



UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI  
INGEGNERIA INFORMATICA,  
MODELLISTICA, ELETTRONICA  
E SISTEMISTICA

DIMES

## **Report Tecnico**

# **Dispositivo per il monitoraggio indoor della qualità ambientale e inoltro dati tramite SMS**

*Corso di Progettazione Industriale*

*Corso di Laurea Magistrale in Ingegneria Elettronica*

*Unical, aa 2020/2021*

Andrea Alecce Matricola 214611

Prof. G. Cocorullo

# Sommario

<b>1. Introduzione</b>	<b>3</b>
<b>2. Struttura del progetto</b>	<b>3</b>
<b>3. Ricerca, ideazione e formalizzazione dei processi di sviluppo del progetto.</b>	<b>4</b>
3.1. <i>Stato dell'arte</i>	4
3.1.1. <b>UNIFI Indoor air quality (IAQ)</b>	4
3.1.2. <b>UpSense Air, Air+ e Wave</b>	4
3.1.3. <b>Airthings Wave Plus</b>	4
3.2. <i>Definizione dei servizi, tecnologie, strategie e metodi utilizzati in ambito commerciale</i>	5
<b>4. Definizione delle specifiche e progettazione del sistema hardware</b>	<b>5</b>
4.1. <i>Definizione delle specifiche funzionali, prestazionali e di sicurezza del dispositivo</i>	5
4.2. <i>Selezione delle tecnologie</i>	6
4.2.1. <b>Sensore di Temperatura</b>	6
4.2.2. <b>Sensore di Gas</b>	6
4.2.3. <b>Sensore di Movimento</b>	7
4.2.4. <b>Led RGB</b>	7
4.2.5. <b>Display</b>	8
4.2.6. <b>Buzzer</b>	8
4.2.7. <b>Microcontrollore</b>	9
4.3. <i>Progettazione dell'elettronica e del firmware</i>	10
4.3.1. <b>Test Sensore di Temperatura e Umidità DHT22</b>	10
4.3.2. <b>Test Sensore di Gas MQ-5</b>	11
4.3.3. <b>Test Sensore di Movimento PIR</b>	14
4.3.4. <b>Test Led RGB</b>	15
4.3.5. <b>Test Display</b>	16
4.4. <i>Analisi consumi</i>	18
4.5. <i>Visualizzazione dinamica dei dati sul display</i>	18
4.6. <i>Gestione GPIO</i>	19
4.7. <i>Pin Change Interrupt</i>	20
4.8. <i>Sistema anti rimbalzo (debounce)</i>	21
4.9. <i>Realizzazione Prototipo</i>	23
4.10. <i>Progettazione circuito</i>	24
4.11. <i>Progettazione case</i>	28
4.12. <i>Stima costi</i>	31
<b>5. Riferimenti</b>	<b>32</b>

## 1. Introduzione

Obiettivo di questo report è quello di fornire una documentazione tecnica ed esaustiva sui processi, le scelte e le metodologie adottate nella progettazione e realizzazione del dispositivo in oggetto, ovvero un *Dispositivo per il monitoraggio indoor della qualità ambientale e inoltre dati tramite sms*.

L'aria ci fornisce l'ossigeno necessario alla vita. Le attività umane e alcuni materiali utilizzati comunemente possono però rilasciare sostanze nell'aria, alcune delle quali possono causare problemi a persone, piante e animali. La qualità dell'aria può essere espressa dalla concentrazione di diversi inquinanti come monossido di carbonio (CO), anidride solforosa, biossido di azoto e ozono.

L'inquinamento deve essere considerato anche all'interno delle nostre case, uffici e scuole. Alcuni di questi inquinanti possono essere creati da attività al chiuso come fumare e cucinare. Generalmente, si spende molto tempo all'interno delle abitazioni, motivo per il quale bisogna prestare attenzione alle condizioni in cui si vive.

Il progetto prevede quindi l'ideazione e realizzazione di un dispositivo smart adatto per contesti abitativi, uffici, ospedali, palestre, trasporti, scuole, per monitorare la qualità dell'aria interna e in generale dei parametri ambientali ad esso associata.

Il sistema è basato su diversi sensori e sistemi di notifica (visivo e uditivo, locale e in remoto), discussi e approfonditi nei prossimi paragrafi. Altra funzionalità implementata è la notifica da remoto dello stato ambientale tramite SMS, riportando anche l'eventuale assenza di corrente nella rete elettrica domestica.

Il progetto è realizzato in collaborazione con il collega Andrea Migali, che si occuperà dello sviluppo dei sistemi di invio/ricezione di sms e di alimentazione, mentre io (Andrea Alecce) mi occuperò del sistema di controllo e del sistema di acquisizione dati e visualizzazione dei risultati.

## 2. Struttura del progetto

Il progetto è strutturato in alcuni *macro-obiettivi*, a loro volta organizzati in *attività* come riportato di seguito.

- Ricerca, ideazione e formalizzazione dei processi di sviluppo del progetto.
  - Stato dell'arte
  - Definizione dei servizi, tecnologie, strategie e metodi utilizzati in ambito commerciale
- Definizione delle specifiche e progettazione del sistema
  - Definizione delle specifiche funzionali e prestazionali del dispositivo
  - Selezione delle tecnologie
  - Progettazione dell'elettronica e del firmware
  - Studio e progettazione dell'involucro
- Realizzazione del prototipo di dispositivo
  - Realizzazione dell'elettronica
  - Realizzazione del firmware
  - Realizzazione dell'involucro
  - Testing, validazione e integrazione dell'hardware
  - Integrazione, test e validazione dell'intero sistema hardware+software

- Sperimentazione dei servizi e del dispositivo

Le attività saranno riportate in questo report, aggiornandolo passo passo.

### **3. Ricerca, ideazione e formalizzazione dei processi di sviluppo del progetto.**

In questa prima fase ci si è concentrati sulla ricerca dello stato dell'arte di dispositivi di monitoraggio della qualità dell'aria indoor (da qui in poi chiamati *IAQ Device*, *Indoor Air Quality Device*). Questa ricerca e studio è utile per determinare se e quale fascia di mercato possa coprire l'idea progettuale. Inoltre, è fonte e ispirazione per quanto riguarda le tecnologie utilizzate, gli approcci realizzativi, le architetture e le soluzioni adottate.

#### **3.1. Stato dell'arte**

##### **3.1.1. UNIPI Indoor air quality (IAQ)**

Questo dispositivo è capace di misurare fino a sei parametri (temperatura, umidità, concentrazione di CO<sub>2</sub>, concentrazione di VOC, pressione barometrica e intensità della luce ambientale). Si avvia e configura tramite Wi-Fi, offre diverse tipologie di connettività (RS485, Wi-Fi, LoRaWAN) e supporta vari protocolli di comunicazione (Modbus TCP, Modbus RTU, MQTT, REST). Fruibile tramite interfaccia web, per la visualizzazione dei dati misurati e la configurazione del dispositivo. Unica interfaccia locale è un LED multicolore per l'indicazione della qualità dell'aria e dello stato del dispositivo. Il circuito è racchiuso in un box in ABS (IP20). L'alimentazione può essere fornita a 24V@0.5A oppure tramite Micro USB. Le dimensioni sono 120 × 80 × 25 mm. Il prezzo è di circa 154€.

##### **3.1.2. UpSense Air, Air+ e Wave**

UpSense ha realizzato una famiglia di dispositivi per il monitoraggio ambientale, diversificandoli per parametri rilevabili. Upsense Air rileva la presenza di Monossido di Carbonio (CO), Composti Organici Volatili (COV), Temperatura e Umidità. UpSense Air+ rileva la presenza di Anidride Carbonica (CO<sub>2</sub>), Composti Organici Volatili (COV), Temperatura e Umidità. UpSens Wave è un rilevatore di campi elettromagnetici che misura l'intensità del campo elettrico e quindi ti permette di proteggere te stesso e proteggere i tuoi cari dall'elettrosmog. Integra una batteria ricaricabile agli ioni di Litio da 3.7V e capacità 1000mAh ed un display LCD TFT 2.4" touchscreen resistivo. Le dimensioni sono 108 × 74 × 27 mm. Il costo varia da tra i 169€ dell'Air base fino a 245€ per l'Air+.

##### **3.1.3. Airthings Wave Plus**

Questo dispositivo è capace di misurare sei parametri, ovvero Radon, CO<sub>2</sub>, COV, umidità, temperatura e pressione dell'aria. I dati raccolti possono essere consultati tramite app da smartphone, dashboard online oppure tramite led luminoso. Si collega tramite Bluetooth 5.0 oppure Hub dedicato (Airthings SmartLink). Viene alimentato con 2 batterie AA. Le dimensioni sono 12 cm di diametro e 3.6 cm di altezza. Il costo è pari a circa 269€.

### **3.2. Definizione dei servizi, tecnologie, strategie e metodi utilizzati in ambito commerciale**

Dall'indagine condotta sui prodotti attualmente in commercio sul monitoraggio della qualità dell'aria, è possibile individuare alcuni punti in comune:

- Presenza di sensori di temperatura, umidità, gas;
- Sistema di notifica locale;
- Sistema di notifica da remoto;
- Dimensioni contenute.

Sulla base di questi punti si definiscono le funzionalità base di cui il dispositivo in fase di realizzazione sarà necessariamente dotato.

## **4. Definizione delle specifiche e progettazione del sistema hardware**

Lo scopo principale del sistema è il monitoraggio dei parametri ambientali indoor. Ci si concentrerà sui valori di temperatura, umidità, presenza o meno di concentrazioni di gas pericolose per l'uomo. Si ritiene utile l'integrazione di un sensore di movimento per la rilevazione della presenza dell'utente all'interno dell'abitazione, per evitare inutili trasmissioni di dati, che possono essere fruiti in locale tramite display. Nel caso di assenza dell'utente, si utilizzerà il sistema tramite SMS per le notifiche sullo stato dell'ambiente. Inoltre, tale sistema permetterà anche la segnalazione dell'interruzione dell'energia elettrica.

### **4.1. Definizione delle specifiche funzionali, prestazionali e di sicurezza del dispositivo**

Sulla base delle linee guida e delle specifiche generali ricavate precedentemente, saranno definite le specifiche dettagliate del sistema hardware in termini di:

- funzioni che il dispositivo dovrà avere e modalità di interazione con l'utente
- prestazioni delle componenti singole e dell'apparato complessivo
- sicurezza personale e sicurezza dei dati

Per quanto riguarda le funzioni del dispositivo, è previsto il monitoraggio dei parametri ambientali quali temperatura, umidità, concentrazioni di gas nocivo, movimento. Questi parametri saranno misurati dai relativi sensori, elaborati e nel caso di valori inadatti, notificati immediatamente all'utente. Inoltre, sarà implementata la segnalazione dell'interruzione dell'energia elettrica.

Le modalità di interazione con l'utente saranno limitate alla segnalazione dei valori misurati automaticamente o su richiesta. Potranno essere fruibili in locale, oppure tramite SMS da remoto: nel primo caso, si utilizzeranno display e led RGB.

Il dispositivo sarà alimentato costantemente dalla rete elettrica, ma si prevede l'utilizzo di un sistema di backup per garantire la segnalazione di assenza di alimentazione. Per questo motivo, i vincoli sulle prestazioni energetiche dei vari componenti non sono molto stringenti, ma dimensionate al caso di assenza di alimentazione da rete elettrica. Nonostante ciò, il dispositivo non dovrà risultare un "peso" economico sulla bolletta, limitando i consumi tramite scelte progettuali hardware e software.

Si prevede l'integrazione di alcuni accorgimenti per la sicurezza personale nell'utilizzo, come protezione da sovracorrente o sovratensione.

I dati raccolti saranno totalmente anonimi. Non ci sarà possibilità di identificare l'utente e l'unica trasmissione di dati verso l'esterno avviene tramite messaggio SMS all'utente interessato.

## 4.2. Selezione delle tecnologie

Saranno valutati e selezionati i componenti ottimali ai fini dell'integrazione nel progetto hardware, in base a:

- dimensioni fisiche
- consumo energetico
- prestazioni funzionali

### 4.2.1. Sensore di Temperatura

Sono stati selezionati i seguenti sensori di temperatura, di cui si riporta una tabella comparativa:

Sensori Temperatura	n° pin dati	Corrente [mA]	Tensione [V]	Precisione	Costo [€]*
DS18B20	1	1	da 3 a 5	±0.5°C	3.97
DHT11	1	da 0,5 a 2,5	da 3 a 5	±2°C	4.24
DHT22	1	da 1 a 1,5	da 3,3 a 6	±0.5°C	4.23
LM35C	1	0.6	da 4 a 30	±0.2°C	5.59

Figura 1 - Tabella comparativa sensori temperatura

Tutti questi sensori sono digitali e sfruttano un protocollo di comunicazione a un conduttore, definito appunto one-wire.

Da un confronto di questi componenti, si è scelto il sensore di temperatura **DHT22**. Tra le motivazioni, il fatto che implementi sia il sensore di temperatura che quello di umidità (per cui non è richiesto l'uso di uno esterno) come il DHT11 ma, a distinguerlo da quest'ultimo, è la precisione nettamente migliore e il costo praticamente paragonabile.

### 4.2.2. Sensore di Gas

Sono stati selezionati i seguenti sensori di Gas, di cui si riporta una tabella comparativa:

Sensore Gas	n° pin dati	Corrente [mA]	Tensione [V]	Dimensioni [mm]	Costo [€]*
MQ-x	1	160	5	23 (diam) x 20 (h)	5.84
TGS2600-B00	1	42	5	9.2 (Diam) x 7.	21,55
Sensirion SGP40	2 (I2C)	3.5	3.3	2.44 x 2.44 x 0.85	6,61

Figura 2 - Tabella comparativa sensori di gas

Da questo confronto, si è scelto il sensore di gas della famiglia **MQ-x**. Tra le motivazioni si citano il costo inferiore e la semplicità di utilizzo. In particolare, si seleziona il sensore MQ-5 che si caratterizza per una sensibilità al GPL e metano. Nulla vieta di utilizzare un diverso sensore della famiglia, come ad esempio l'MQ-2 che è sensibile al fumo e al gas infiammabile, in quanto questi sensori sono facilmente interscambiabili.

Più nel dettaglio, questi sensori di gas sono realizzati con tecnologia *MOS (Metal Oxide Semiconductor)* e sono chiamati anche *Chemiresistors* poiché la rilevazione si basa sulla variazione della resistenza del materiale semiconduttore quando il gas ne viene a contatto. Quindi, se la concentrazione è alta, la resistenza diminuisce; se la concentrazione è bassa, la resistenza aumenta.

Il principio di funzionamento è abbastanza semplice ed è simile per ogni sensore di gas MQ. L'elemento sensibile *SnO<sub>2</sub>* ha una conduttività inferiore nell'aria pulita rispetto a quando è esposto al gas di cui è sensibile. Inoltre, maggiore è la presenza di gas, maggiore sarà la conducibilità e quindi minore la sua resistività. Ciò avviene perché quando il semiconduttore (*SnO<sub>2</sub>*) viene riscaldato in aria pulita, gli elettroni donori nel biossido di stagno sono attratti verso l'ossigeno che viene assorbito sulla superficie del materiale di rilevamento. Ciò degrada il flusso di corrente elettrica. In presenza di gas, la quantità di ossigeno in superficie diminuisce mentre gli elettroni reagiscono con i gas. Gli elettroni vengono quindi rilasciati nel biossido di stagno, consentendo alla corrente di fluire liberamente attraverso il sensore.

#### 4.2.3. Sensore di Movimento

Sono stati selezionati i seguenti sensori di movimento, di cui si riporta una tabella comparativa:

Sensore Movimento	n° pin dati	Corrente [mA]	Tensione [V]	Costo [€]*
<b>PIR HC-SR501</b>	1	65	5	2.4
Adafruit PIR	1	x	5	8.43
Gravitech PIR Sensor	1	x	da 5 a 12	10.98
Parallax PIR Sensor	1	23	da 3 a 6	12.7

Figura 3 - Tabella comparativa sensori movimento

Da un'analisi sulla quantità di pin richiesta per la comunicazione con il microcontrollore, dalla corrente di esercizio, dalla compatibilità in termini di tensioni e dal costo, si è scelto il sensore di movimento **PIR HC-SR501**.

Più nel dettaglio, il sensore *PIR (Passive InfraRed)* è un sensore a infrarossi passivo in grado di fornire un segnale in uscita alto quando rileva nel suo campo visivo il movimento un corpo. Il principio di funzionamento si basa sul sensore di tipo piroelettrico, ovvero capace di generare un segnale a fronte di una fonte di calore. Tutto ciò che ci circonda emette energia sotto forma di radiazioni luminose.

#### 4.2.4. Led RGB

Si vuole dotare il dispositivo di un'interfaccia visiva tramite LED per le notifiche. Il LED verrà usato in maniera continuativa, per fornire una prima informazione generale sullo stato dell'ambiente, senza usare display, SMS o altro. A tal fine, è stato valutato l'utilizzo di LED RGB, per indicare in maniera chiara e univoca con i diversi colori le diverse informazioni da fornire all'utente.

Sono stati selezionati i seguenti led RGB, di cui si riporta una tabella comparativa:

Led RGB	n° pin dati	Corrente [mA]	Tensione [V]	Dimensioni [mm]	Intensità luminosa (mcd)	Costo [€]*
<b>WS2812B</b>	1	circa 50	da 3.5 a 5.3	5x5	420, 720, 200	0.4404
<b>Kingbright WP154A4SUREQBFZGW</b>	3	30 (per led)	5	5 (Diam)	1000, 1300, 500	1.47
<b>Inolux HV-5RGB60</b>	3	25 (per led)	5	5 (Diam)	600, 1000, 500	0.84

Figura 4 - Tabella comparativa led RGB

Il led *WS2812B* sfrutta una tecnologia integrata, per cui il driver per ogni singolo led è implementato all'interno del dispositivo stesso. In ingresso accetta un segnale digitale opportunamente codificato. Permette una connessione a cascata, fornendo un controllo dinamico di ogni singolo led tramite un unico pin. Ha una intensità luminosa minore, richiede una maggiore complessità software a fronte di una implementazione hardware semplice e un basso costo. Gli altri due componenti sono semplici led RGB a foro passante: differiscono tra loro per intensità luminosa e costo. Richiedono, rispetto al *WS2812B*, più pin per il controllo del singolo colore.

Si sceglie quindi di utilizzare inizialmente *Inolux HV-5RGB60*, valutando successivamente la necessità di avere più pin disponibili ed eventualmente orientarsi sul *WS2812B* che ne richiede meno.

#### 4.2.5. Display

Si vuole dotare il dispositivo di un'interfaccia visiva tramite display per rendere fruibili le informazioni raccolte dal sistema. Il display, quando attivato su richiesta dell'utente o in caso di parametro ambientale fuori range, fornirà in dettaglio informazioni sulla temperatura, umidità, concentrazione di gas.

Sono stati selezionati i seguenti display, di cui si riporta una tabella comparativa:

Display	n° pin dati	Tecnologia	Driver	Risoluzione [pixel]	Tensione [V]	Dimensioni [mm]	Costo [€]*
<b>1602</b>	8 o 12	STN (LCD)	SPLC780D1	5 x 8 (a carattere)	5	80 x 36 x 1.6	7,77
<b>1.3" OLED I2C</b>	2 (I2C)	OLED	SH1106	128 x 64	da 3.3 a 5V	36 x 34 x 4	11
<b>1.8" TFT SPI</b>	4 (SPI)	TFT (LCD)	ST7735	128 x 160	da 3.3 a 5V	58 x 34 x 4	16,90
<b>2.4" TFT SPI Touch</b>	4 (SPI) o 12	TFT (LCD)	ILI9341 (video), XPT2046 (touch)	240 x 320	da 3.3 a 5V	70 x 50 x 15	23,29

Figura 5 - Tabella comparativa display

Il display **1602** è una soluzione versatile ed economica, grazie alla sua interfaccia minimale. Infatti, l'output viene fornito tramite 16 caratteri disposti in 2 righe (da cui il nome). Ha una risoluzione di 5 x 8 pixel a carattere. Per permettere la visualizzazione dell'intero set di caratteri ASCII è necessario usare tutti e 12 i pin dati, per un set ridotto è possibile usarne 8.

**1.3" OLED I2C** è un piccolo display monocromatico con una risoluzione di 126 x 64 pixel, con cui si può interfacciare tramite protocollo I2C.

**1.8" TFT SPI** è un display LCD a colori con una risoluzione di 126 x 160 pixel, con cui si può interfacciare tramite protocollo SPI.

**2.4" TFT SPI Touch** è un display LCD a colori con una risoluzione di 240 x 320 pixel. Implementa una funzione touch con driver dedicato. Ci si può interfacciare tramite protocollo SPI o in modalità 8 bit richiedendo però 12 collegamenti.

Valutati tutti i dispositivi, si sceglie di utilizzare **1.8" TFT SPI** in quanto di dimensioni sufficienti ad una facile visualizzazione, garantita anche dalla risoluzione. È a colori, quindi permette una lettura e interpretazione più chiara del dato visualizzato. Richiede un numero di pin relativamente basso. La funzione touch presente nell'ultimo dispositivo non è fondamentale e può essere agevolmente implementata tramite un pulsante.

#### 4.2.6. Buzzer

Per informare tramite notifica di tipo audio si dota il dispositivo di un buzzer. Il buzzer verrà utilizzato solo all'avvio del dispositivo e nel caso in cui ci sia una variazione, nei parametri ambientali, tale da



essere notificata. Si sceglie di usare un buzzer in quanto semplice da pilotare, economico, poco ingombrante e garantisce un sufficiente volume per essere udito anche a distanza.

Sono stati selezionati i seguenti buzzer, di cui si riporta una tabella comparativa:

Buzzer	Tipologia	Tensione [V]	Corrente max [mA]	Frequenza [Hz]	Pressione Sonora	Dimensioni [mm]	Costo[€]*
<b>CUI DEVICES CMI-9650C-030</b>	Elettromagnetico	da 2 a 5	15	2700/3000	78	Ø9.6 x 5.0	2,07
<b>TDK SD1614T5-B1</b>	Elettromagnetico	da 3 a 6	80	2048	85	16x14	1,36
<b>Kobitone 254-EMB125-RO</b>	Elettromagnetico	da 4 a 7	105	2300± 300	85	12x9,5	1,54
<b>TDK PS1420P02CT</b>	Piezoelétrico	5		2000	70	Ø14 x 11	0,517
<b>Murata PKM22EPPH4007-B0</b>	Piezoelétrico	da 1,5 a 15 (max)		4000	92	Ø22 x 12	0,805
<b>Visaton PB 9.11 - 5 V</b>	Piezoelétrico	da 3 a 7		4100	70	15x11x1.75 (SMD)	1,46

Figura 6 - Tabella comparativa buzzer

Nella ricerca e selezione della migliore soluzione, si è tenuto conto della tipologia (elettromagnetico o piezoelétrico), la tensione di lavoro, la corrente assorbita, la frequenza sonora in grado di generare, le dimensioni e il costo. Sono stati individuati 3 buzzer di tipo elettromagnetico e 3 di tipo piezoelétrico. Come si evince, i piezoelétrici hanno mediamente un costo inferiore. Tuttavia, le dimensioni ridotte rendono appetibili gli elettromagnetici in applicazioni con poco spazio a disposizione.

Si seleziona il modello **CUI DEVICES CMI-9650C-030** per le ridotte dimensioni, la bassa corrente assorbita. Nonostante sia il buzzer più costoso, non si discosta eccessivamente dagli altri esemplari.

#### 4.2.7. Microcontrollore

Al fine di gestire tutti i sensori e attuatori selezionati, è necessario individuare un microcontrollore adatto. Si decide di adottare una soluzione già sviluppata su breadboard per rendere più immediata la realizzazione del sistema, sia in termini hardware che software, a costo di un ingombro maggiore. Le principali specifiche richieste alla scheda con microcontrollore sono principalmente:

- Dimensione contenuta;
- Costo contenuto;
- GPIO sufficienti per gestire le periferiche;
- Compatibile con protocolli di comunicazione utilizzati dalle periferiche;

Sono stati selezionati le seguenti schede con microcontrollore, di cui si riporta una tabella comparativa:

Schede Microcontrollori	MCU	Alimentazione [V]	Tensione Logica [V]	Iout GPIO [mA]	Clock [MHz]	Interrupt	PWM	Comunicazione	Dimensioni [mm]	Costo
Arduino Nano Every	ATMega4809	5, oppure da 7 a 21	5	20	20	Tutti i pin digitali	5 (D3, D5, D6, D9, D10)	UART, SPI, I2C	45x18	9,82 *
Arduino Nano	ATMega328	5, oppure da 7 a 12	5	40	16	D2 e D3	6 (D3, D5, D6, D9, D10, D11)	UART, SPI, I2C	45x18	7,38 **
Teensy 2.0	ATMEGA32U4	5	5	20	16	4 (D5, D6, D7, D8)		UART, SPI, I2C	30x18	17,56 *
Adafruit Feather M0 Basic Proto	ATSAMD21G18	3.3	3.3	20	48	Tutti i pin digitali	Tutti i pin digitali	UART, SPI, I2C	51x23	16,90 *
Adafruit Feather 32u4 Basic Proto	ATmega32u4	3.3	3.3	20	8	4 (0, 1, 2, 3)	Tutti i pin digitali	UART, SPI, I2C	51x23	16,90 *
Adafruit Feather 328P	ATmega328p	3.3	3.3	20	8	2 (2, 3)		UART, SPI, I2C	51x23	10,59 *
STM32F031K6	ST32F0 Arm Cortex M0	3.3, 5 oppure da 7 a 12	3.3	25	48	Tutti i pin digitali		UART, SPI, I2C	50x18	9,76 *
Sparkfun Pro Micro	ATmega32U4	5	5	20		5 (0, 1, 2, 3, 7)		UART, SPI, I2C	32x18	16,90 *
SparkFun SAMD21 Mini	ATSAMD21G18	da 3.5 a 6.0	3.3	10	48	Tutti i pin digitali		UART, SPI, I2C	10x7	17,74 *

Figura 7 - Tabella comparativa MCU

Sebbene si sia inizialmente selezionato *Arduino Nano Every*, si decide di utilizzare *Arduino Nano* in quanto già a disposizione.

Le motivazioni che hanno portato alla scelta di Arduino Nano (o Nano Every) sono un *trade of* tra:

- Alimentazione relativamente flessibile;
- Tensione logica compatibile con i sensori/attuatori in uso;
- Interrupt sufficienti agli scopi progettuali;
- Dimensioni contenute;

- Costo minore.

Ciò che ha influenzato maggiormente la scelta è il costo e le dimensioni.

### 4.3. Progettazione dell'elettronica e del firmware

Si illustra lo schema a blocchi di massima previsto per il sistema. Questo approccio alla progettazione *bottom-up* permette, una volta definiti i sottosistemi e i relativi componenti, di determinare alcune specifiche interne al sistema come il numero di pin GPIO necessari, le correnti richieste al sistema di alimentazione etc.

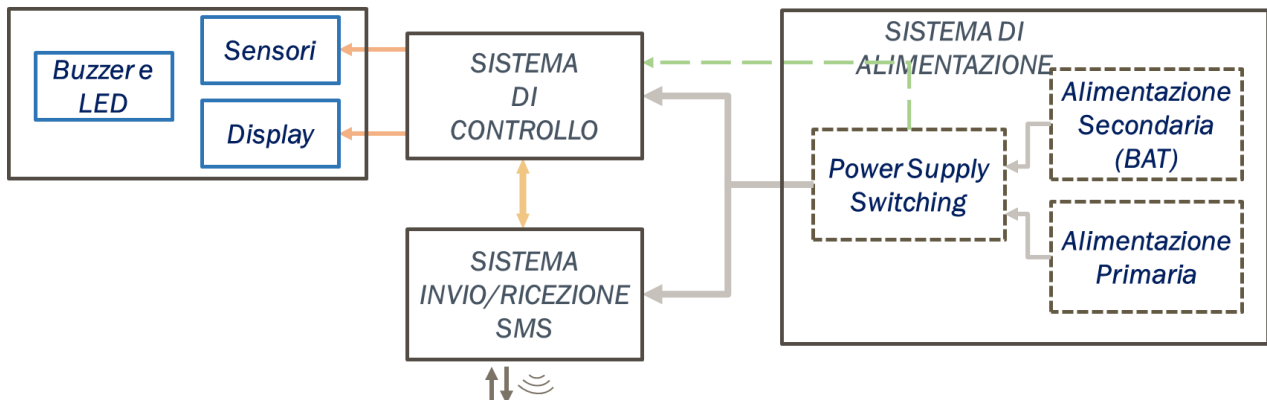


Figura 8 - Schema a blocchi sistema complessivo

#### 4.3.1. Test Sensore di Temperatura e Umidità DHT22

Il sensore di temperatura e umidità selezionato è il DHT22. Questo sensore ha 3 pin per Vcc, GND e dati. Per il segnale dei dati, di tipo digitale, è richiesto un resistore di pullup. Si realizza un semplice circuito di test per il dispositivo. Per il controllo, si utilizza un Arduino Nano. Il test prevede l'acquisizione della temperatura e umidità misurata dal sensore e la relativa visualizzazione.

Una volta cablato il circuito su breadboard, si procede alla scrittura di un semplice firmware in C nell'ambiente di sviluppo IDE di Arduino. L'interfaccia con il sensore è facilitata dall'utilizzo della libreria *DHT.h*. Si definisce il tipo di sensore (DHT22) e il pin digitale utilizzato, dopodiché si istanzia un oggetto *dht* del tipo:

***DHT dht(DHTPIN, DHTTYPE);***

tramite il quale è possibile accedere alle funzioni di lettura dei parametri, come segue:

***float t = dht.readTemperature();***

***int h = dht.readHumidity();***

dai quali si ottengono i valori di temperatura e umidità, istante per istante. Si decide di implementare e testare anche una funzionalità di *memoria minimo e massimo* per i parametri sopra citati. A tal fine, si sfrutta la memoria EEPROM del microcontrollore, utilizzando la libreria *EEPROM.h*. Si confrontano i dati misurati con quelli salvati precedentemente in memoria, sovrascrivendoli se maggiori/minori.

Il test ha permesso di verificare il funzionamento del circuito e l'implementazione software, confermando le scelte fatte.

#### 4.3.2. Test Sensore di Gas MQ-5

Il sensore di gas selezionato è il MQ-5. Questo sensore ha 6 pin: 2 per alimentare l'*heater* a 5V e 2 coppie per gli elettrodi A e B. Vengono usati due coppie di pin per gli elettrodi al fine di attenuare per quanto possibile la resistenza serie parassita di questi collegamenti. Il segnale rilevato in tensione sarà di tipo analogico. Si realizza un semplice circuito di test per il dispositivo. Per il controllo, si utilizza un Arduino Nano.

Il test prevede l'acquisizione della variazione di tensione, dovuta alla variazione di conducibilità misurata dal sensore, la conversione in concentrazione di gas attraverso il grafico fornito nel datasheet e la visualizzazione del dato finale elaborato.

Considerata la complessità nel ricavare l'informazione attesa dal sensore, si decide di utilizzare il seguente workflow:

- Definizione di un modello a partire dallo studio sul principio di funzionamento;
- Analisi matematica del modello;
- Uso dei risultati ottenuti dal modello sul firmware;
- Realizzazione circuitale;
- Calibrazione e test.

Dall'analisi del modello, si ricava l'espressione della resistenza del sensore in funzione di parametri noti:

$$\begin{aligned} V &= V_{RS} + V_{RL} = I \cdot (R_S + R_L) = I \cdot R_S + I \cdot R_L = \\ &= \frac{V \cdot R_S}{R_S + R_L} + V_{OUT} \\ V - V_{OUT} &= \frac{R_S}{R_S + R_L} \cdot V \\ (V - V_{OUT}) \cdot (R_S + R_L) &= V \cdot R_S \\ V_{OUT} \cdot R_S &= V \cdot R_L - V_{OUT} \cdot R_L \\ R_S &= \frac{(V - V_{OUT})}{V_{OUT}} \cdot R_L \end{aligned}$$

Dallo studio del sensore e del suo funzionamento, si ricavano le equazioni che ne descrivono le caratteristiche e sensibilità, come riportato nella seguente tabella:

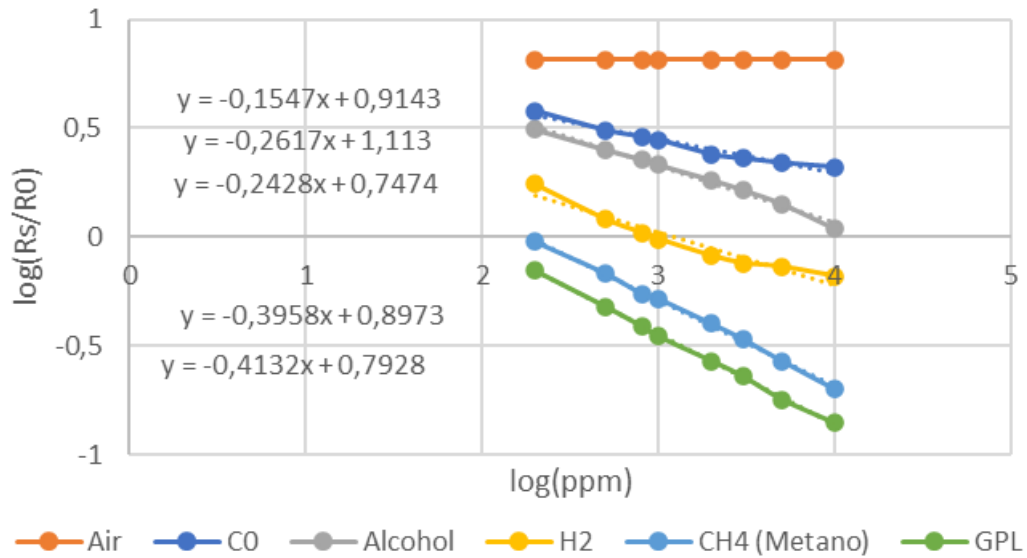


Figura 9 - Grafico delle caratteristiche sensore

Da cui si ricavano i valori del coefficiente angolare e dell'intercetta che approssima la caratteristica del sensore. Questi valori vengono usati nel firmware per ricavare le concentrazioni in ppm dei gas a cui il dispositivo è sensibile.

Al fine di ottenere delle acquisizioni quanto più accurate e veritiere possibile, è necessario procedere ad una prima fase di calibrazione del sensore: dovremo infatti ricavarci il valore di  $R_0$  in condizioni di aria pulita. Questo processo è comunque suscettibile di imprecisioni e non è ripetibile: teoricamente si dovrebbe procedere in ambiente isolato, conoscendone temperatura e umidità, introducendo una concentrazione di gas nota e calibrare il sistema su di essa. Ma per valutazioni nel nostro campo di interesse l'approccio qui seguito è comunque accettabile.

Per la calibrazione è stato scritto un firmware in C illustrato dal seguente *diagramma di flusso*:

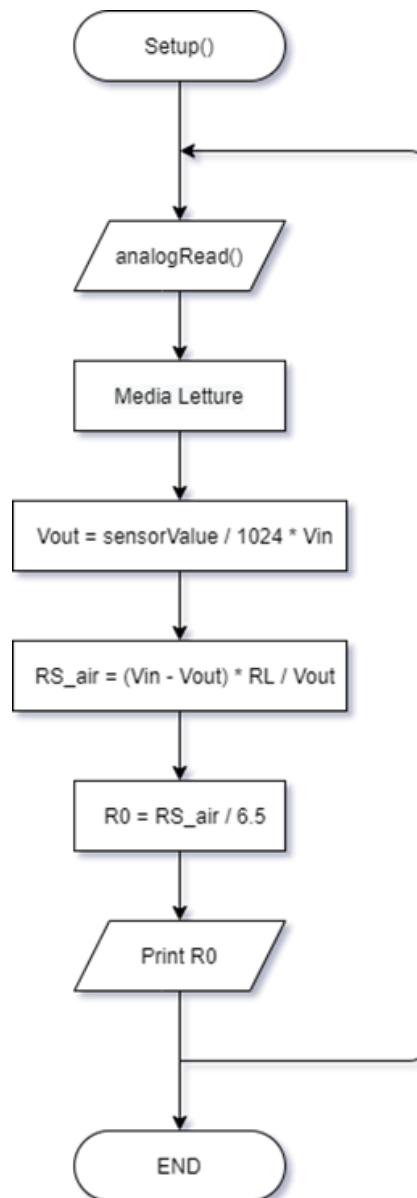


Figura 10 - Diagramma di flusso calibrazione

Inizialmente viene definito il pin al quale abbiamo collegato il sensore MQ-5, oltre che alcune variabili utili ai nostri fini. Inizializzata e impostata la velocità della seriale, si entra subito nel loop nel quale viene effettuata una media su 1000 campioni del segnale in ingresso. Una volta mediato, tale valore viene scalato in base alla risoluzione del microcontrollore e al suo livello logico: i pin analogici lavorano a 10 bit (1024 valori) con logica a 5 Volt. In questo modo si ottiene una misura della tensione  $V_{out}$  dovuta al partitore, precedentemente analizzato. Applicando la formula ricavata, si è in grado di acquisire il valore di resistenza assunta dal sensore.

Successivamente, si usa il risultato ottenuto per la misura effettiva della concentrazione di gas, implementandolo nel firmware illustrato dal seguente *diagramma di flusso*:

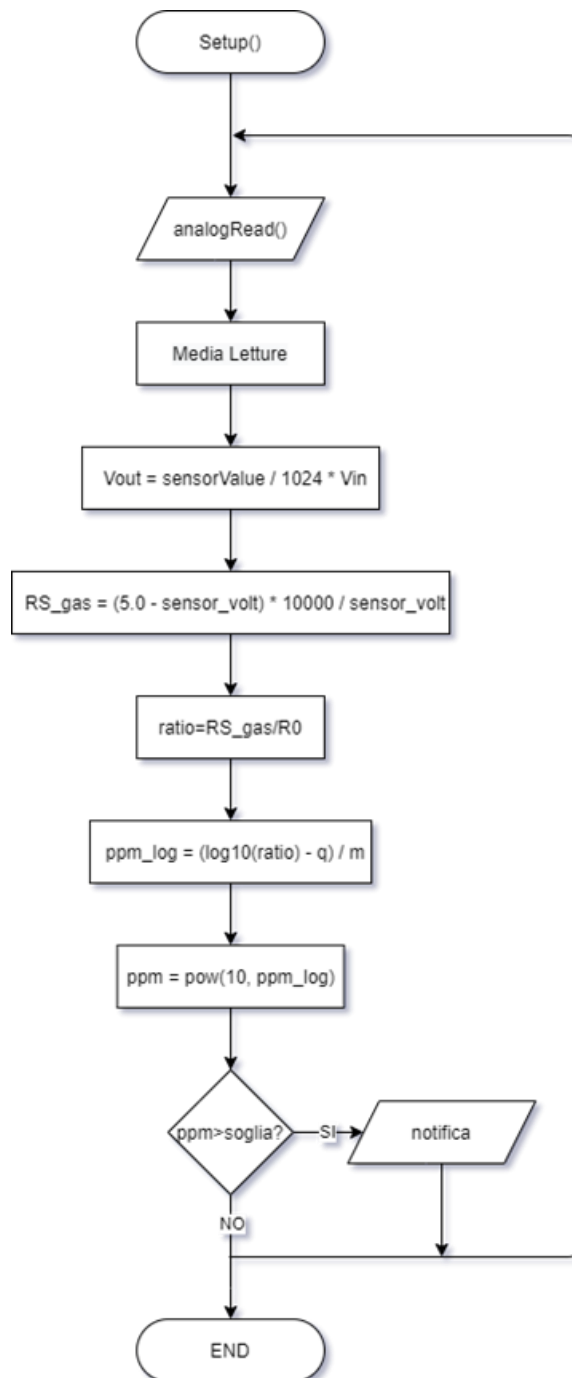


Figura 11 - Diagramma di flusso task

Il test ha permesso di verificare il funzionamento del circuito e l'implementazione software, confermando le scelte fatte.

#### 4.3.3. Test Sensore di Movimento PIR

Il sensore PIR (Passive InfraRed) è un sensore a infrarossi passivo in grado di fornire un segnale in uscita digitale quando rileva nel suo campo visivo il movimento un corpo. Il sensore in questione, modello HC-SR501 è realizzato su una scheda PCB: sul lato superiore si trova il vero e proprio sensore PIR coperto da una lente ottica, sul lato inferiore invece risiedono alcuni componenti e i pin

per l'alimentazione e trasmissione dei dati. Tra i componenti visibili, si evidenziano due potenziometri (trimmer, in arancione) grazie ai quali è possibile variare, a seconda delle esigenze:

- la distanza di sensibilità operativa del modulo, intervallo compreso tra 3-7 m;
- i tempi di uscita del segnale, tra i 3-300 sec.

Il modello HC-SR501 è adatto per essere usato con le schede Arduino, con tensione di alimentazione a 5 volt e logica in uscita a 3.3 volt.

Il test prevede la verifica dell'effettiva rilevazione del movimento e segnalazione al microcontrollore (nel caso in esame, un Arduino Nano). Realizzate le dovute connessioni, si esegue un semplice codice in C per acquisire il dato in digitale fornito dal PIR per accendere un led. Si è reso necessario tarare opportunamente i due trimmer per ottenere una rilevazione a circa un metro con una sufficiente sensibilità.

Il test condotto ha permesso di verificare il corretto funzionamento del sensore, permettendone una taratura ottimale per gli scopi del progetto e ha portato alla realizzazione del codice per il monitoraggio del movimento.

#### 4.3.4. Test Led RGB

Per il Led RGB si analizza la piedinatura (in figura) e si realizza un semplice circuito per il controllo dei colori.

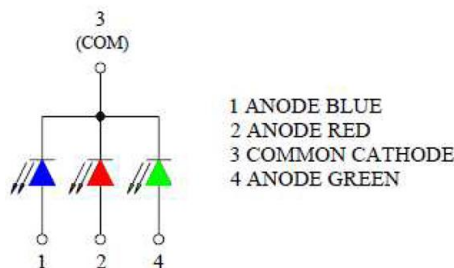


Figura 12 - Pinout led RGB

In figura invece, le caratteristiche elettriche su cui ci si è basati per il dimensionamento dei resistori, necessari per limitare la corrente.

### Electrical and Optical Characteristic ( @ 25 degree C )

Parameter	R	G	B	Test Condition
Luminous Intensity:				
Min (mcd)	1600	2000	1000	If=20mA
Typ (mcd)	3200	4000	2000	If=20mA
Forward voltage (Vf)				
Typ	2.0	3.4	3.4	If=20mA
Max	2.6	3.8	3.8	If=20mA
Dominant Wavelength (nm)	624	525	470	If=20mA
Viewing Angle (Deg)	25	25	25	If=20mA

Parameter	R	G	B	Test Condition
Luminous Intensity:				
Min (mcd)	350	450	250	If=20mA
Typ (mcd)	600	1000	500	If=20mA
Forward voltage (Vf)				
Typ	2.0	3.4	3.4	If=20mA
Max	2.6	3.8	3.8	If=20mA
Dominant Wavelength (nm)	624	525	470	If=20mA
Viewing Angle (Deg)	60	60	60	If=20mA

■ Brightness tolerance = +/- 10%

Figura 13 - Caratteristiche ottico-elettriche led RGB

Dal dimensionamento dei resistori, sono stati ricavati i seguenti valori, supponendo:

- 5V stabili di alimentazione;
- 20mA di corrente per ogni canale RGB;
- le tensioni  $V_{\text{forward}}$  come da datasheet:

$$R = \frac{v_S - v_f}{I_{LED}} = \left\{ \begin{array}{l} R_r = \frac{5V - 2,0V}{20mA} = 150 \Omega \\ R_g = \frac{5V - 3,4V}{20mA} = 80 \Omega \\ R_b = \frac{5V - 3,4V}{20mA} = 80 \Omega \end{array} \right.$$

Realizzato il circuito per i test comprendente le resistenze, si è testato l'effettivo funzionamento e la sufficiente visibilità dei colori.

#### 4.3.5. Test Display

Il display selezionato è un 1.8" TFT SPI. Il test prevede la realizzazione del banco prova di un circuito comprendente il display e un microcontrollore, la scrittura di un codice in C da eseguire su un Arduino Nano e una valutazione sulle informazioni riprodotte a schermo in termini di leggibilità e chiarezza.

Il display utilizza per comunicare il protocollo SPI, per il quale è necessario utilizzare diversi pin:

- CS: chip select pin;
- RST: reset pin;
- DC: data or command selector pin;
- MOSI: SPI Microcontroller Out Serial In pin;
- SCLK: SPI Microcontroller In Serial Out pin.



Mentre per l'alimentazione, è richiesto:

- GND;
- VCC;
- LED: per l'alimentazione della retroilluminazione.

Per l'utilizzo con Arduino Nano è richiesto quindi di individuare i pin mappati nativamente per la comunicazione SPI, facilmente individuabili dal suo pinout.

Effettuati i collegamenti, si passa alla parte software, per la quale si rende necessario l'utilizzo di alcune librerie: in particolare, si utilizza la libreria "*SPI.h*" per comunicare con il display e la libreria "*Adafruit\_ST7735.h*" per interfacciarsi più facilmente con il driver del display. Si istanzia un oggetto *tft* del tipo:

```
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_MOSI, TFT_SCLK, TFT_RST);
```

che può essere richiamato per definire il testo da mostrare, la dimensione, il colore e la posizione sul display.

Dopo qualche semplice prova per verificare il funzionamento del display e come interfacciarsi, si decide di implementare un primo firmware comprendente la visualizzazione dei dati misurati dai sensori precedentemente studiati e testati. Effettuati i dovuti collegamenti, si implementano nel firmware la parte di acquisizione dati dai sensori di temperatura, umidità, gas e movimento, per poi essere mostrati a monitor sul display.

Al fine di rendere più leggibile, flessibile e rendere semplice un eventuale debug del codice, vengono definite alcune funzioni. In particolare, sono state realizzate e definite le seguenti funzioni:

- *getGas()*: acquisisce il dato grezzo, ne fa una media, lo processa e restituisce il valore Rs/R0;
- *conversione()*: per ogni tipo di gas, implementa la relativa caratteristica ricavata da datasheet e fornisce un valore in ppm e un valore percentuale;
- *rect()*: date le dimensioni, il colore e la posizione, crea un oggetto linea visualizzabile su display;
- *textGas()*: date le dimensioni, il colore e la posizione, crea un oggetto testo visualizzabile su display;
- *textTemp()*: stampa su display la temperatura istantanea, la massima e la minima;
- *textHum()*: stampa su display l'umidità percentuale istantanea, la massima e la minima;
- *eraseGas()*: dopo la visualizzazione del singolo gas, è richiesto che venga eliminato per non incorrere alla sovrapposizione di più caratteri;
- *intShow()*: funzione di interrupt, se viene ricevuto un segnale dal sensore di movimento oppure viene acceso manualmente tramite pulsante, avvia la visualizzazione dei dati su display.

Nella seguente figura è mostrato il risultato ottenuto dal test:

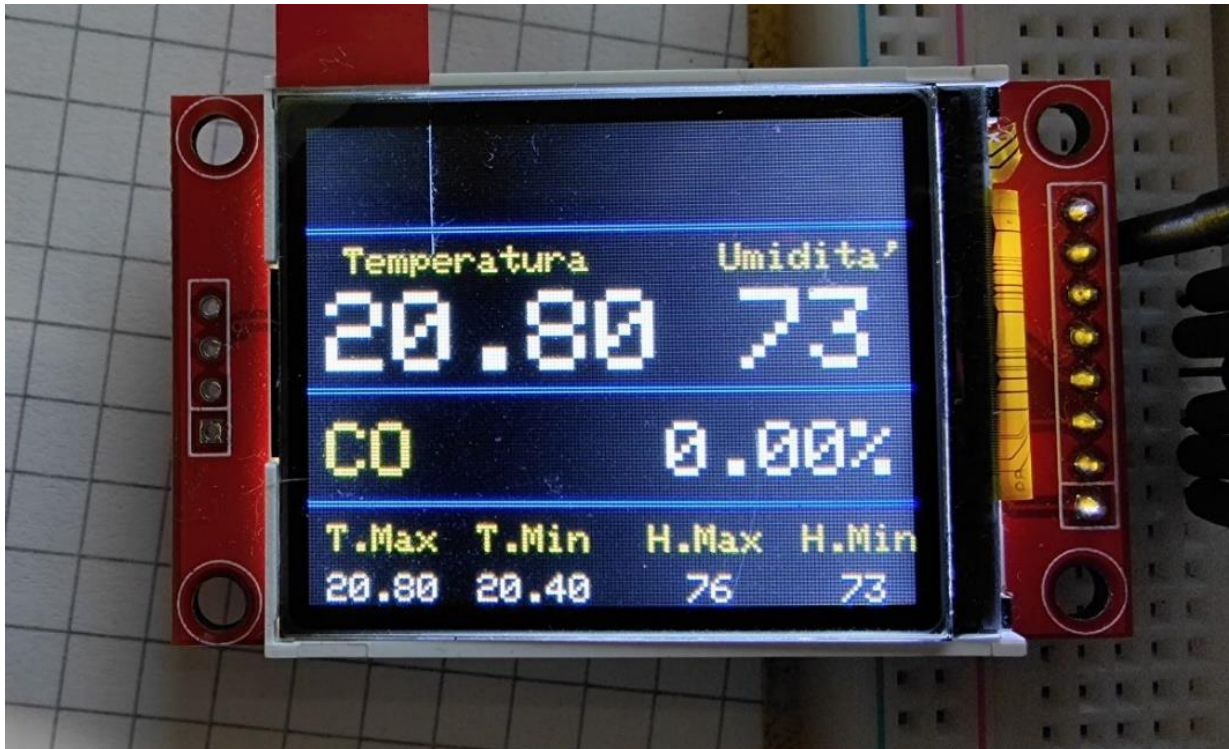


Figura 14 - Schermata display

#### 4.4. Analisi consumi

Per dimensionare lo stadio di alimentazione e la batteria di backup è necessario determinare l'assorbimento in corrente richiesto dal sistema complessivo. Si riporta dunque le correnti in gioco per ciascun componente selezionato, dichiarate sui relativi datasheet e confrontati con quelle effettivamente misurate sul banco prova. Le misure sono effettuate con la stessa strumentazione (multimetro), a regime termico ed elettrico, a distanza di poco tempo.

Assorbimento in corrente	Max da datasheet [mA]	Misurato [mA]
Arduino Nano	20	16.7
DHT22	1.5	1.41
PIR HC-SR501	65	12
MQ-5	160	110
Display 1.8" TFT	50	44.3
Led RGB	25	20,7
Buzzer	15	16.3
<b>Totale</b>	<b>336.5</b>	<b>200.71</b>

Figura 15 - Tabella consumi

Come si evince, la corrente totale assorbita si aggira intorno ai 200 mA.

#### 4.5. Visualizzazione dinamica dei dati sul display

Si decide di implementare una funzione tale da accendere il display e visualizzare i dati solo nel caso in cui si rileva la presenza dell'utente (tramite sensore PIR) o su richiesta esplicita (tramite pulsante). Utilizzando gli interrupt associati a questi segnali, è possibile comandare il pin relativo alla

retroilluminazione del display, permettendone l'accensione o meno. Ciò rende la visualizzazione più dinamica e interattiva con l'utente, oltre che permettere un notevole risparmio in termini energetici e di consumo.

Normalmente il pin della retroilluminazione è posto ad una porzione di circuito in grado di fornire sufficiente corrente, come l'alimentazione o uno stadio dedicato. Non è quindi consigliabile collegare tale pin direttamente al microcontrollore. Si sceglie di utilizzare un semplice mosfet, con funzione di switch on off, per collegare o meno il pin di retroilluminazione all'alimentazione, comandato dal microcontrollore. Uno schema della porzione di circuito è visibile in figura:

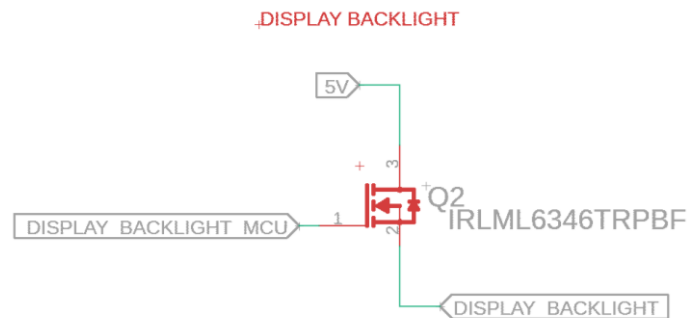


Figura 16 - Circuito controllo retroilluminazione

Si sceglie di non interrompere l'intera alimentazione del display (ad esempio, tramite GND), in quanto interromperebbe la comunicazione SPI con il micro e renderebbe più lenta e dispendiosa la riaccensione.

#### 4.6. Gestione GPIO

La board con microcontrollore selezionata è Arduino Nano, su base ATmega328. Fornisce 8 pin analogici, 22 digitali, 6 dei quali PWM.

Nella seguente figura, il pinout complessivo della scheda in esame.

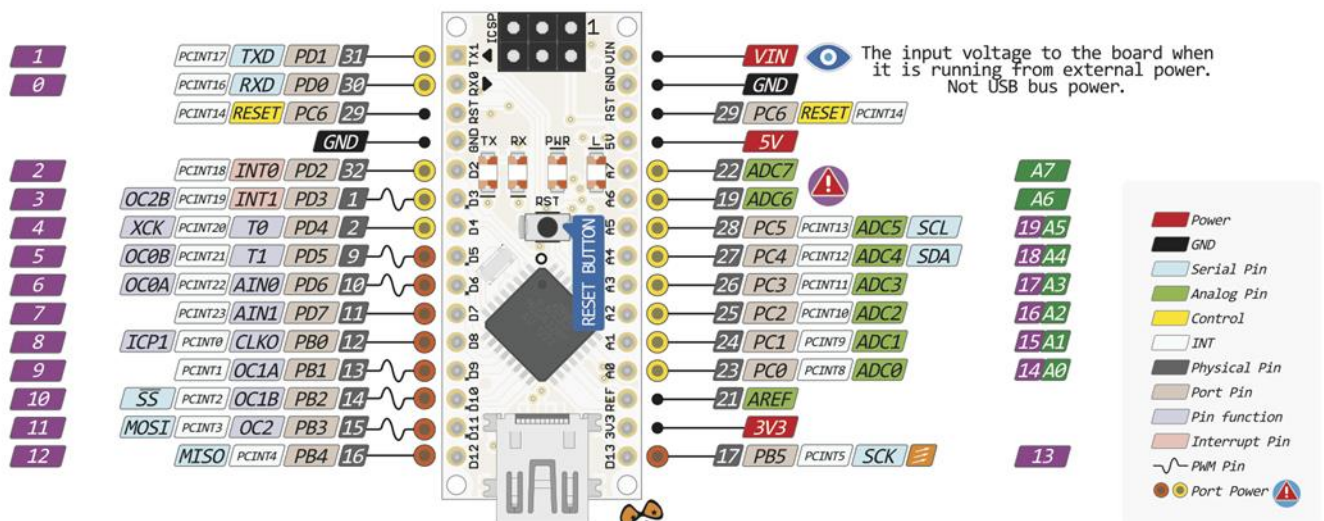


Figura 17 - Pinout Arduino Nano

Sulla base delle analisi e test condotti precedentemente, si individua la seguente associazione ottimale per i terminali dei dispositivi e i pin dell'Arduino Nano.

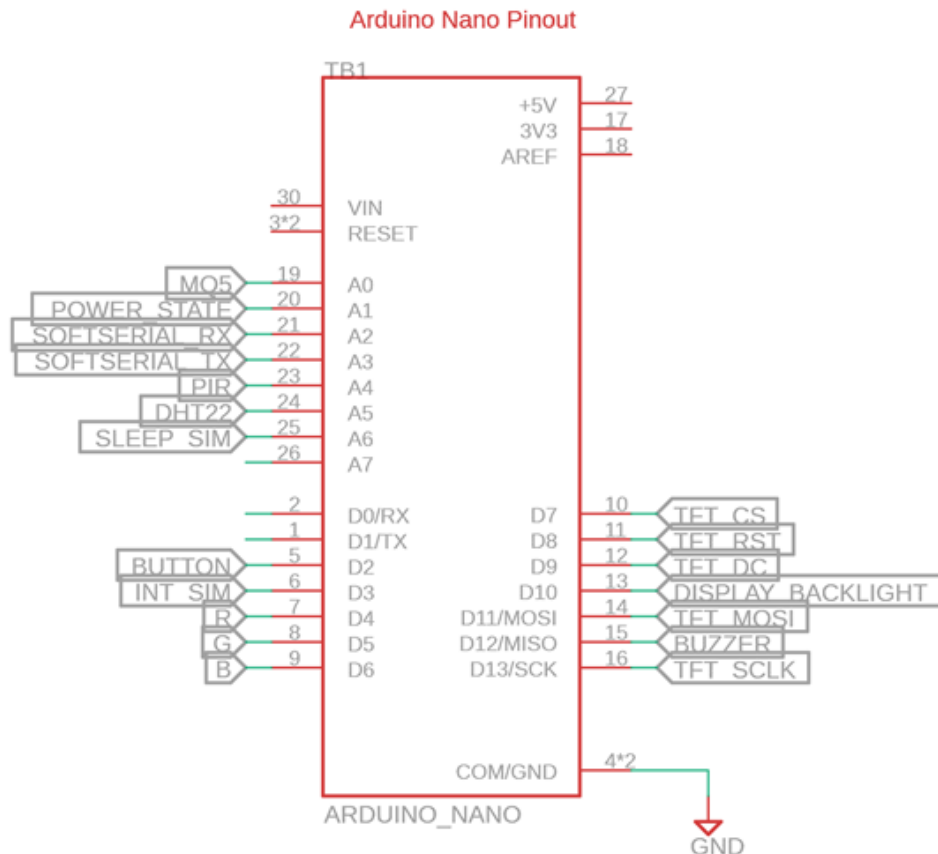


Figura 18 - Schema di collegamento usato per l'Arduino Nano

Come si nota, si è evitato di utilizzare i pin digitali D0 e D1, preposti per la comunicazione seriale, utile quest'ultima in fase di debug. In quanto più stringenti nelle loro funzioni, si è data priorità ai segnali analogici, disposti sui relativi pin. Successivamente sono stati occupati i pin per i segnali di interrupt, che nel sistema saranno:

- dal pulsante fisico;
- dal modulo GSM SIM800L;
- dal sensore di movimento PIR.

Siccome Arduino Nano mette a disposizione nativamente solo due hardware interrupt nativamente, è stato necessario abilitarne altri tramite **Pin Change Interrupt**, come discusso in seguito. L'alimentazione avviene attraverso la linea 5V per l'Arduino come per tutti i dispositivi utilizzati. Successivamente, si effettuano i collegamenti per il display. Sfruttando il protocollo SPI, si è reso necessario utilizzare i pin predisposti sulla board. Infine, si distribuiscono i rimanenti segnali nei pin liberi.

#### 4.7. Pin Change Interrupt

Come introdotto precedentemente, si è reso necessario ampliare il numero di pin utilizzabili come interrupt. Analizzando il datasheet del microcontrollore ATmega328 e mettendo mano nei registri opportuni, è stato possibile sbloccare tutti i pin GPIO con funzionalità interrupt. Ovviamente, ci si è limitati al numero utile per il progetto.

In particolare, facendo sempre riferimento al datasheet del ATmega328, è necessario effettuare 3 passaggi, ovvero:

1. Attivare la porta. Ad esempio:

`PCICR /= 0b00000111; // abilito le porte B, C e D`

PCICR – Pin Change Interrupt Control Register								
Bit	7	6	5	4	3	2	1	0
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Figura 19 - Registro PCICR

2. Attivare i pin di Interrupt. Ad esempio:

`PCMSK1 /= 0b00010000; // PCINT12 (A4)`

PCMSK2 – Pin Change Mask Register 2								
Bit	7	6	5	4	3	2	1	0
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

PCMSK1 – Pin Change Mask Register 1								
Bit	7	6	5	4	3	2	1	0
(0x6C)	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

PCMSK0 – Pin Change Mask Register 0								
Bit	7	6	5	4	3	2	1	0
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Figura 20 - Registri PCMSKX

3. Scrivere una funzione ISR (Interrupt Service Routine) per ciascun Interrupt. Ad esempio:

```
ISR(PCINT1_vect) { // // Port C, PCINT8 - PCINT14
    show = true;
}
```

Una volta implementata questa soluzione, si è testata per verificare l'effettivo funzionamento. Ciò ha permesso di aumentare il numero di pin di interrupt fruibili, consentendo una migliore e più reattiva gestione degli eventi. In particolare, torna utile per la visualizzazione dei dati in locale.

#### 4.8. Sistema anti rimbalzo (debounce)

Come precedentemente discusso, sono stati introdotti alcuni segnali sfruttando gli interrupt. La potenzialità degli interrupt è tale da dover comunque essere gestita e non abusata. Una successione di interrupt in un breve intervallo di tempo potrebbe comportare malfunzionamenti, sovraccarico e accavallamento nella richiesta delle risorse etc.

A tal fine, si decide di studiare e implementare un sistema di debouncing (anti rimbalzo) sui pin interessati. Un tipico segnale affetto da rimbalzo, acquisito tramite oscilloscopio, è visibile in figura:

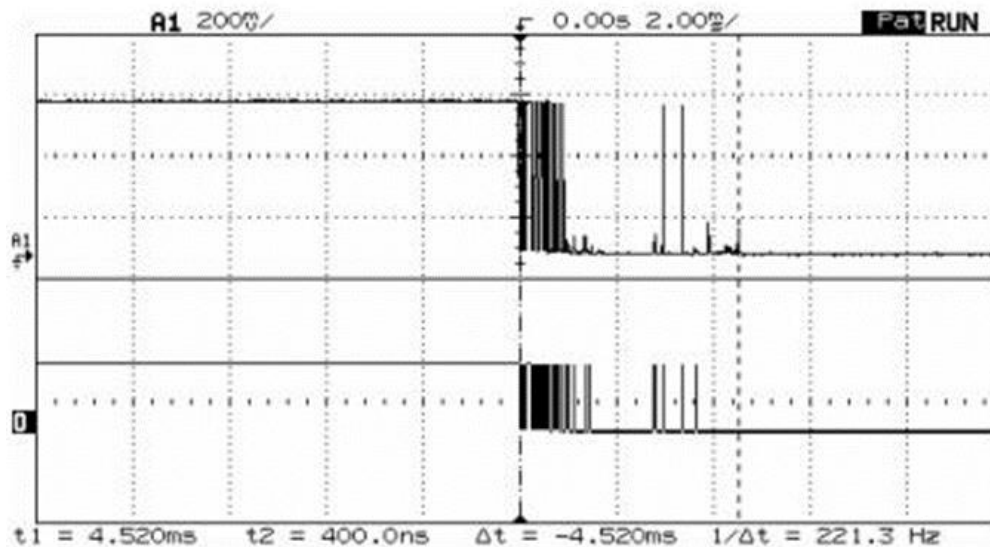


Figura 21 - Esempio di bounce

In generale, questi picchi si riscontrano in elementi meccanici come pulsanti o switch. La loro presenza è totalmente aleatoria, anche se statisticamente elementi usurati, vecchi o malfunzionanti ne sono maggiormente affetti. La frequenza ad essi associata è anch'essa aleatoria, partendo dai nanosecondi fino a pochi millisecondi.

La variabilità e stocasticità di questi eventi rende difficile l'analisi e conseguente dimensionamento di un sistema di debounce. Esistono tuttavia dei metodi genericamente utilizzati per attenuare questa problematica. In generale, un sistema debounce può essere implementato in modo:

- Software;
- Hardware.

Un debounce software si basa sull'uso di un semplice delay appena viene rilevato il segnale, ovvero un intervallo temporale nel quale vengono trascurate tutte le variazioni del segnale. Questa soluzione è generalmente la più usata per semplicità e costo.

Un debounce hardware è invece realizzato fisicamente con una coppia passiva RC, con l'uso di un diodo per velocizzare la carica oppure con un integrato dedicato, come in figura:

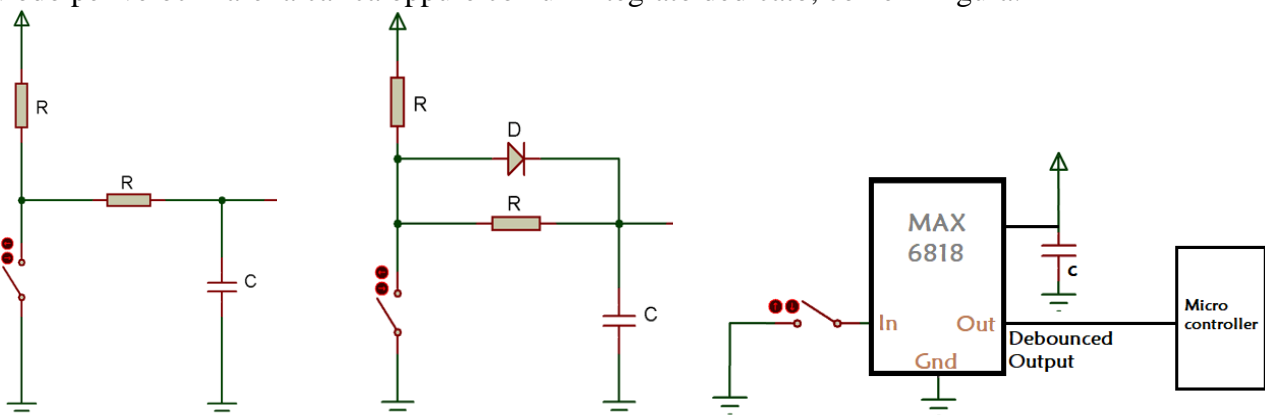


Figura 22 - Implementazioni hardware di sistemi di debounce

Le soluzioni software e hardware non vanno in conflitto e possono essere usate entrambe.



Nel voler utilizzare un sistema hardware, è necessario dimensionare il circuito RC tale da avere una costante di tempo associata paragonabile al segnale affetto da rimbalzo. Ad esempio, supponendo una durata del problema pari a meno di 5ms, bisogna garantire che la tensione sul condensatore stia sopra la soglia alta  $V_{th}$  (per Arduino Nano è pari a 2.6V).

Un esempio di rimbalzo è stato acquisito manualmente tramite oscilloscopio, riportato in figura:

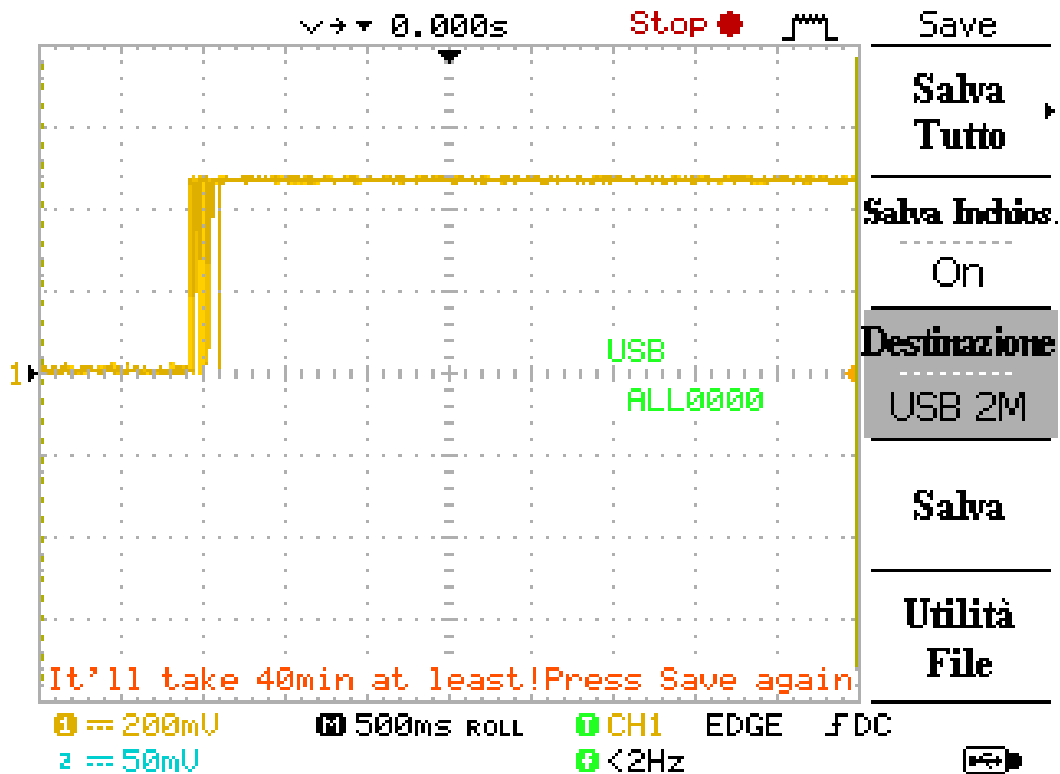


Figura 23 - Rimbalzo catturato tramite oscilloscopio

Si sceglie arbitrariamente il valore di  $C = 1\mu F$  (più limitato in termini di costi, spazio occupato etc. rispetto a  $R$ ) e calcoliamo  $R$  (per  $V_0 = 5V$ ):

$$\tau = R * C \rightarrow R = 5k$$

#### 4.9. Realizzazione Prototipo

Si è realizzata una prima versione prototipale su breadboard per verificare che non ci fossero problematiche elettriche o relative a compatibilità tra i *device*, sovraccarico computazionale, gestione dei vari eventi.

Il sistema è mostrato, in funzione, nella seguente figura:

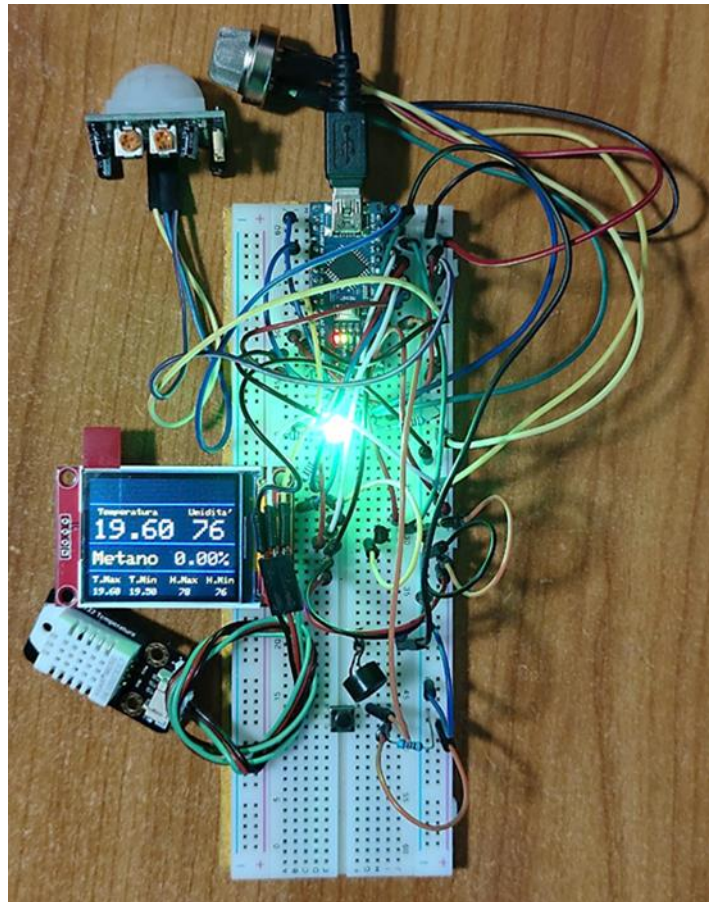


Figura 24 - Prototipo su breadboard

È possibile visualizzare l'intero codice del progetto al seguente repository GitHub: [https://github.com/DeanSupertramp/Progettazione\\_Industriale](https://github.com/DeanSupertramp/Progettazione_Industriale)

#### 4.10. Progettazione circuito

Per la realizzazione della scheda elettronica si rende necessario l'utilizzo di un software EDA (Electronic Design Automation) per la progettazione dello schematico e dello sbroglio del PCB. Sul mercato esistono diverse soluzioni, professionali o meno, con diverse funzionalità come servizi cloud e visione e modifica dei componenti 3D. Alcuni esempi sono *Altium Designer*, *Autodesk Eagle*, *OrCad*, *KiCad* o *EasyEDA*.

Si sceglie di utilizzare Autodesk Eagle in quanto fruibile tramite licenza didattica per gli studenti e offre un sistema professionale alla progettazione elettronica. Inoltre, fa parte dello stesso ambiente Autodesk nel quale è sviluppato Fusion 360, CAD di modellazione 3D utile per la progettazione del case del dispositivo.

Nella seguente figura, si riporta lo schematico complessivo del sistema realizzato:



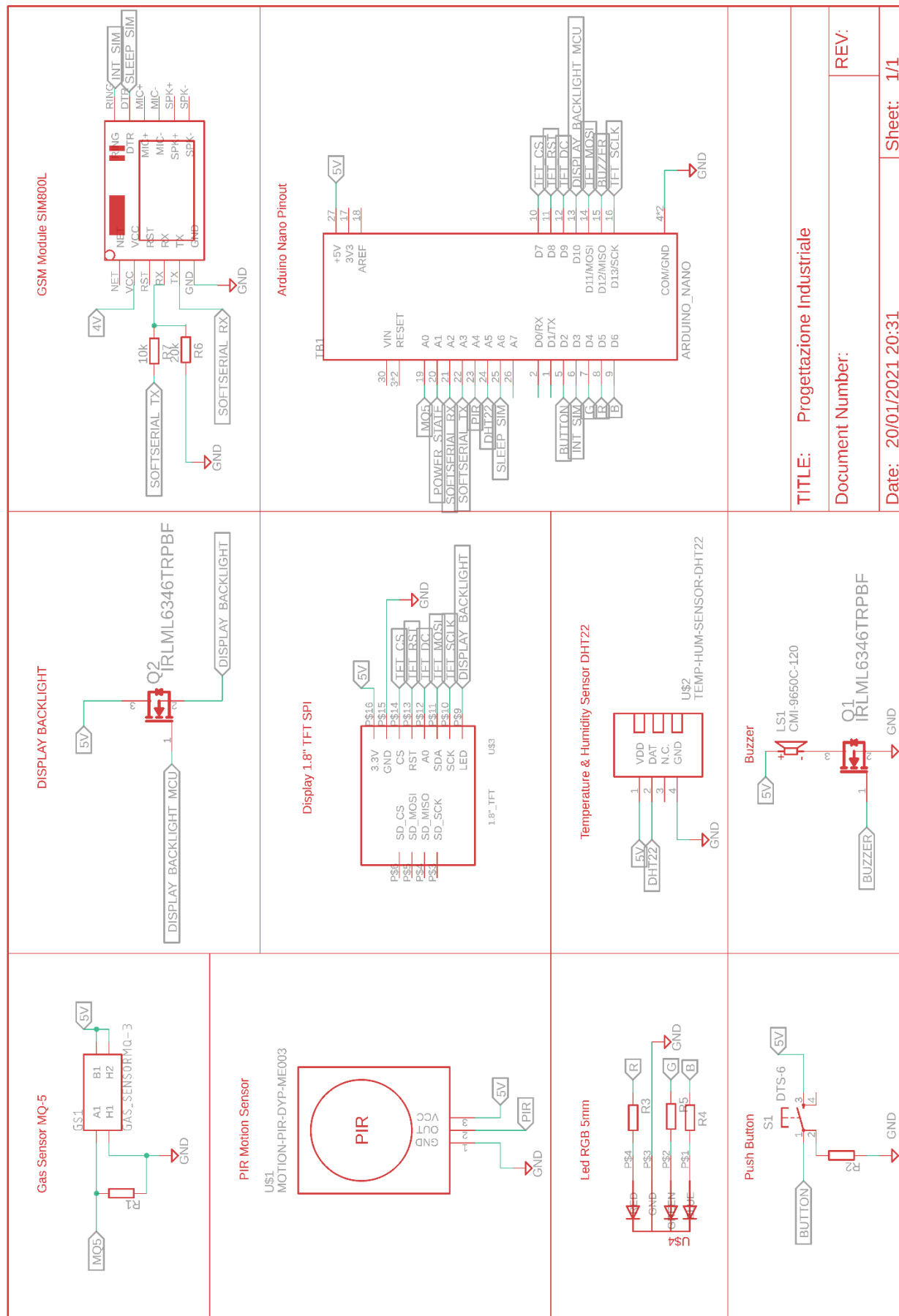


Figura 25 - Schematico complessivo, realizzato su Eagle

Mentre nella seguente figura, lo sbroglio con le dimensioni effettive del pcb, pari a (60.96 x 81.28) mm.

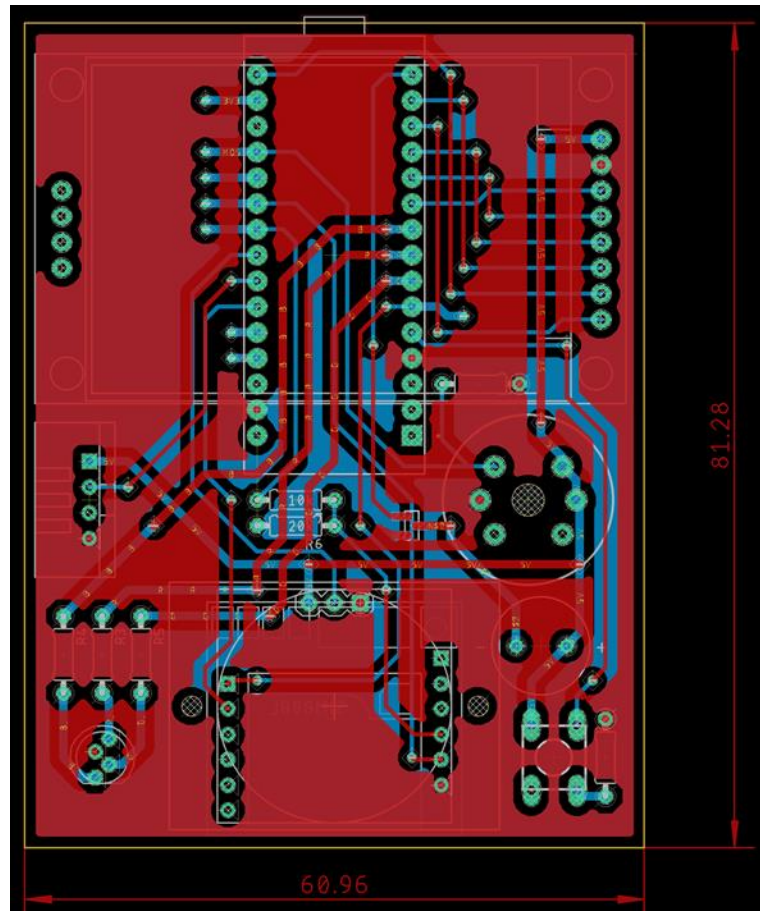


Figura 26 - Sbroglio del circuito, realizzato su Eagle

Si decide progettare e sviluppare il circuito su 2 layer (top e bottom), con thickness 0.8mm, rame 35 micron.

La dimensione delle piste è stata determinata sulla base del carico in corrente a cui sono sottoposte e relativo aumento di temperatura. In generale, sono state adottate delle dimensioni maggiori per alimentazioni e potenza. In particolare, sono state usate due dimensioni:

- 0,4mm per le piste a bassa corrente;
- 0,8mm per le piste ad alta corrente.

Queste dimensioni sono state ottenute ipotizzando:

- Corrente massima di 0,5A;
- Spessore rame 35 micron;
- Aumento temperatura 10°;
- Temperatura ambiente 25°.

Si è seguito lo standard **IPC-2221**. È possibile utilizzare tool online o software per il calcolo delle dimensioni. I passaggi effettuati sono comunque calcolare l'area:

$$Area[mm^2] = \left( \frac{Corrente[A]}{k * (temp_{aumento}[^{\circ}C])^b} \right)^{1/c}$$

Quindi, calcolare la larghezza come:

$$Larghezza[mm] = \frac{Area[mm^2]}{Spessore[mm] * 1,378}$$

Dove si assume da IPC-2221:

- $k = 0,024$ ,
- $b = 0,44$ ,
- $c = 0,725$

c sono costanti risultanti dall'interpolazione della curva alle curve *IPC-2221*.

Dalle specifiche imposte, si richiede per la massima corrente una larghezza delle piste pari a 0.3003865208 mm. Scegliendo una dimensione di 0.8mm, si esclude qualsiasi problematica relativa alle temperature

Di seguito, un esempio di rendering 2D del PCB con vista top e bottom:

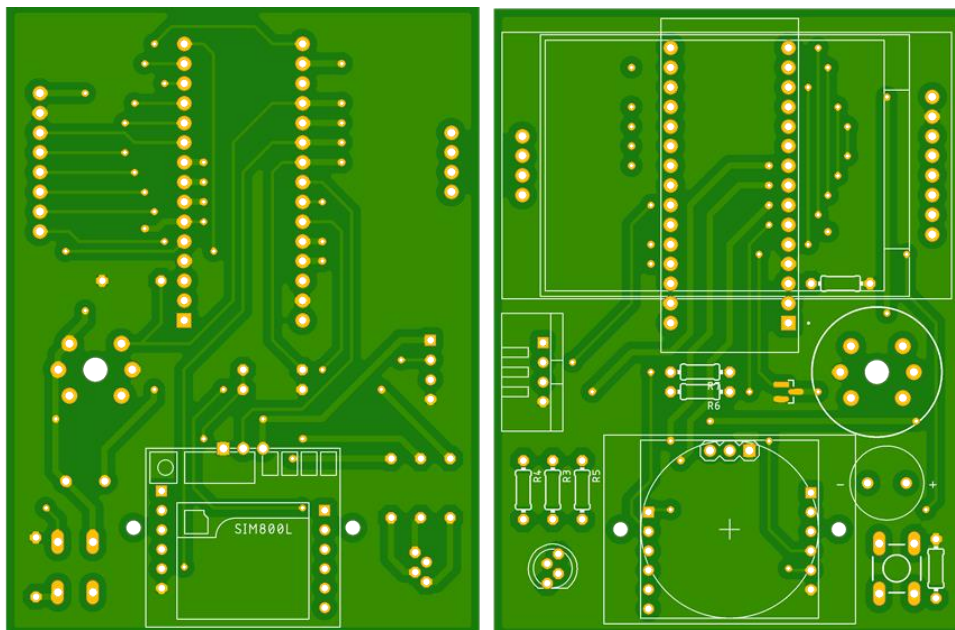
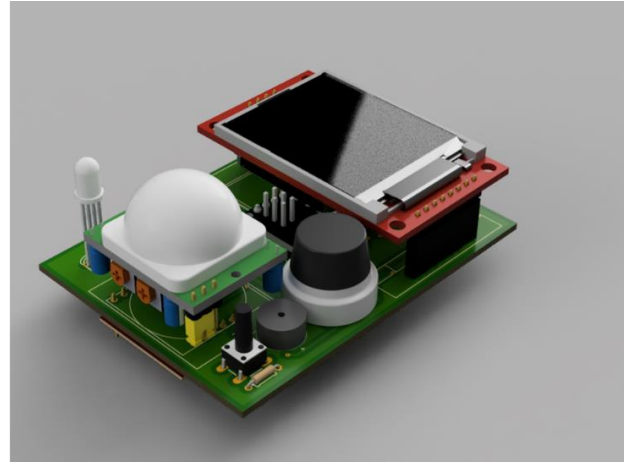
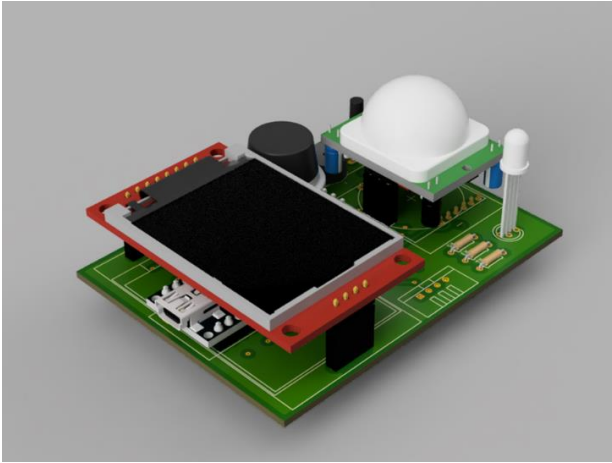


Figura 27 - Rendering 2D del PCB, realizzati su Eagle

Successivamente, tramite *Autodesk Fusion 360*, sono stati inseriti i modelli 3D dei componenti e generato un render del circuito, come riportato di seguito:



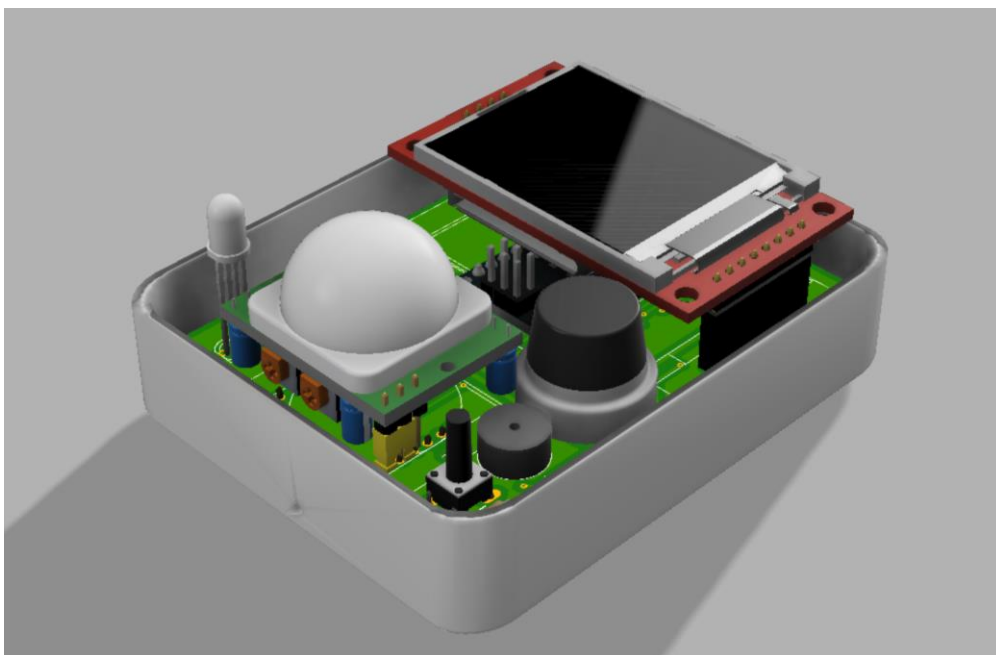
*Figura 28 - Rendering 3D del PCB , realizzati su Fusion 360*

#### **4.11. Progettazione case**

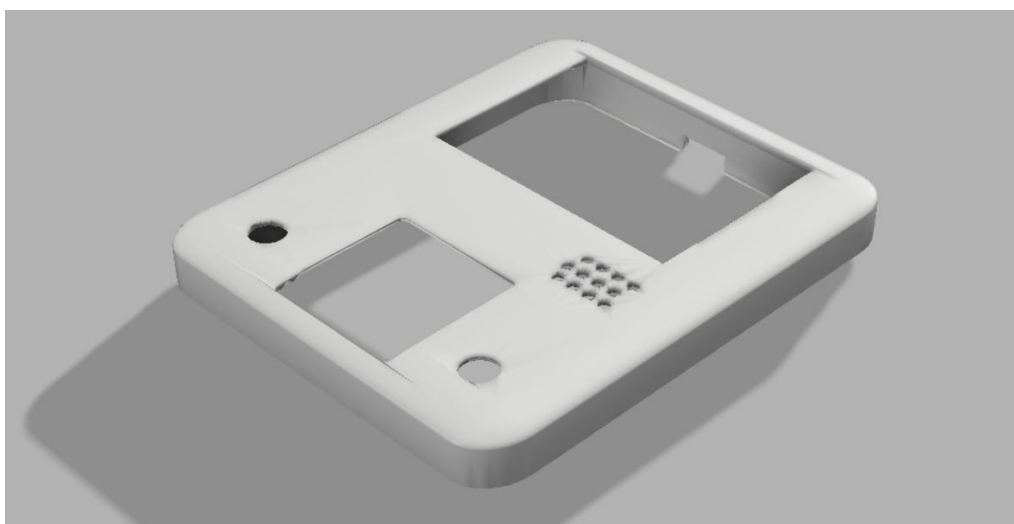
Si progetta infine un case prototipale per contenere il circuito. Si decide di realizzarlo in due componenti ad incastro, senza prevedere l'uso di viti o sistemi di serraggio. Ciò torna utile in fase prototipale perché permette un facile accesso all'interno, necessario finché si tratta di un prototipo. In un prodotto definitivo è comunque consono utilizzare sistemi come viti per mantenere perfettamente chiuso il dispositivo.

Le pareti del case hanno uno spessore di 0.5cm e prevedono diverse finestre utili. Il sistema vano/coperchio è stato pensato ad incastro. Alcuni fori sono stati integrati in prossimità del sensore di gas e il sensore di temperatura/umidità, per permettere un contatto diretto con l'aria dell'ambiente. Sono state realizzate due finestre per il display, il sensore PIR e il led di notifica. È stata lasciata una feritoia per l'USB del microcontrollore e una per una eventuale antenna esterna. Infine, è stato inserito un adattatore ad incastro per il pulsante.

Alcuni esempi di render sono visibili nelle seguenti figure:



*Figura 29 - Rendering del case con circuito*

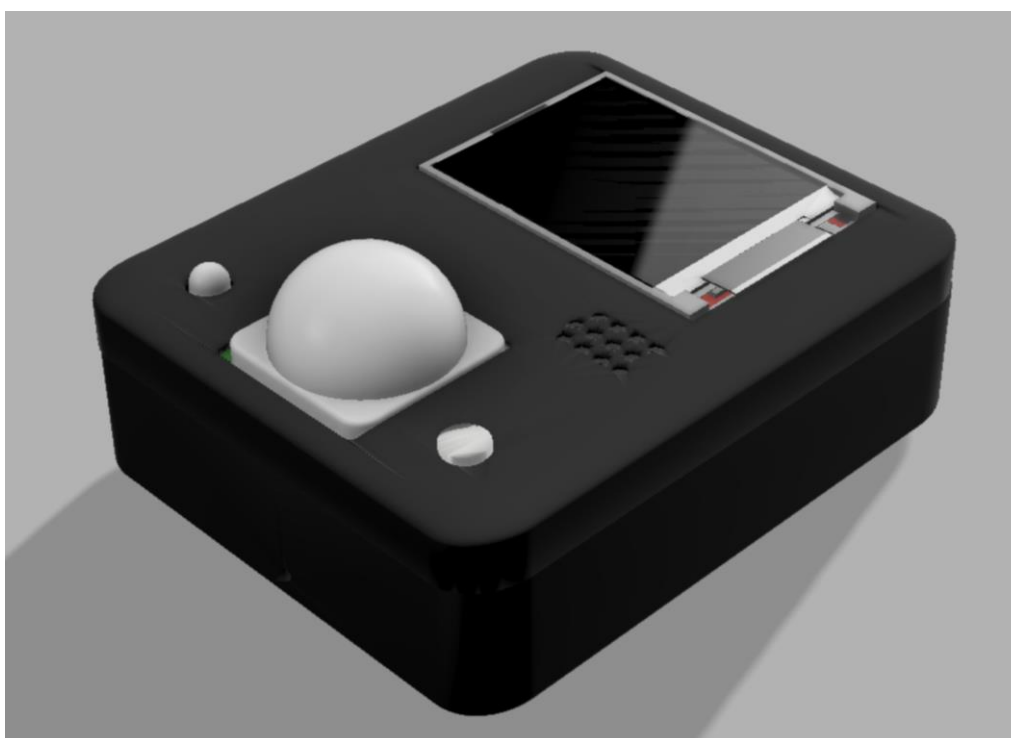


*Figura 30 - Parte superiore del case*

Infine, il dispositivo completo in due varianti, nero e bianco:



*Figura 31 - Rendering del dispositivo finito, versione bianca*



*Figura 32 - Rendering del dispositivo finito, versione nera*

#### 4.12. Stima costi

Per la progettazione e realizzazione del sistema si è tenuto conto, in fase di selezione dei componenti, dei relativi costi in rapporto soprattutto a prodotti già diffusi in ambito commerciale.

Di seguito, si riporta in tabella l'elenco dei componenti e costi:

Componenti	Costo [€]
Arduino Nano	7.38
DHT22	4.23
PIR HC-SR501	2.4
MQ-5	5.84
Display 1.8" TFT	16.9
Led RGB	0.84
Buzzer	2.07
PCB + montaggio	10
Case	10
<b>Totale</b>	<b>59.66</b>

*Figura 33 - Tabella costi*

Alla cifra totale, vanno comunque considerate le spese per gli elementi passivi come condensatori e resistori (di valore comunque nettamente trascurabile, dell'ordine dei centesimi di euro).

C'è da considerare inoltre la spesa per il PCB e montaggio componenti, stimabile intorno alla decina di euro.

Infine, la realizzazione del case si può ipotizzare avvenga tramite stampa 3D per una prima fase prototipale, con un costo aggiuntivo della decina di euro.

Il costo finale si aggira intorno ai 60 euro che, considerata la fase prototipale che ha costi maggiori rispetto alla produzione industriale, si rende competitiva rispetto ai costi dei prodotti commerciali.

## 5. Riferimenti

- 1) Abraham, S., & Li, X. (2014). A Cost-effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications. *Procedia Computer Science*, 34, 165–171. <https://doi.org/10.1016/j.procs.2014.07.090>
- 2) Castell, N., Dauge, F. R., Schneider, P., Vogt, M., Lerner, U., Fishbain, B., Broday, D., & Bartonova, A. (2017). Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? *Environment International*, 99, 293–302. <https://doi.org/10.1016/j.envint.2016.12.007>
- 3) New Digital Metal-Oxide (MOx) Sensor Platform. (2018). *Sensors*, 18(4), 1052. <https://doi.org/10.3390/s18041052>
- 4) Postolache, O. A., Pereira, J. M. D., & Girao, P. M. B. S. (2009). Smart Sensors Network for Air Quality Monitoring Applications. *IEEE Transactions on Instrumentation and Measurement*, 58(9), 3253–3262. <https://doi.org/10.1109/tim.2009.2022372>
- 5) ST7735 datasheet: <https://www.displayfuture.com/Display/datasheet/controller/ST7735.pdf>
- 6) LCM datasheet: <https://cdn-shop.adafruit.com/datasheets/TC1602A-01T.pdf>
- 7) Buzzer CMI-9650C-030 datasheet: <https://www.cuidevices.com/product/resource/cmi-9650c-030.pdf>
- 8) Led RGB Inolux Datasheet: <http://www.inolux-corp.com/datasheet/Inolux%20Lamp/TH%20Lamp/HV-5RGBXX%205mm%20Full-Color%20Series%20v.1.2.pdf>
- 9) Atmega328 datasheet: [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)
- 10) Golda, A., & Kos, A. (2006). Analysis And Design Of Pstat Temperature Sensor In Digital CMOS VLSI Circuits. *Proceedings of the International Conference Mixed Design of Integrated Circuits and System*, 2006. MIXDES 2006., 415–420. <https://doi.org/10.1109/mixdes.2006.1706612>
- 11) DS18B20 datasheet: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- 12) Prockop, L. D., & Chichkova, R. I. (2007). Carbon monoxide intoxication: An updated review. *Journal of the Neurological Sciences*, 262(1–2), 122–130. <https://doi.org/10.1016/j.jns.2007.06.037>
- 13) MacLean, R. R., Valentine, G. W., Jatlow, P. I., & Sofuoglu, M. (2017). Inhalation of Alcohol Vapor: Measurement and Implications. *Alcoholism: Clinical and Experimental Research*, 41(2), 238–250. <https://doi.org/10.1111/acer.13291>



- 14) Honeycutt, W. T., Ley, M. T., & Materer, N. F. (2019). Precision and Limits of Detection for Selected Commercially Available, Low-Cost Carbon Dioxide and Methane Gas Sensors. *Sensors*, 19(14), 3157. <https://doi.org/10.3390/s19143157>
- 15) Sriwati, Ikhsan Ilahi, N., Musrawati, Baco, S., Suyuti' Andani Achmad, A., & Umrinah, E. (2018). Early Leakage Protection System of LPG (Liquefied Petroleum Gas) Based on ATmega 16 Microcontroller. *IOP Conference Series: Materials Science and Engineering*, 336, 012021. <https://doi.org/10.1088/1757-899x/336/1/012021>
- 16) LM34/LM35 Precision Monolithic Temperature Sensors. National Semiconductor Application Note 460, October 1986
- 17) IPC-2221B, Generic Standard on Printed Board Design