRIP COLOR SIGNAL

[ID1#0]	NONE
[ID1#1]	RED
[ID1#2]	ORANGE
[ID1#3]	YELLOW
[ID1#4]	GREEN
[ID1#5]	TEAL
[ID1#6]	BLUE
[ID1#7]	INDIGO
[ID1#8]	VIOLET
[ID1#9]	RAINBOW

[ID0#2] LED STRIP BRIGHTNESS SIGNAL

[ID1#0]	LOW
[ID1#1]	MEDIUM
[ID1#2]	HIGH

[ID0#3] LCD TEMPERATURE SIGNAL

[ID1#TEMPERATURE_VALUE:INT]

Descrizione codice Bridge

Il bridge utilizza la librerie `paho-mqtt` e `serial` per gestire le comunicazioni.

Vengono effettuate le connessioni al seriale (COM5) e al broker MQTT. Lo script riceve dai topic a cui ha sottoscritto i comandi, li decodifica e traduce e li invia in seriale. Inoltre, riceve dati da seriali riguardanti i sensori ed effettua il meccanismo inverso, pubblicando i dati sui topic adeguati. A tale scopo è utilizzata la classe Bridge, che a regime resta in attesa dei segnali.

Il codice può essere visionato al path `\iot-projectino\arduino_mqttBridge`.

Descrizione codice MCU

Il microcontrollore esegue essenzialmente le routine atte ad implementare la struttura a stati sopra descritta. Ha il compito di ricevere, costruire ed inviare pacchetti seriali.

Parte del lavoro consiste nel pilotaggio di basso livello degli attuatori. A tale fine sono state utilizzate librerie fornite dai produttori dei pezzi utilizzati, o librerie custom offerte da terzi.

Librerie

- **Adafruit LED Backpack Library** by Adafruit
- **Adafruit NeoPixel** by Adafruit
- **Adafruit BusyIO** by Adafruit
- **Adafruit Unified Sensor** by Adafruit
- **Ultrasonic** by Erick Simoes

Il codice può essere visionato al path [\iot-projectino\arduino\main](#).