

Rivellino Smart Sensors

Introduzione

Il sistema **Rivellino Smart Sensors** è progettato per monitorare le condizioni ambientali all'interno del **Rivellino degli Invalidi**, parte del **Museo Pietro Micca** di Torino.

I sensori installati nei tre moduli principali **Tirante (A)**, **Galleria (B)** e **Polveriera (C)** raccolgono dati su:

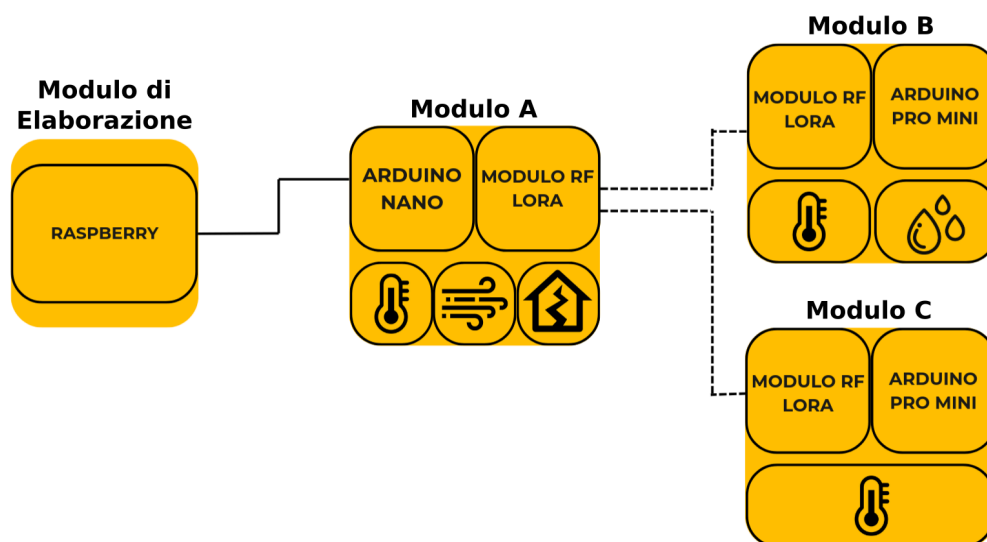
- Temperatura e Umidità
- Qualità dell'aria (CO e NO₂)
- Vibrazioni
- Allagamento

Questi dati vengono trasmessi via seriale ad un Raspberry Pi e letti da un ricevitore scritto in Python, che li elabora e li salva automaticamente in file CSV.

Il sistema garantisce un monitoraggio continuo e affidabile degli ambienti ipogei, contribuendo alla **tutela e alla conservazione del patrimonio storico**.

I moduli

- [Modulo Ricevitore](#) per l'elaborazione dei dati
- [Modulo A](#) → tirante (Temperatura e Umidità, Aria, Vibrazioni)
- [Modulo B](#) → galleria (Batterie, Temperatura e Umidità, Aria, Allagamento)
- [Modulo C](#) → polveriera (Batterie, Temperatura e Umidità)

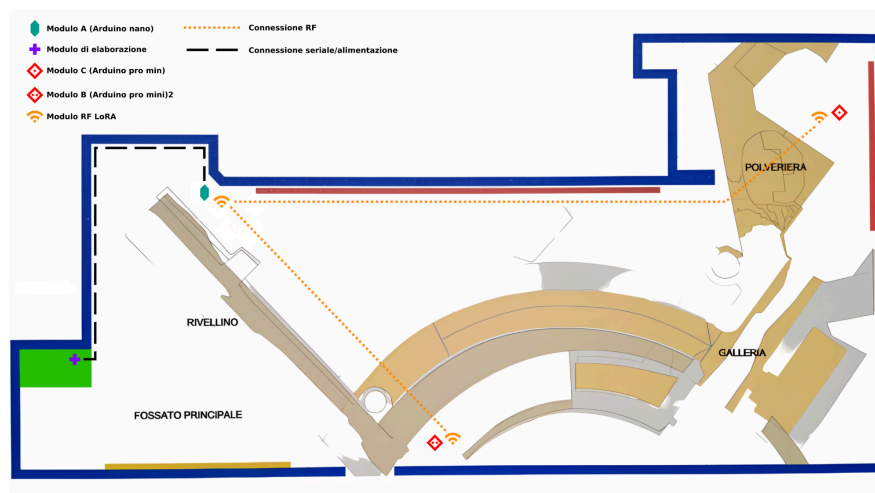


I dati

ID	Descrizione	Valore	Modulo
a	Batteria	%	B
b	Batteria	%	C
c	Temperatura	°C	A
d	Temperatura	°C	B
e	Temperatura	°C	C
f	Umidità	%	A
g	Umidità	%	B
h	Umidità	%	C
i	Aria (CO)	ppm	A
j	Aria (NO2)	ppm	B
k	Vibrazioni	Hz	A
l	Vibrazioni	m/s ²	A
m	Allagamento	0 / 1	B

I dati verranno elaborati dal ricevitore e salvati all'interno dei files CSV.

Posizionamento



Il ricevitore

Assicurarsi di avere l'ultima versione di Python3 installata sul dispositivo (in questo caso sul Raspberry Pi 5).

Eseguire `pip install pyserial` nel terminale per installare la libreria pyserial.

Eseguire `python3 main.py` nel terminale, all'interno della cartella in cui si trova il programma ricevitore.

Il programma proverà a connettersi alla porta seriale. Quando la connessione sarà stabilita verrà mostrata la porta utilizzata e la velocità di comunicazione impostata.

Il programma riceverà dati periodicamente. Esempio di dati inviati da parte dei moduli:

- moduloA: `c28f58j0.06k15.5l1.9i0.07`
- moduloB: `a61d27g56m1`
- moduloC: `b62e28h42`

I dati verranno salvati nella cartella "DatiSensori" nei seguenti files:

File CSV	ID del dato
allagamento.csv	m
ariaCO.csv	i
ariaNO2.csv	j
batterie.csv	a, b
vibrazioni.csv	k, l
temperatura_umidita.csv	c, d, e, f, g, h

Il file di configurazione `config.py` permette di modificare alcuni parametri relativi al programma di ricezione dei dati, tra cui:

- Selezionare la porta di comunicazione
- Selezionare la velocità di comunicazione (baud rate)
- Selezionare il percorso dei files CSV
- Rinominare i files CSV e di inserirne di nuovi

- Mappare nuovi sensori
- Modificare le colonne dei files CSV
- Scegliere se mostrare i dati ricevuti sul terminale
- Personalizzare i messaggi di log

Le informazioni di comunicazione ed eventuali errori saranno salvati all'interno di:
[logfile.log](#)

Librerie Arduino

Estrarre [libraries.zip](#) all'interno della cartella

[C:\Users\%USERPROFILE%\Documents\Arduino\](#), riavviare eventualmente Arduino IDE.

Configurazione LoRa

- Lora Modulo A
 - [AT+ADDRESS=1](#)
 - [AT+NETWORKID=5](#)
 - [AT+IPR=9600](#)
- Lora Modulo B
 - [AT+ADDRESS=2](#)
 - [AT+NETWORKID=5](#)
 - [AT+IPR=9600](#)
- Lora Modulo C
 - [AT+ADDRESS=3](#)
 - [AT+NETWORKID=5](#)
 - [AT+IPR=9600](#)

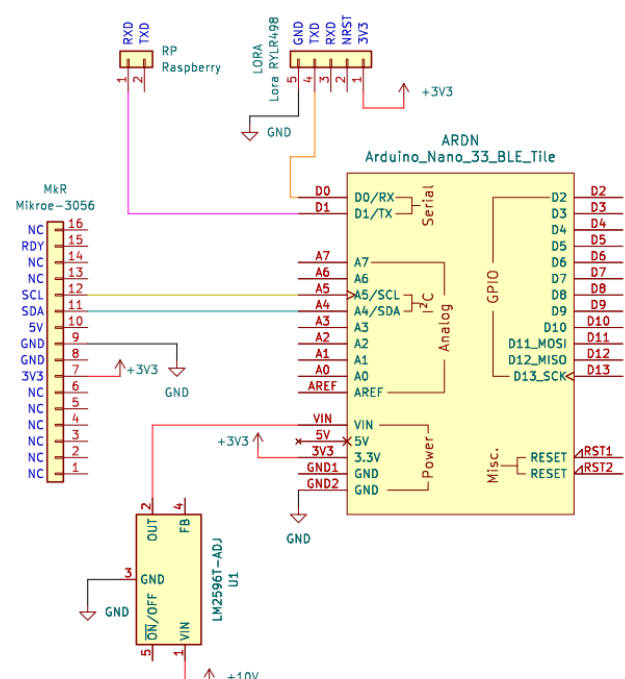
Modulo A

Componenti:

- Arduino Nano 33 BLE Sense Rev2
- Sensore Qualità Aria Mikroe3056
- Modulo LoRa RYLR498

Esempio di dati inviati:

[c28f58j0.06k15.511.9i0.07](#)

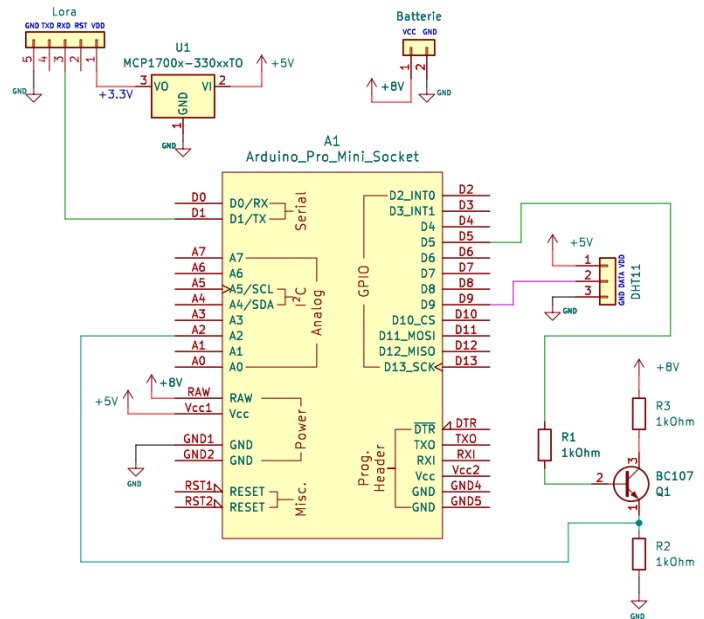


Modulo C

Componenti:

- Arduino Pro Mini
- Sensore Temp. e Umidità DHT11
- Modulo LoRa RYLR498
- Batterie AA Alcaline x5 (serie)

Esempio di dati inviati: **b62e28h42**



ID	Descrizione	Valore
b	Batteria	%
e	Temperatura	°C
h	Umidità	%

ITS AEROSPAZIO PIEMONTE

CORSO EMBT02

Rivellino Smart Sensors

<https://github.com/Simv135/Rivellino-Smart-Sensors>