



Università Politecnica Delle Marche

FACOLTÀ DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA

**Studio e sviluppo di opto-sensori per l'analisi
del PM2.5 e PM10
Study and development of
opto-sensors for the analysis of PM2.5 and PM10**

Candidato:

Gigli Michele

Matricola S1093162

Relatore:

Ippoliti Gianluca

Correlatore:

Di Buò Gianluca

Un ringraziamento speciale va alla mia famiglia, in particolare a mio padre Simone e mia nonna Flora, per il loro sostegno ed incoraggiamento: è grazie a loro e a tutti gli altri familiari se oggi raggiungo questo importante traguardo.

Desidero poi ringraziare l'egregio Prof. Gianluca Ippoliti e l'azienda IDEA. Soc. Coop. per la possibilità offertami con questo lavoro di tesi, tramite il quale ho potuto mettere a frutto molte delle conoscenze acquisite in questi anni di studio all'università Politecnica delle Marche.

Venendo ai ragazzi del mio corso di studio vorrei ringraziare: Alessio Martinelli e Damiano Santarelli per la vicinanza dimostratami, per l'aiuto e gli ottimi suggerimenti nei momenti più opportuni.

Un grazie di cuore poi al Prof. Francesco Piva, senza il quale l'attività di tirocinio sarebbe stata sicuramente molto più difficile e meno piacevole.

Ringraziare poi Andrea Rubechini e Nazzareno Villa sarebbe troppo poco, loro durante il mio periodo di lavoro all'interno di IDEA mi hanno fatto capire che anche lavorando ci si può divertire e si possono creare amicizie stupende.

Un ringraziamento speciale va fatto ai miei nonni: Flora, Luciana, Giuseppe e Giuliano, che sono sempre stati i miei primi sostenitori in tutto e per tutto.

Non posso che concludere l'elenco con un ringraziamento particolare a Virginia Luzi, la sorpresa più bella di quest'ultimo periodo universitario, e a tutti i miei amici: Federico E., Federico R., Matteo, Emanuel, Ettore, Alberto, Giacomo, Pietro, Francesco B., Elisa, Chiara, Martina, Riccardo, Lucia, Lorenzo, Niccolò, Francesco M., Andrea, per la loro vicinanza, amicizia e lealtà nei momenti migliori e peggiori della mia vita.

Sommario

In primo luogo, lo scopo di questa tesi è di implementare un sensore commerciale per la misura della densità del particolato ambientale $PM_{2.5}$ e PM_{10} all'interno di DoNet, progettato dall'azienda IDEA Soc. Coop. e già dotato di sensori per la misura di temperatura, umidità, luminosità, CO_2 e TVOC dell'aria.

In un secondo momento, invece, verrà studiato come realizzare da zero un sensore per la misura della densità di particolato ambientale, partendo dalla sua progettazione e arrivando allo sviluppo di un prototipo funzionante.

DoNet rappresenta un nodo di una rete di sensori, che fornisce i dati raccolti ad un gateway, per mezzo del quale vengono salvati in un database e mostrati all'utente attraverso un'applicazione web, raggiungibile da un qualsiasi Internet browser.

La tesi è suddivisa nei capitoli mostrati di seguito:

1. **Introduzione al particolato:** viene spiegato meglio che cos'è il particolato ambientale e che relazione ha con la diffusione dei virus e delle malattie.
2. **Implementazione di un sensore per la misura della densità di particolato ambientale all'interno di DoNet:** una volta scelto il sensore è stata analizzata la sua implementazione in DoNet sia dal punto di vista hardware sia dal punto di vista software;
3. **Implementazione embedded di un sensore per la misura della densità di particolato ambientale $PM_{2.5}$ e PM_{10} :** progettazione e realizzazione teorica e pratica di un sensore per la misura della densità di $PM_{2.5}$ e PM_{10} ;
4. **Sviluppo prototipi:** definizione di due prototipi di tale sensore;
5. **Risultati finali:** analisi dei risultati ottenuti.

Indice

Abstract	4
1 Introduzione	14
1.1 Idea società cooperativa	14
1.2 Introduzione al particolato ambientale (PM _{2.5} e PM ₁₀)	16
1.2.1 Particolato e salute	16
1.2.2 Particolato e Virus	18
2 Implementazione di un sensore per la misura della densità di particolato ambientale all'interno di DoNet	22
2.1 Che cos'è DoNet?	22
2.1.1 Chip	22
2.1.2 Sensori	23
2.1.3 Broker MQTT	24
2.1.4 Programmazione dei dispositivi IoT	25
2.1.4.1 MicroPython	25
2.1.4.2 MicroPython per ESP32	25
2.1.4.3 Gestione della memoria e programmazione della scheda	26
2.1.5 Ponte per la connessione da USB a TTL	27
2.2 Sensori per la misura del particolato ambientale	28
2.2.1 Stato dell'arte	28
2.2.1.1 LED infrarossi	28
2.2.1.2 Scattering laser	29
2.2.1.3 Considerazioni	29
2.3 Valutazione di alcuni dei sensori in commercio	31
2.3.1 SM-UART-04L	32
2.3.2 HUMA-I 150	35
2.3.3 Tera NEXT-PM	37
2.3.4 Omron B5W-LD0101	42
2.4 Confronto fra sensori e scelta sensore da implementare in DoNet	44
2.4.1 Sistema di raccolta dati	44
2.4.2 Raccolta dati	47
2.4.3 Scelta del sensore	48
2.5 Implementazione SM-UART-04L su DoNet	50
2.5.1 Implementazione Hardware	50
2.5.2 Implementazione Software	51
2.5.2.1 Listati	52
2.6 Risultati Ottenuti	54
3 Implementazione embedded di un sensore per la misura della densità di particolato ambientale PM_{2.5} e PM₁₀	55
3.1 Tecnologie prese in considerazione per l'implementazione del sensore	56
3.1.1 Citofluorimetria (citometria a flusso)	56
3.1.2 Nanosight	59
3.1.3 Dynamic light scattering	60
3.1.3.1 Step implementativi	60
3.1.3.2 Perché non utilizzare la DLS per la misura della densità del particolato ambientale	65

3.1.4	Scattering di Mie	66
3.1.5	Tecnologia scelta per l'implementazione del sensore: scattering laser basato sulla teoria di Mie	70
3.2	Strumenti utilizzati per l'implementazione del sensore	71
3.2.1	Microcontrollore: Esp32	71
3.2.2	Laser: SYD1230 (Rosso)	71
3.2.3	Fotorilevatore: SFH 2401	73
3.2.4	Convertitore analogico-digitale: DollaTek ADS1115	75
3.2.5	Ventola: AERZETIX	77
3.3	Firmware implementato	78
3.3.1	Listati	79
4	Prototipi	83
4.1	Prototipo 1	84
4.2	Prototipo 2 - finale	86
4.3	Relazione fra intensità di luce scatterata e dimensione di particolato PM _{2.5} e PM ₁₀	88
4.4	Relazione fra intensità di luce scatterata di PM _{2.5} e PM ₁₀ e densità di particolato PM _{2.5} e PM ₁₀	90
5	Risultati finali	92
6	Conclusioni	97
6.1	Sviluppi futuri	97
	Bibliografia	99

Elenco delle figure

1.1	Logo IDEA Soc. Coop.	14
1.2	Dimensioni	16
1.3	Decessi a causa del particolato ambientale, 2017	17
1.4	Tabella superamento limiti	17
1.5	Correlazione influenza aviaria e particolato	18
1.6	Correlazione virus sinciziale umano e particolato	18
1.7	Espansione sars-cov-2 Italia	19
1.8	Curve di espansione sars-cov-2 Italia	19
1.9	Covid e recettore ACE-2	20
1.10	Titolazione virale sars-cov-2	20
2.1	DoNet	22
2.2	Esp32	22
2.3	Computer Raspberry Pi 3 Model B	24
2.4	Chipset FT232RL	27
2.5	connessione da USB a TTL	27
2.6	Tecnologia led	28
2.7	Tecnologia laser	29
2.8	Schema implementativo tecnologia led	29
2.9	Schema implementativo tecnologia laser	30
2.10	SM-UART-04L	32
2.11	SM-UART-04L datas	32
2.12	SM-UART-04L caratteristiche	33
2.13	SM-UART-04L caratteristiche	33
2.14	SM-UART-04L caratteristiche	33
2.15	SM-UART-04L possibili errori	34
2.16	Huma-i 150	35
2.17	Ciclo di funzionamento Huma-i	35
2.18	Informazioni sullo stato della qualità dell'aria	36
2.19	Valori di stato variabili Huma-i 150	36
2.20	Specifiche tecniche Huma-i 150	36
2.21	Tera NEXT-PM	37
2.22	Tera NEXT-PM specifiche tecniche	37
2.23	Tera NEXT-PM specifiche elettriche	38
2.24	Protocollo UART	38
2.25	Protocollo ModBus	39
2.26	Tera NEXT-PM comandi	40
2.27	Tera NEXT-PM conversione	40
2.28	Tera NEXT-PM stato	41
2.29	Omron B5W-LD0101	42
2.30	Omron B5W-LD0101 incense	42
2.31	Omron B5W-LD0101 PM _{2,5} conversion	43
2.32	Omron B5W-LD0101 PM ₁₀ conversion	43
2.33	SM-UART-04L alimentazione	44
2.34	Convertitore logico bidirezionale	45
2.35	Schema dei collegamenti	45
2.36	Device realizzato	46
2.37	Misurazioni effettuate con incenso introdotto all'interno dell'ambiente e subito rimosso, senza però utilizzare la ventilazione artificiale	47

2.38	Misurazioni effettuate con incenso introdotto all'interno dell'ambiente e subito rimosso, utilizzando però in questo caso la ventilazione artificiale	47
2.39	Misurazioni effettuate con incenso lasciato fuoriuscire dall'ambiente prima del rilevamento	47
2.40	Tabella caratteristiche sensori	48
2.41	Implementazione hardware SM-UART-04L su DoNet	50
2.42	Implementazione software SM-UART-04L su DoNet	51
2.43	Ciclo di funzionamento DoNet con SM-UART-04L	51
2.44	Risultati ottenuti tramite database Netsons	54
2.45	Risultati ottenuti PM _{2.5} e PM ₁₀ con nuova struttura DoNet	54
3.1	Possibile implementazione sensore PM _{2.5} e PM ₁₀	55
3.2	Possibile implementazione citometria a flusso	57
3.3	Impulso elettrico	58
3.4	Quantizzazione impulso elettrico	58
3.5	Tecnica nanosight	59
3.6	Dynamic light scattering	60
3.7	Scattering	61
3.8	Relazione fra intensità di luce scatterata, dimensione della particella e angolo di scattering.	67
3.9	Grafico scattering in avanti di una singola particella di particolato	68
3.10	Possibile implementazione principio scattering Mie	69
3.11	Primo esempio di sensore per il particolato ambientale che utilizza come tecnica di misurazione della densità di particolato lo scattering laser	70
3.12	Secondo esempio di sensore per il particolato ambientale che utilizza come tecnica di misurazione della densità di particolato lo scattering laser	70
3.13	Esp32	71
3.14	SYD1230	71
3.15	Specifiche laser SYD1230	72
3.16	SFH 2401	73
3.17	Sensibilità spettrale relativa SFH 2401	73
3.18	Maximum ratings SFH 2401	74
3.19	Caratteristiche SFH 2401	74
3.20	ADS1115	75
3.21	Maximum ratings ADS1115	75
3.22	Caratteristiche ADS1115	76
3.23	Ventola AERZETIX 20x20x8	77
3.24	Firmware sensore particolato ambientale	78
3.25	File Risultati.txt	78
4.1	Prototipo 1 sensore particolato ambientale	84
4.2	Secondo ed ultimo prototipo sensore particolato ambientale	86
4.3	Sistema finale realizzato per l'implementazione del sensore per la misurazione della densità di particolato ambientale	87
4.4	Gaussiana rilevamento particolato	88
5.1	Confronto fra risultati ottenuti sensore IDEA e SM-UART-04L in condizioni normali all'interno dell'ufficio industria 4.0 di IDEA con ricircolo aria	92
5.2	Confronto fra risultati ottenuti sensore IDEA e SM-UART-04L in condizioni normali all'interno del deposito al primo piano di IDEA senza ricircolo aria	93
5.3	Confronto fra risultati ottenuti sensore IDEA e SM-UART-04L in condizioni di medio inquinamento dell'aria simulato con incenso all'interno del deposito al primo piano di IDEA senza ricircolo aria	94

5.4	Confronto fra risultati ottenuti sensore IDEA e SM-UART-04L in condizioni di estremo inquinamento dell'aria simulato con incenso all'interno del deposito al primo piano di IDEA senza ricircolo aria	95
-----	---	----

Elenco delle tabelle

1 Confronto ESP32-ESP8266 23

Elenco dei listati

1	smuart.py	52
2	prima parte introdotta da linea 83 a linea 84 donet.py	53
3	modifica valori da inviare a database in donet.py	53
4	terza modifica effettuata da linea 345 a linea 349 in donet.py	53
5	main.py txt file	79
6	Interrupt service routine for data acquisition called by a timer interrupt	79
7	Main part of the code main.py	80
8	class ADS1115	80
9	Funzione di correlazione dimensione particolato-intensità luce scatterata	88
10	Funzione di correlazione densità particolato-intensità luce scatterata	90

1 Introduzione

1.1 Idea società cooperativa

Prima di procedere con l'introduzione al lavoro svolto per la tesi di laurea, è mio interesse personale fare i più sentiti ringraziamenti all'azienda che ha reso possibile questo progetto e ad ognuno dei suoi collaboratori per la loro disponibilità e competenza.

IDEA Soc. Coop. è una società di ingegneria la cui sede si trova in Ancona che fornisce soluzioni personalizzate ai clienti negli ambiti dell'automazione industriale, elettronica, robotica, domotica ed informatica oltre che essere molto attiva nel campo di ricerca e sviluppo.

IDEA nasce ad Ancona nel 2006 come spin off universitario al seguito della vincita del concorso eCapital.



Figura 1.1: Logo IDEA Soc. Coop.

Successivamente viene costituita la società sotto forma di cooperativa con sede legale nel dipartimento DIIGA (Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione) dell'Università Politecnica delle Marche. Questa vicinanza all'università permette una rapida crescita e la nascita di numerose collaborazioni con alcune delle più note aziende del territorio, collaborazioni grazie alle quali si sono sviluppate numerose competenze negli ambiti sopra citati. Attualmente l'azienda è composta di 16 collaboratori, tutti altamente qualificati.

Tra le attività svolte dall'azienda troviamo:

- **Sviluppo hardware:** a partire dallo studio di fattibilità fino al design dei circuiti stampati senza escludere fasi di test e collaudo delle soluzioni proposte;
- **Sviluppo firmware:** l'azienda è in grado di fornire, tramite l'utilizzo delle più moderne piattaforme di sviluppo, soluzioni per ogni tipo di architettura, a partire dalle più semplici fino alle più moderne e complesse architetture a multiprocessore. È inoltre prevista un'ampia fase di testing atto a garantire un'elevata affidabilità e robustezza del prodotto finale;
- **Sviluppo software:** progettazione, sviluppo e manutenzione di sistemi software su specifica del cliente dalla fase preliminare di analisi dei requisiti alla definizione delle specifiche funzionali fino alla realizzazione;

- **Automazione industriale:** sviluppo software tramite piattaforma Labview National Instrument oltre che il riadattamento ed ottimizzazione di macchine esistenti tramite ri-progettazione dei circuiti elettrici, riorganizzazione del quadro elettrico ed implementazioni di logiche di controllo tramite PLC;
- **Stampa 3D:** dopo la richiesta di un preventivo è possibile proseguire alla stampa tramite diversi tipi di materiale. L'azienda fornisce anche consulenza riguardo il tipo di materiale più adatto al progetto;
- **Ricerca e sviluppo:** avendo una particolare esperienza pregressa nel campo, l'azienda è in grado di fornire assistenza ai clienti per lo sviluppo di piattaforme innovative, dalla definizione delle specifiche di progetto, all'ottenimento dei finanziamenti fino alla realizzazione del progetto.

Risulta essere di particolare interesse la metodologia di gestione dei progetti adottata dall'azienda. Infatti, grazie al supporto di avanzati strumenti informatici (Easy Redmine), la risorsa che dovrà gestire il progetto sarà in grado di monitorare l'avanzamento dello stato dei lavori ed il completamento dei task precedentemente assegnati ai collaboratori. In più la piattaforma permette di mantenere una documentazione aggiornata dell'intero progetto ed il monitoraggio delle attività viene consentito anche ai clienti garantendo la più totale trasparenza.

1.2 Introduzione al particolato ambientale (PM_{2.5} e PM₁₀)

Per PM, acronimo di particulate matter, si intendono tutte quelle particelle di piccole dimensioni che sono in sospensione nell'atmosfera. Una prima distinzione viene effettuata sulla base delle loro dimensioni. Vengono dette PM_{2.5} le particelle di diametro inferiore ai 2.5µm mentre le PM₁₀ sono tutte quelle di diametro inferiore ai 10µm.

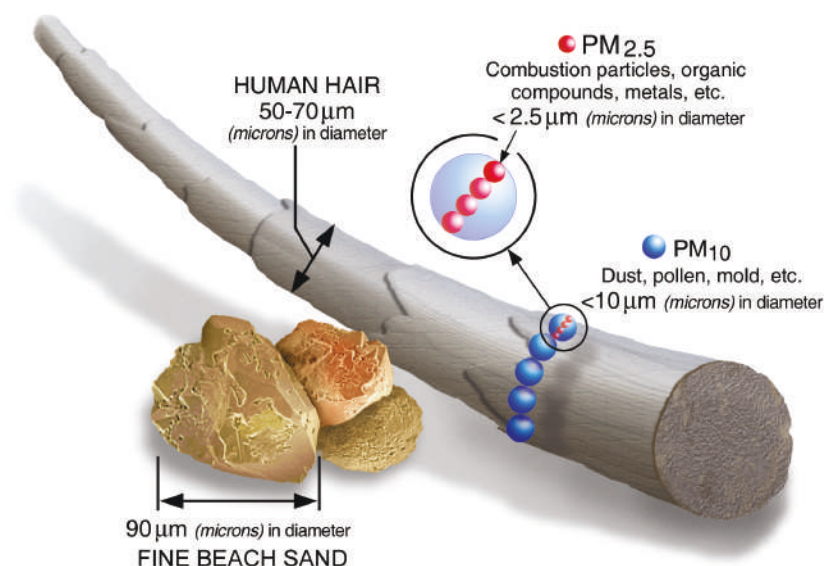


Figura 1.2: Dimensioni

Per fare un paragone basti pensare che un capello umano risulta avere un diametro compreso tra i 50µm ed i 70µm mentre un granello di sabbia ha dimensioni di circa 90µm.

Sono composte sia da particelle solide che liquide come polvere, fumo, aerosol (microgocce di sostanze liquide), e possono rimanere in circolazione nell'atmosfera per lunghi periodi.

Esiste poi un'ulteriore classificazione che si basa sulla provenienza delle PM:

- **Primarie:** possono avere origine antropica (dovute all'attività umana) e possono essere associate ai processi di combustione legati alla necessità di produrre energia elettrica, all'industria, ai trasporti, alla combustione di biomasse di origine legnosa, dall'erosione degli pneumatici e dell'asfalto. Possono poi essere di origine naturale legate all'eruzione dei vulcani ed alla dispersione dei pollini;
- **Secondario:** vengono generate da combinazioni di reazioni chimiche che hanno origine in atmosfera.

1.2.1 Particolato e salute

Il particolato aerodisperso è in grado di assorbire gas e vapori tossici. In questo modo aumentano le concentrazioni dei gas inquinanti che raggiungono le zone più profonde del polmone, trasportati dalle particelle di PM₁₀ e PM_{2.5}.

Numerosi studi hanno evidenziato una correlazione tra esposizione acuta a particolato aerodisperso e sintomi respiratori, alterazioni della funzionalità respiratoria, ricoveri in ospedale e mortalità per malattie respiratorie.

Si stima che attualmente nel mondo muoiono più di 2mln di persone a causa dell'alta percentuale di particolato atmosferico, di cui circa 400 000 in Europa:

Number of deaths from air pollution, 2017

Number of deaths from air pollution is the sum of attributed deaths to both indoor and outdoor air pollution.

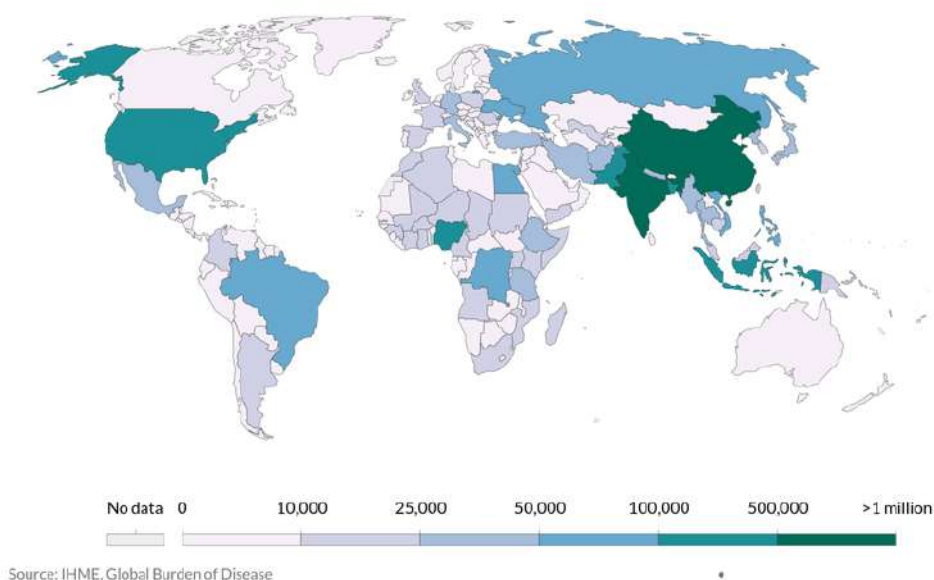


Figura 1.3: Decessi a causa del particolato ambientale, 2017

Come conseguenza delle considerazioni effettuate sull'aumento del tasso di mortalità mondiale l'AIRC (Associazione Internazionale per la Ricerca sul Cancro) ha inserito le micropolveri nel gruppo uno, ovvero nella tabella contenente tutti gli agenti cancerogeni per l'uomo.

Per questo sono state prese delle precauzioni a livello legislativo imponendo un limite alle emissioni giornaliere di particolato in maniera da mantenere un massimo di $50\mu\text{g}/\text{m}^3$ da non superare per più di 35 giorni l'anno.

Dalla prossima tabella possiamo notare che nel 2020 molte province Italiane hanno largamente superato questo limite:

REGIONE	n.stazioni >35 superamenti (limite Ue e Italia)	n.stazioni <=35 e > 3 superamenti	n. stazioni <= 3 superamenti (valore rif. OMS)	Totale
ABRUZZO	0	6	5	11
BASILICATA	0	4	6	10
CALABRIA	0	15	5	20
CAMPANIA	10	22	2	34
EMILIA- ROMAGNA	24	16	3	43
FRIULI VENEZIA GIULIA	5	16	2	23
LAZIO	6	25	19	50
LIGURIA	0	8	21	29
LOMBARDIA	55	8	2	65
MARCHE	0	13	5	18
MOLISE	1	0	0	1
PA BOLZANO	0	0	5	5
PA TRENTO	0	4	2	6
PIEMONTE	20	10	3	33
PUGLIA	0	40	12	52
SARDEGNA	0	8	16	24
SICILIA	1	13	1	15
TOSCANA	1	21	12	34
UMBRIA	5	10	6	21
VALLE D'AOSTA	0	5	0	5
VENETO	27	6	2	35
ITALIA	155	250	129	534

Figura 1.4: Tabella superamento limiti

1.2.2 Particolato e Virus

In generale, è noto come il particolato atmosferico possa fungere da carrier per molti virus ed agenti chimici in generale, e come esso possa costituire un substrato che favorisca la sopravvivenza dei virus in aria. A tal proposito, nel caso di precedenti contagi virali, alcune ricerche hanno evidenziato una correlazione tra la proliferazione dei virus al variare delle concentrazioni di particolato ambientale:

- **2010:** si è osservata una correlazione di tipo esponenziale tra le infezioni di influenza aviaria e le concentrazioni di PM_{10} e $PM_{2.5}$. Si è visto poi come l'influenza poteva effettivamente essere trasportata per lunghe distanze dalle tempeste di polveri asiatiche.

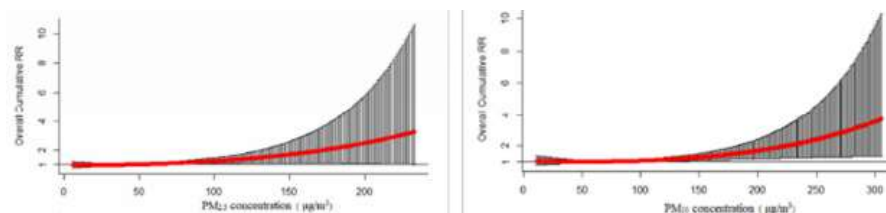


Figura 1.5: Correlazione influenza aviaria e particolato

- **2016:** si è evidenziata una relazione tra la diffusione del virus respiratorio sinciziale (VRS) umano e le concentrazioni di particolato

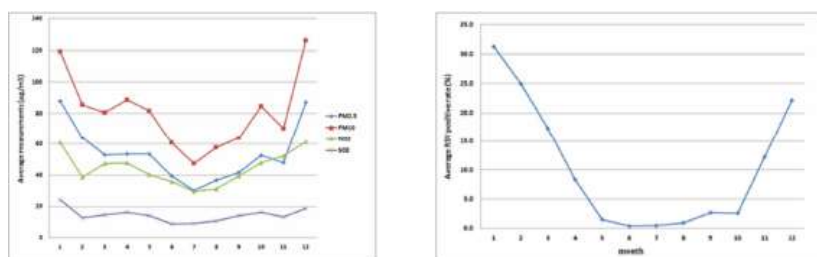


Figura 1.6: Correlazione virus sinciziale umano e particolato

- **2017:** il numero di casi di morbillo su 21 città cinesi nel periodo 2013 – 2014 varia in relazione alle concentrazioni di $PM_{2.5}$. I ricercatori dimostrano che un aumento delle concentrazioni di $PM_{2.5}$ pari a $10\mu g/m^3$ incide significativamente sull'incremento del numero di casi di virus del morbillo. I ricercatori suggeriscono di ridurre le concentrazioni di $PM_{2.5}$ per diminuire la diffusione dell'infezione.
- **2020:** uno dei maggiori fattori di diffusione giornaliera del virus del morbillo in Lanzhou (Cina) sono i livelli di inquinamento di particolato atmosferico. In relazione all'evidenza che l'incidenza del morbillo sia associata all'esposizione a $PM_{2.5}$ ambientale in Cina, i ricercatori suggeriscono che politiche efficaci di riduzione dell'inquinamento atmosferico possono ridurre l'incidenza del morbillo.

Sulla base di questa rassegna scientifica, storicamente ricostruita, si può quindi dedurre che il particolato atmosferico (PM_{10} e $PM_{2.5}$) costituisce un efficace vettore per il trasporto, la diffusione e la proliferazione delle infezioni virali.

Sulla stessa linea delle considerazioni sopra citate, un'analisi dei focolai in Italia del più recente *sars-cov-2*, mostra come quest'ultimo abbia proliferato maggiormente nelle aree con più alta media di superamenti delle concentrazioni consentite di particolato 1.7.

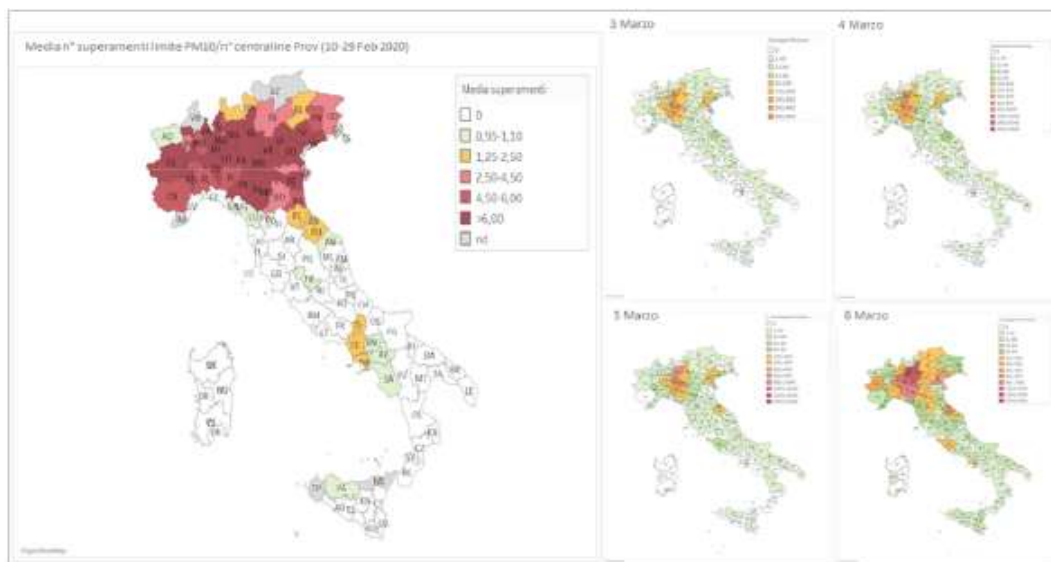


Figura 1.7: Espansione sars-cov-2 Italia

Inoltre, dal grafico in figura 1.8 si può notare come le curve di espansione dell'infezione nelle regioni del sud Italia presentano andamenti esponenziali perfettamente compatibili con i modelli epidemiologici, mentre nelle regioni come la Lombardia sono evidenti delle accelerazioni anomale dette BOOST in concomitanza (tenendo a mente il periodo di incubazione del virus) alla presenza di elevate concentrazioni di particolato:

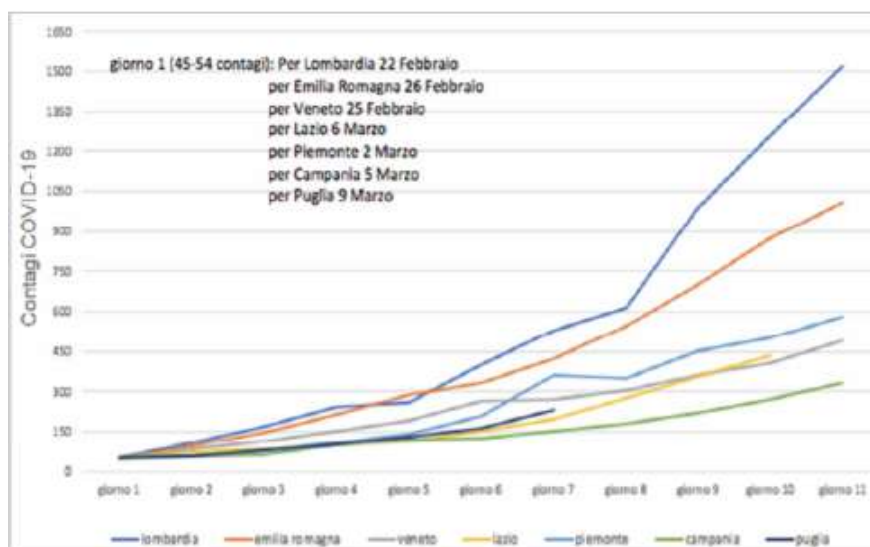


Figura 1.8: Curve di espansione sars-cov-2 Italia

Si certa di capire se effettivamente il particolato ambientale possa essere causa di una nuova modalità di contagio. In particolare dei ricercatori dell'ospedale San Raffaele di Milano hanno dimostrato su cavie come all'esposizione cronica al particolato ambientale $PM_{2.5}$ e PM_{10} si associa un'iper-espressione delle cellule polmonari del recettore ACE – 2 noto per essere la "chiave d'accesso" del nuovo coronavirus alle nostre cellule (1.9,[1])

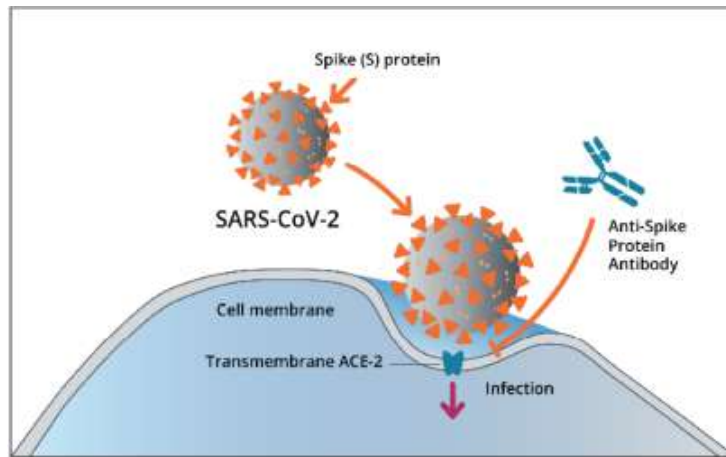


Figura 1.9: Covid e recettore ACE-2

È dimostrato come le droplets (particelle di saliva con dimensioni maggiori di $5\mu\text{m}$) rilasciate dagli starnuti e dai colpi di tosse siano la modalità di contagio primaria del virus ed è anche dimostrato come esse vengano facilmente stoppate dalla resistenza dell'aria e depositate al suolo in raggio di un metro per effetto della gravità.

Le particelle più piccole emanate dalla respirazione possono invece restare in aria per molto più tempo, per questo è plausibile che l'inalazione di queste ultime portino ad una nuova via di contagio. Tuttavia bisogna considerare diversi fattori per valutare l'effettiva probabilità di contagio.

Uno degli elementi da considerare è l'emivita del virus nell'aerosol. L'emivita di un agente in generale è il tempo per cui la sua attività si riduce del 50%. Ricerche recenti dimostrano come l'emivita di sars-cov-2 aerosolizzato sia di circa un'ora in ambiente controllato e la sua titolazione virale in esso resta alta per più ore come mostrato nella prossima figura (1.10):

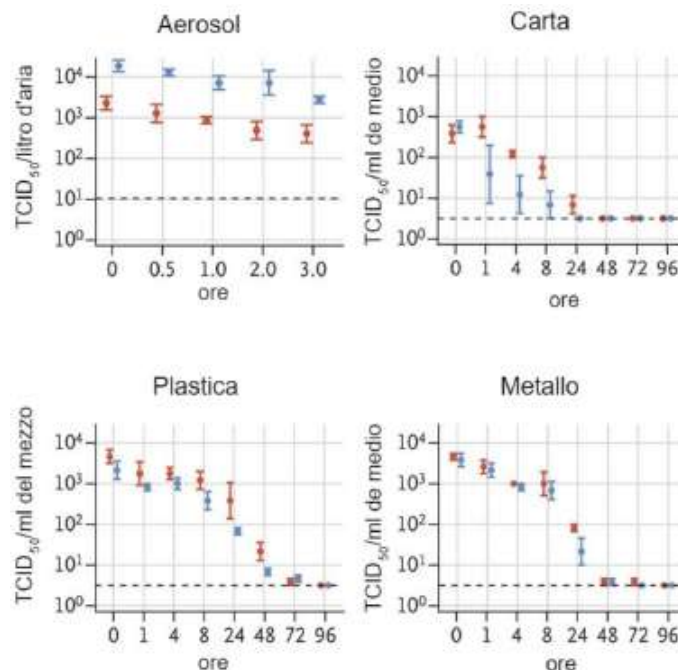


Figura 1.10: Titolazione virale sars-cov-2

Un'altro aspetto da considerare è la concentrazione delle particelle infette nell'atmosfera sia in ambienti aperti che chiusi ([2]). In Italia si sono effettuati test su 34 campioni di PM10

prelevati in ambienti industriali della provincia di Bergamo per un periodo continuativo di 3 settimane. Le analisi confermano la presenza di RNA del SARS-Cov-2 in almeno 8 delle 22 giornate esaminate.

Un'analisi di campioni di aerosol presi in alcune aree pubbliche di Wuhan nel mese di Febbraio 2020, hanno evidenziato come la concentrazione di particelle infette sia così bassa da rendere pressoché nulla la probabilità di inalazione di queste, e quindi improbabile il contagio tramite esse (3 copie/m³ contro le 10⁸ particelle totali).

La situazione risulta essere completamente diversa in ambienti chiusi come ospedali, in cui possono esserci individui infetti ed uno scarso ricircolo di aria. In questi ambienti le condizioni meteorologiche sono più stabili, dando quindi vita ad un ambiente favorevole alla sopravvivenza del virus. Un'analisi di campioni di aerosol in due ospedali di Wuhan hanno mostrato un'alta concentrazione di particelle contenenti RNA virale nelle aree con presenza di pazienti infetti (19 copie/metro cubo) e nelle aree atte alla rimozione dei dispositivi di sicurezza del personale medico (18-42 copie/metro cubo). Pertanto risulta essere molto più alto il rischio di contagio dovuto alle particelle presenti in aria.

L'individuazione del virus sulle PM però non dimostra l'effettiva possibilità che esse costituiscano una nuova e diversa modalità di contagio. Bisogna dunque fare molta attenzione nel tradurre una elevata concentrazione delle PM in una eccessiva vulnerabilità della popolazione ai virus, bisogna continuare le ricerche in quanto nulla è stato ancora dimostrato.

Quello che possiamo fare è monitorare le concentrazioni di particolato atmosferico ed utilizzarle come campanello di allarme e tentare di contenere la proliferazione del virus.

2 Implementazione di un sensore per la misura della densità di particolato ambientale all'interno di DoNet

2.1 Che cos'è DoNet?

DoNet è un dispositivo IoT e come tale è un device elettronico capace di comunicare direttamente o indirettamente con la rete Internet.

Il nome deriva dal gioco di parole con il termine "Donut" dove "Don" sta per Donut e "net" sta per network (2.1):



Figura 2.1: DoNet

I Donet sono in grado di rilevare i parametri ambientali e inviarli al broker remoto attraverso la connessione con l'access point locale. Essi sono composti da un microcontrollore e da una serie di sensori per la misura di tali parametri.

2.1.1 Chip

Per quanto riguarda l'acquisizione dei dati dai sensori si è scelto di utilizzare un chip che attualmente fa da padrone nell'ambito dell'IOT, l'ESP32 ([3]) di *Espressif System* figlio del meno recente ESP8266. Il motivo per cui è molto utilizzato è che presenta al suo interno tutti gli hardware necessari alla connettività wireless come wifi e bluetooth, oltre che per le sue elevate prestazioni computazionali.

ESP32 è un sistema a bassi consumi basato su una serie di microcontrollori per la gestione della connettività ed alimentato da un microprocessore *Tensilica Xtensa LX6 dual-core*.



Figura 2.2: Esp32

Nella tabella 1 vengono mostrate le principali caratteristiche tecniche di ESP32 confrontate con il suo predecessore ESP8266. Si possono facilmente notare le prestazioni superiori del chip più recente, infatti oltre a supportare la tecnologia Bluetooth, presenta un WiFi più performante (fino a 150 Mbps) del suo predecessore (72.2 Mbps). ESP32 ha il grande vantaggio di possedere molti più pin general purpose, la maggior parte dei quali selezionabili tramite firmware come pin di comunicazione per il vari protocolli supportati (SPI, I2C, UART) grazie alle funzioni integrate di multiplexing che consentono di assegnare funzionalità diverse allo stesso pin. Inoltre ESP32 presenta 2 canali diversi per il DAC (funzione fondamentale per la riuscita di questa ricerca) oltre che risolvere problemi di utilizzo simultaneo di ADC e WiFi riscontrati nell'ESP8266.

Specifiche tecniche		
Hardware	Esp32	Esp8266
CPU	Tensilica Xtensa LX6 32 bit Dual-Core at 160/240 MHz	Tensilica LX106 32 bit at 80 MHz (up to 160 MHz)
SRAM	520kB	36 kB disponibili
FLASH	2 MB (max 64 MB)	4 MB (max 16 MB)
Voltage	from 2.2 to 3.6 V	from 3.0 to 3.6 V
Operating Current	80 mA di media	80 mA di media
WiFi	802.11 b/g/n	802.11 b/g/n
Bluetooth	Bluetooth 4.2 + BLE	AGO
GPIO	36	17
Software PWM	16 canali	8 canali
SPI/I2C/I2S/UART	4/2/2/2	2/1/2/2
ADC	12 bit 18 canali	10 bit 1 canale
DAC	8 bit 2 canali	

Tabella 1: Confronto ESP32-ESP8266

2.1.2 Sensori

I sensori utilizzati da DoNet sono i seguenti:

1. Il sensore **Sth21**, prodotto dalla *Sensirion*, composto da due trasduttori, uno per la misura di temperatura e uno per la misura dell'umidità relativa, un amplificatore, un convertitore analogico-digitale (ADC), una memoria OTP (One Time Programmable), un'unità di elaborazione digitale e un'interfaccia I^2C ;
2. Il sensore **CCS811**, prodotto dalla *AMS*, per monitorare la qualità dell'aria. Esso integra: un sensore di gas a ossido di metallo (MOX), per rilevare una vasta gamma di composti organici volatili (VOC), un microcontrollore con un convertitore analogico-digitale e un'interfaccia I^2C . Il sensore è in grado di fornire valori di CO2 (anidride carbonica) in ppm e di TVOC (composti organici volanti totali) in ppb.
3. Il sensore **VEML7700**, prodotto dalla *Vishay*, per misurare l'intensità di luce in lux. È composto da un fotodiodo, un amplificatore, un convertitore analogico-digitale e supporta la comunicazione mediante interfaccia I^2C .

Nell'hardware del dispositivo sono stati aggiunti anche dei led per aumentare il livello di user-friendliness.

Il dispositivo, inoltre, viene alimentato da una tensione di 5V che viene immediatamente convertita a 3,3V.

2.1.3 Broker MQTT

Si è scelto di installare in un computer Raspberry un broker Mosquitto MQTT, per interconnettere tutti i messaggi inviati dai diversi nodi. Un computer Raspberry è un single-board computer progettato per ospitare sistemi operativi basati su kernel Linux o RISC OS. La versione di Raspberry di cui si fa uso è la Raspberry Pi 3 Model B (2.3) avente le seguenti caratteristiche hardware e software:



Figura 2.3: Computer Raspberry Pi 3 Model B

- SOC Broadcom BCM2837;
- CPU da 1.2 GHz 64-bit quad-core ARM Cortex-A53;
- Memoria SDRAM: 1GB LPDDR2 (900 MHz);
- Sistema operativo: Ubuntu 18.04.3 LTS.

Come server MQTT si è scelto Mosquitto perchè è open source ed è il più diffuso. Per installare questo servizio in un computer con SO Ubuntu è sufficiente aprire il terminale ed eseguire i seguenti comandi:

```
$ sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
$ sudo apt-get update
$ sudo apt-get install mosquitto
```

Installato il broker MQTT si può installare anche il client:

```
$ sudo apt-get install mosquitto-clients
```

Dopo aver eseguito questi comandi il server sarà attivo. Per verificare il suo stato, è possibile digitare il comando:

```
$ sudo service mosquitto status
```

Per fermare il server MQTT è sufficiente digitare il comando:

```
$ sudo service mosquitto stop
```

Per avviare il server MQTT è possibile digitare il comando:

```
$ sudo service mosquitto start
```

2.1.4 Programmazione dei dispositivi IoT

Per la programmazione dei dispositivi IoT è stato utilizzato l'ambiente di sviluppo integrato (Integrated Development Environment IDE), *Thonny*. Thonny è un ambiente di sviluppo per Python IDE pensato per i principianti e i programmatori che desiderano imparare a sviluppare applicazioni in questo linguaggio.

All'interno di tale software oltre a programmare in Python è disponibile l'opportunità di implementare all'interno della scheda ESP32 il linguaggio di programmazione *MicroPython*.

2.1.4.1 MicroPython

MicroPython ([4]) è un'implementazione software, scritta in linguaggio C, del noto linguaggio di programmazione Python 3 ottimizzato per la sua esecuzione in ambienti dalle prestazioni ridotte come i microcontrollori o sistemi embedded in generale. Il progetto è interamente open source ed il codice sorgente è interamente disponibile su GitHub. Al suo interno contiene alcuni moduli caratteristici del linguaggio Python oltre all'implementazione di moduli atti all'accesso da parte del programmatore all'hardware di basso livello.

MicroPython è ricco di funzionalità avanzate come un prompt interattivo per il debug, comprensione delle varie strutture dati native di Python, gestione delle eccezioni e molto altro. È comunque abbastanza compatto da adattarsi e funzionare in soli 256 kB di memoria Flash e 16 kB di RAM. Come anticipato precedentemente vengono fornite all'interno di MicroPython dei moduli per l'accesso all'hardware, customizzati per i diversi tipi di controllori, tra i più comuni troviamo moduli per il controllo di GPIO, ADC, DAC, PWM, I2C, SPI ed UART.

In generale MicroPython risulta una valida alternativa ai classici linguaggi di programmazione per sistemi embedded come C, C++ e/o Arduino in quanto essendo MicroPython basato su un interprete Python i codici sviluppati risultano molto più puliti ed human friendly, caratteristica che rende lo sviluppo del codice molto più rapido, in più è possibile far eseguire al controllore un comando alla volta rendendo il debug del codice molto più rapido ed efficiente.

2.1.4.2 MicroPython per ESP32

In questo paragrafo vengono mostrati i passaggi necessari alla programmazione di ESP32 con MicroPython. L'intero progetto è stato sviluppato in un device con installato un sistema operativo Windows 10, per cui le informazioni seguenti sono valide per dispositivi con la stessa tipologia di sistema operativo.

Si necessita di un interprete Python per l'installazione dei vari pacchetti che si necessita. Per l'installazione di Python si rimanda il lettore alla sezione downloads del sito ufficiale ([5]). Con l'installazione di Python è incluso il package manager *pip* che verrà utilizzato per l'installazione dei vari pacchetti.

A questo punto è necessario tramite il package manager installare il tool per la gestione della memoria flash dei chip della casa produttrice Espressif System : *esptool.py*. Per l'installazione di quest'ultimo aprire il prompt dei comandi (digitando "cmd" sulla barra di ricerca di Windows) e lanciare il comando:

Pip install esptool

A questo punto bisognerà caricare il firmware di MicoPython sulla memoria flash dell'ESP32. Per farlo bisogna scaricare dal sito ufficiale MicroPython ([6]) nella sezione download il file di estensione .bin relativo al microprocessore che stiamo utilizzando (nel nostro caso ESP32). Prima di caricare il firmware sul controllore e di iniziare a far eseguire comandi allo stesso, bisogna cancellare completamente tutti i dati sulla memoria flash. Per fare ciò bisogna collegare la board al computer e, dal prompt dei comandi, digitare il seguente comando:

esptool .py -port (porta_seriale) erase_flash

A questo punto, per caricare il firmware sulla memoria flash basta lanciare il comando tenendo premuto il tasto BOOT della scheda:

esptool .py -chip esp32 -port (porta_seriale) write_flash -z 0 x1000 (path_firmware)

In "porta_seriale" va sostituita la porta seriale corrispondente alla nostra board (nel mio caso COM3), mentre per "path_firmware" si intende la directory in cui è allocato il file .bin scaricato precedentemente.

2.1.4.3 Gestione della memoria e programmazione della scheda

Siamo a tutti gli effetti pronti a programmare la scheda utilizzando degli script MicoPython. Vale la pena discutere il file system ed alcuni files che MicroPython crea nella root directory:

- **boot.py**: è il file che viene eseguito all'accensione del sistema. Contiene codice di basso livello per l'avvio del controllore;
- **main.py**: questo file non viene creato di default, ma è il codice che viene eseguito immediatamente dopo l'esecuzione del precedente. Qualsiasi modulo sviluppato deve essere contenuto e/o richiamato all'interno di questo file, altrimenti non viene eseguito.

Per le applicazioni di grandi dimensioni, nonostante la memoria di ESP32 risulti relativamente capiente, potrebbe essere necessario rendere i files i più "compatti" possibile, al fine di ottimizzare l'occupazione della memoria. Per questi scopi vi sono 3 tipologie di files supportati da MicroPython:

- **Estensione .py**: sono i files sviluppabili da desktop e vengono compilati a runtime dal compilatore MicroPython per essere inseriti poi in memoria RAM ed essere eseguiti. La memoria RAM del dispositivo è divisa in 3 zone distinte: la prima è dedicata ai moduli .py (come main.py), la seconda è dedicata alla compilazione del codice e la terza rimane a disposizione per l'esecuzione del programma, per questo motivo questa tipologia di files è utilizzabile solo per applicazioni di dimensioni relativamente piccole;
- **Estensione .mpy**: partendo dai files di estensione .py è possibile precompilare i codici (tramite il tool mpy-cross) riducendone di molto le dimensioni. I files .mpy vengono comunque gestiti come la tipologia precedente per cui, con moduli di dimensione molto grande, si potrebbero avere gli stessi problemi;
- **Moduli congelati**: sono i files .bin come il firmware MicroPython. Vengono caricati sulla memoria flash del sistema e per questo sono le tipologie di precompilati da utilizzare per i codici molto lunghi. Per poter generare uno di questi file è necessaria la toolchain esp-opend-sdk.

Qualunque sia la tipologia di file che si decide di utilizzare, per il loro caricamento sulla scheda e la loro consecutiva esecuzione da parte della stessa, si necessita di un'altro tool sviluppato da Adafruit chiamato ampy. Per la sua installazione basta aprire il prompt dei comandi e lanciare il comando:

pip install adafruit - ampy

Per caricare un file sulla nostra scheda bisogna navigare all'interno della directory fino alla cartella contenente il file in questione, per poi digitare il comando:

ampy -p (porta_seriale) -b 115200 put (nome_file)

A questo punto abbiamo tutte le informazioni utili alla programmazione della scheda.

2.1.5 Ponte per la connessione da USB a TTL

Per implementare la connessione da USB a TTL, necessaria per il corretto funzionamento del CHIP esp32 integrato all'interno di DoNet, è fondamentale utilizzare un **Chipset FT232RL** (2.4) che comprende principalmente:

- Un indicatore di comunicazione del ricetrasmittitore RXD/TXD;
- Sei pin che da destra a sinistra sono: DTR, RXD, TXD, VCC, CTS, GND.

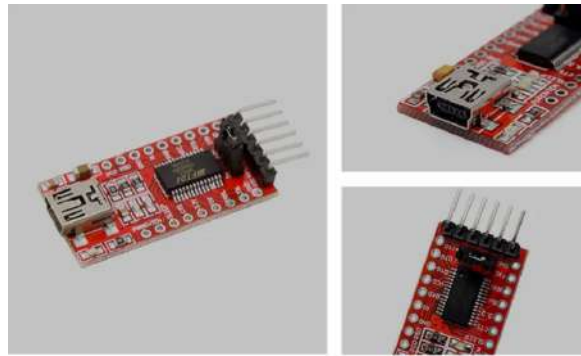


Figura 2.4: Chipset FT232RL

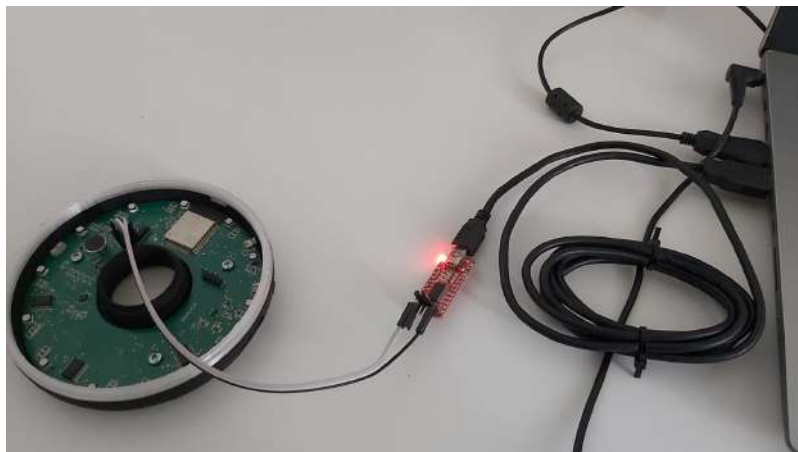


Figura 2.5: connessione da USB a TTL

Utilizzando tale ponte, la connessione risulterà essere molto più solida e stabile. Prima di tale implementazione evidenziava molti problemi, che ora andremo ad elencare, che però, grazie a tale soluzione, sono stati risolti:

- Connessione instabile;
- Crash inaspettati del PC;
- Non sempre il dispositivo si connetteva correttamente;
- Non sempre il dispositivo si connetteva al primo tentativo;
- Non sempre la scrittura dei file andava a buon fine;
- La riuscita della connessione sembrava randomica.

2.2 Sensori per la misura del particolato ambientale

Per poter implementare un sensore per il rilevamento delle micro polveri (particolato ambientale) all'interno di DoNet, risulta necessario passare in rassegna due delle diverse tecnologie attualmente presenti sul mercato ed effettuare una caratterizzazione di alcuni dei sensori così da poter realizzare una scelta appropriata.

2.2.1 Stato dell'arte

Prima di analizzare una rassegna di sensori di questo tipo, andremo ad analizzare le due tecnologie più importanti su cui essi si basano.

Vedremo come alcuni principi risultano molto più efficaci di altri a discapito di dimensioni e costi eccessivamente elevati. Bisogna anche tener presente che, per la nostra applicazione, necessitiamo di un sensore con alcune caratteristiche:

- **Dimensioni ridotte:** per poter essere implementato all'interno di un dispositivo che verrà utilizzato in ambienti chiusi;
- **Costi ridotti:** in quanto il dispositivo deve poter essere acquistabile da utenti;
- **Accuratezza:** il sensore in questione dovrà fornire delle buone misurazioni, dovrebbe essere poco rumoroso e dovrà risultare poco dispendioso in termini di energia elettrica.

2.2.1.1 LED infrarossi

La prima tecnologia che prendiamo in considerazione è quella più economica. Essa basa il suo funzionamento sul principio della diffrazione della luce.

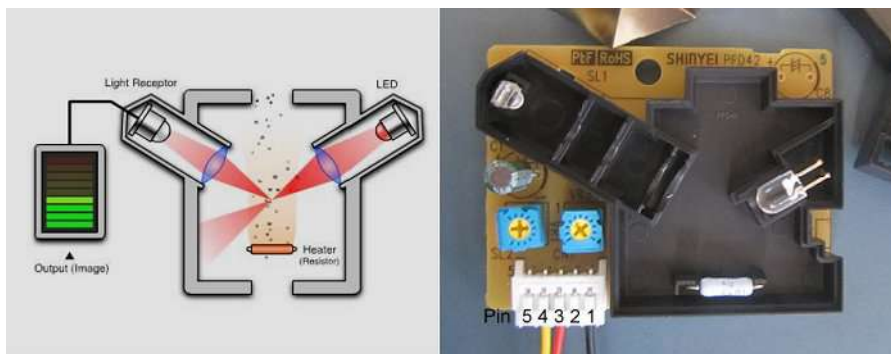


Figura 2.6: Tecnologia led

Il rilevatore è una camera attraversata da una corrente d'aria. Nel caso delle immagini sopra il flusso d'aria all'interno della camera viene generato da una resistenza che, scaldandosi, genera un moto convettivo dell'aria portandola a muoversi verso l'alto trascinando con se le particelle. Su un lato della camera c'è un emettitore a infrarossi e sull'altro lato c'è un sensore a infrarossi.

Le particelle che attraversano la camera riflettono il fascio di luce ed attivano il trasduttore posto dal lato opposto. In questo modo è possibile "contare" le particelle che attraversano la camera di misurazione ottenendo una stima delle concentrazioni di particolato.

Purtroppo la qualità delle misurazioni di questo tipo di tecnologia è utilizzabile per dare una stima della qualità dell'aria approssimativa in quanto presentano grandi difficoltà nel distinguere le dimensioni effettive delle particelle che lo attraversano.

2.2.1.2 Scattering laser

Questa tecnologia è molto simile alla precedente per quanto riguarda il suo principio di funzionamento. Diversamente da prima, questa tipologia di sensori utilizza un fascio laser come sorgente luminosa che aumenta notevolmente le prestazioni grazie al diametro ridotto del fascio stesso.

All'incirca, il funzionamento di tale tecnica si divide come prima in quattro eventi:

1. Il fascio laser incidente colpisce la particella;
2. La luce del fascio laser interagisce con la particella;
3. La luce esce dalla particella (scattera);
4. La luce che fuoriesce viene rilevata e analizzata dai rilevatori e dal software.

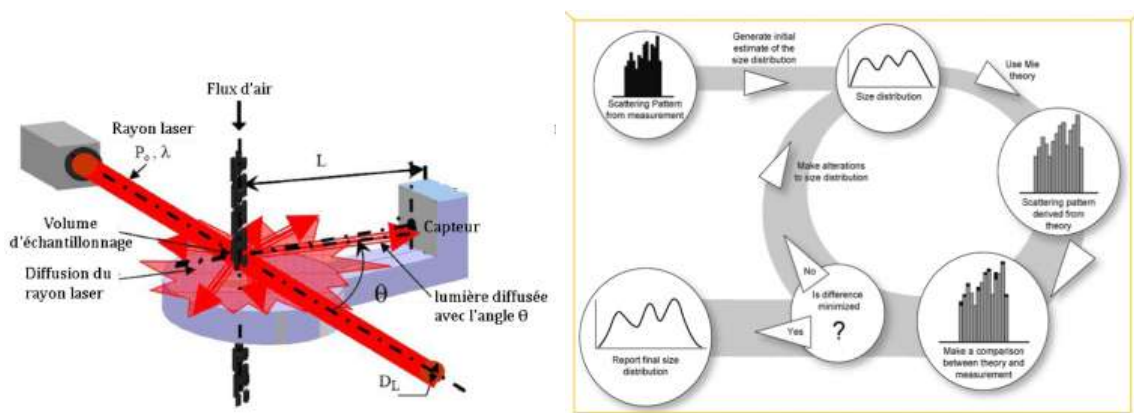


Figura 2.7: Tecnologia laser

2.2.1.3 Considerazioni

Prendiamo quindi in considerazione solo sensori basati sul principio della diffrazione della luce ed effettuando alcune considerazioni, mettiamo in evidenza le loro differenze costruttive e le loro diverse prestazioni.

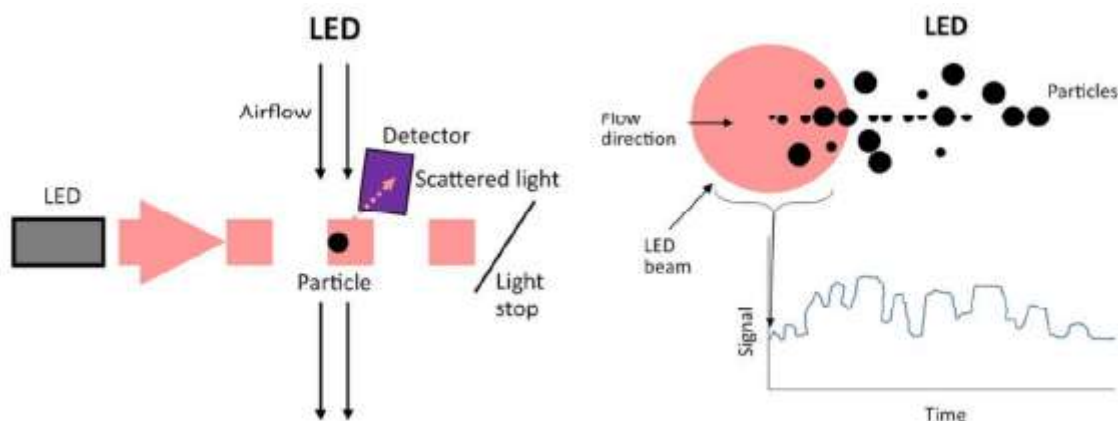


Figura 2.8: Schema implementativo tecnologia led

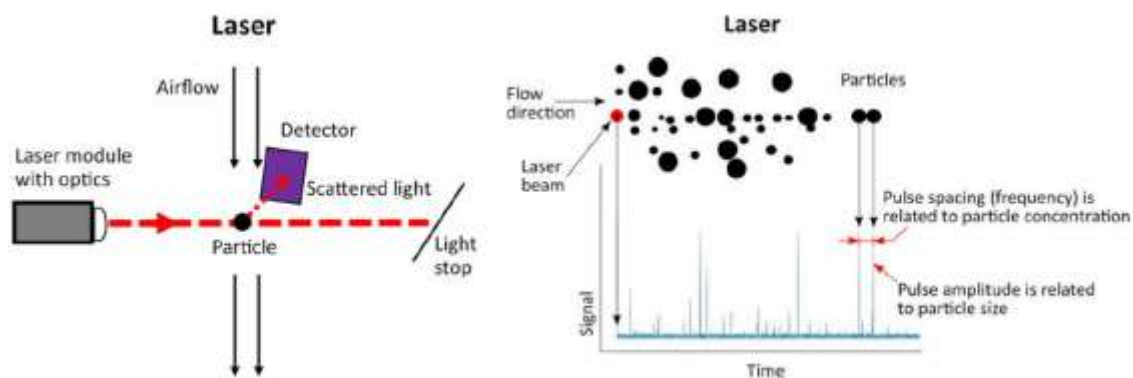


Figura 2.9: Schema implementativo tecnologia laser

Essendo che un fascio laser (2.9) ha un diametro molto inferiore di un fascio LED (2.8) le particelle restituiscono segnali individuali, ed in base all'ampiezza del segnale siamo in grado di distinguere le dimensioni delle particelle stesse.

Come si nota dalla figura inerente la tecnologia LED, a causa dell'ampiezza del fascio le particelle di diametro più piccole non sono facilmente rilevabili.

Va inoltre presa in considerazione la modalità di generazione del flusso d'aria da campionare. Infatti vi sono due soluzioni differenti:

1. **Resistenza:** alcuni sensori presentano una resistenza che scaldandosi genera un moto convettivo che porta l'aria a muoversi verso la stazione di rilevamento trascinando con se le particelle da misurare;
2. **Ventola:** altri sensori invece presentano una ventola attraverso che genera un flusso d'aria costante.

Nonostante la rumorosità nulla e l'efficienza energetica della prima soluzione, bisogna tenere a mente che il flusso costante generato da una ventola ci permette di effettuare delle misurazioni più accurate.

2.3 Valutazione di alcuni dei sensori in commercio

Si sono presi in considerazione sensori low cost per il rilevamento del particolato ambientale, analizzandone diverse caratteristiche. Paragonando poi tali dispositivi è stata presa una decisione su quale implementare all'interno di DoNet.

I sensori che sono stati selezionati sono l'**Omron B5W-LD0101** di *Omron Electronics*, l'**SM-UART-04L** di *Amphenol Advanced Sensors*, il **Tera NEXT-PM** di *Tera Sensors* e lo l'**HUMAI 150** di *Humai*.

Le caratteristiche prese in considerazione sono:

1. **Prezzo:** fondamentale per poter ottenere un prodotto dal prezzo abbordabile e quindi reperibile a qualsiasi tipo di utente;
2. **Interfaccia di comunicazione:** per una più semplice implementazione;
3. **Produttore:** per l'affidabilità costruttiva;
4. **Dimensioni:** per non andare a modificare in maniera radicale il design di DoNet;
5. **Consumi:** così da ottenere un dispositivo che non abbia dei consumi eccessivamente elevati.

2.3.1 SM-UART-04L



Figura 2.10: SM-UART-04L

Questo sensore, prodotto da *Amphenol Advanced Sensors* e venduto al prezzo di circa 20€, utilizza una tecnologia laser con generazione del flusso d'aria da campionare tramite ventola. Come interfaccia di comunicazione utilizza un protocollo seriale (*UART*) ed è in grado di rilevare particelle del diametro compreso tra i 0.3 ed i 10 μm fornendo misurazioni di PM_{10} , $\text{PM}_{2.5}$ e PM_{10} .

Risulta essere poco dispendioso in termini energetici (circa 60mA in working mode) ed il produttore fornisce un grafico di correlazione ottenuto incrociando i dati rilevati dal sensore stesso con quelli ottenuti da un monitor di aerosol di fascia media (TSI DRX), dati ottenuti tramite campionamento del fumo di sigaretta:

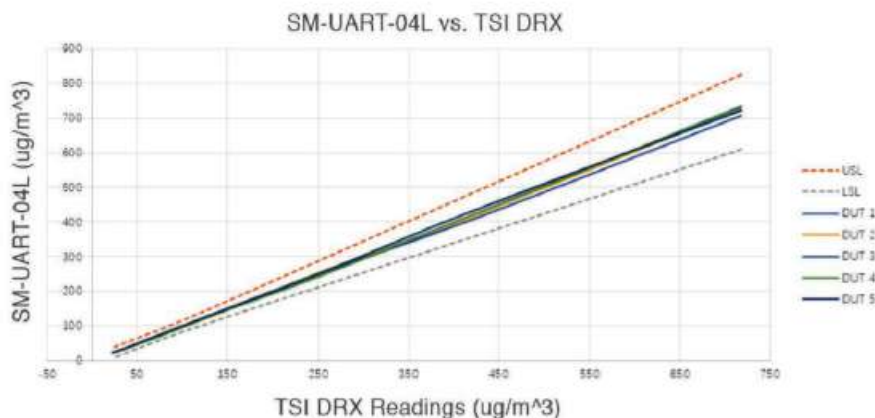


Figura 2.11: SM-UART-04L datas

Come si nota dal grafico in figura 2.11 risulta esserci una forte linearità tra i dati per cui il sensore in questione sembra essere affidabile. Scendendo più nel dettaglio, la trasmissione delle informazioni avviene ad una velocità di 9600 bps (bit per secondo) e presenta due diverse modalità di scambio dati:

1. **Continuous-Mode:** i dati vengono inviati sul bus continuamente fino a che non si riempie il buffer;
2. **Answer-Mode:** i dati vengono inviati solo su richiesta.

Per quanto riguarda le caratteristiche tecniche, il sensore viene alimentato con una tensione tra i 4 ed i 5.5 V e presenta dei consumi di corrente che variano tra i 60 ed i 100 mA. Il produttore garantisce errori sulle misurazioni di $\pm 10 \mu\text{g}/\text{m}^3$ per le misurazioni di concentrazione di particolato nell'intervallo 1-100 $\mu\text{g}/\text{m}^3$, oltre che una vita del sensore di 40 000 ore di lavoro continuo. Per maggiori informazioni riguardo le caratteristiche si rimanda alla figura 2.12:

Absolute Maximum Ratings

Absolute Maximum Ratings			
Parameter	Symbol	Rating	Unit
Supply Voltage	V _{cc}	0 to + 5.5	V
Operating Temperature	T _{opr}	-10 to 50	°C
Storage Temperature	T _{stg}	-30 to 70	°C
Operating Humidity ⁽¹⁾	RH _{opr}	0 to 95	%
Storage Humidity ⁽¹⁾	RH _{stg}	0 to 95	%

1) Non-condensing

Electrical Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
Particle Size	D	0.3	2.5	10	um
Detection Range	D _{reg}	1	—	999	µg/m ³
Resolution	R	—	1	—	µg /m ³
Indication Error ⁽²⁾	D _{err}	1 ~100 ug/m ³	—	—	+/-10 µg /m ³
		100~999 ug/m ³	—	—	+/-10 %
Warm-Up Time	t _{wup} ⁽²⁾	—	5	—	s
Response Time	t _{rsp} ⁽²⁾	—	1	—	s
Laser Life (Average Time Before Re-Calibration)	T	—	40,000	—	hour
Supply Voltage	V _{cc}	4.8	5	5.2	V
Supply Voltage Ripple	V _{cc} Ripple	—	—	30	mV
Current Consumption	I _{cc} ⁽²⁾	—	60	100	mA
Output (UART)	3.3V				

1) Non-condensing

2) Testing at T=25°C, RH=40-60%

Figura 2.12: SM-UART-04L caratteristiche

Conversione Dati

Per l'applicazione che si vuole sviluppare si è deciso di utilizzare il sensore in Answer-Mode al fine di ottenere dati il più sincronizzati possibile ai dati degli altri sensori.

I comandi al sensore vanno inviati secondo una precisa forma descritta dalla figura 2.13, dove i byte D1 e D2 dipendono dal comando (CMD) che si vuole inviare al sensore secondo la figura 2.14. In risposta al comando inerente l'acquisizione dati dal sensore quest'ultimo risponde inviando sulla linea di comunicazione 32 bytes.

Head 1	Head 2	CMD	D1	D2	CRC1	CRC2
0x42	0x4D	CMD	DATAH	DATAL	LRCH	LRCL

Figura 2.13: SM-UART-04L caratteristiche

CMD	DATAH	DATAL	Description	Response
0xE2	X	X	Get reading	Same format as above table
0xE1	X	0x00 – Ask-answer mode	Output mode switch	0x42 0x4D 0x00 0x04 0xE1 0x00 0x01 0x74
		0x01 – Direct output mode		0x42 0x4D 0x00 0x04 0xE1 0x01 0x01 0x75
0xE4	X	0x00 – Standby mode	Standby control	0x42 0x4D 0x00 0x04 0xE4 0x00 0x01 0x77
		0x01 – Working mode		

Figura 2.14: SM-UART-04L caratteristiche

Molti di essi risultano essere superflui o riservati per cui irrilevanti alla conversione dei dati. I bytes che ci interessano al fine di ottenere le misurazioni sono:

1. **bytes 5 e 6:** $\text{PM}_1 \mu\text{g}/\text{m}^3$;
2. **bytes 7 e 8:** $\text{PM}_{2.5} \mu\text{g}/\text{m}^3$;
3. **bytes 9 e 10:** $\text{PM}_{10} \mu\text{g}/\text{m}^3$;
4. **bytes 30:** contenente i vari errori a cui il sensore potrebbe essere sottoposto;
5. **bytes 31 e 32:** crc, per il check di buona riuscita della trasmissione.

Le misurazioni inerenti le concentrazioni di particolato vengono fornite separate in 2 bytes. Per ottenere i valori di particolato calcolati dal sensore bisogna applicare la seguente formula:

$$\text{PM} = 256b_i + b_{i+1}$$

dove i dipende dai valori di particolato che ci interessano secondo l'elenco di sopra.

Utilizzando la stessa formula sugli ultimi 2 bytes otteniamo il CRC (*Cyclic redundancy check*) che viene generato sommando i valori dei bytes precedenti. Se il valore del CRC risulta uguale alla somma di tutti i precedenti allora la trasmissione è andata a buon fine ed i dati sono effettivamente utilizzabili.

I bit del byte 30 contengono gli errori a cui il sensore potrebbe essere sottoposto (2.15):

L_D13	...	Error code 0b0ABCDEFGG A = 1 Laser error B = 1 Laser alarm C = 1 High temperature alarm D = 1 Low temperature alarm E = 1 Fan error F = 1 Fan speed compensation start G = 1 Fan speed alarm
-------	-----	---

Figura 2.15: SM-UART-04L possibili errori

2.3.2 HUMA-I 150



Figura 2.16: Huma-i 150

Huma-i black 150 è un monitor portatile della qualità dell'aria, che misura CO₂, VOC (Volatile Organic Compounds), PM_{2.5}, PM₁₀, temperatura e umidità.

Viene prodotto dalla Humai e venduto al prezzo di 130 euro circa.

Il ciclo di funzionamento di tale sensore è illustrato all'interno della prossima figura (2.17):

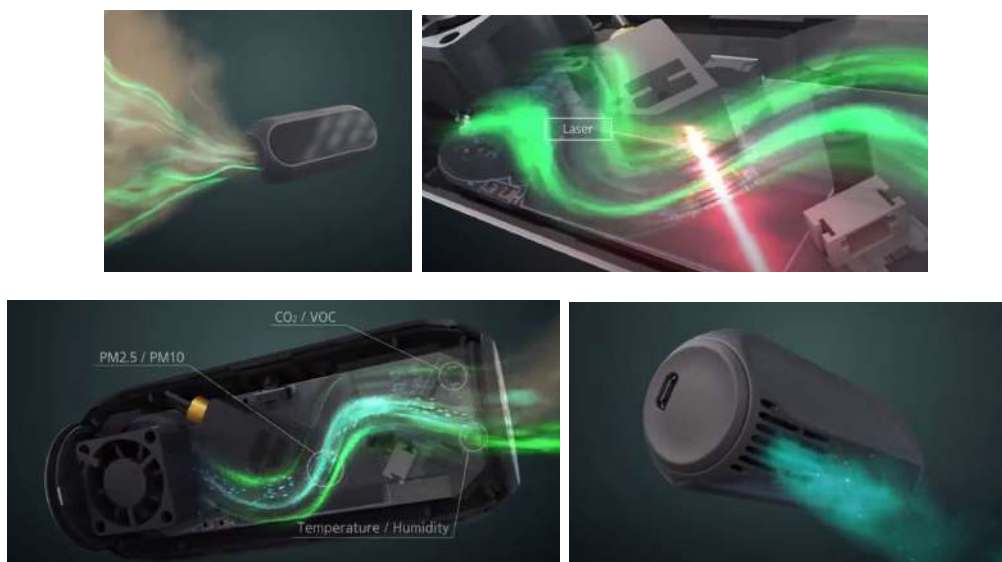


Figura 2.17: Ciclo di funzionamento Huma-i

Andando più nel dettaglio, lo Huma-i possiede un unico bottone che può svolgere diverse funzioni.

Oltre all'accensione, ogni volta che si preme il pulsante di selezione della modalità (power button), la modalità di visualizzazione cambia ciclicamente nel seguente ordine:

1. **Modalità iAQI** (*indice di qualità dell'aria*): l'IQA costituisce uno strumento di comunicazione che rende facilmente accessibili le informazioni sullo stato della qualità dell'aria (2.18). Stabilisce, attraverso il confronto con i parametri della normativa vigente, il grado di soddisfazione della qualità dell'aria e quindi anche il grado di protezione della salute umana;

VALORE NUMERICO	INDICE NUMERICO
0-50	1
51-75	2
76-100	3
101-125	4
125-150	5
151-175	6
>175	7

Figura 2.18: Informazioni sullo stato della qualità dell'aria

2. **Modalità display integrato:** PM_{2.5} / VOC / CO₂;
3. **Modalità OMS:** modalità visiva;
4. **Modalità PM:** visualizzazione grafica dei valori di PM_{2.5} e PM₁₀;
5. **Modalità VOC:** visualizzazione grafica del valore di VOC;
6. **Modalità CO₂:** visualizzazione grafica del valore di CO₂;
7. **Modalità temperatura/umidità (%RH):** visualizzazione grafica dei valori di temperatura e umidità.

Inoltre, il led di stato permette di conoscere lo stato di CO₂, VOC, PM_{2.5} e PM₁₀ in base ai loro valori (2.19):





Unit of measurement		Good 	Moderate 	Poor 	Unhealthy 
$\mu\text{g}/\text{m}^3$	PM2.5	0~15	16~25	26~35	36~
	PM10	0~30	31~50	51~70	71~
ppm	VOC	0.000 ~ 0.220	0.221 ~ 0.660	0.661 ~ 2.200	2.201 ~ 10.000
ppm	CO ₂	GOOD (400 ~ 700)	MODERATE (701 ~ 1,000)	POOR (1,001 ~ 2,000)	UNHEALTHY (2,001 ~ 10,000)

Figura 2.19: Valori di stato variabili Huma-i 150

Infine, riportiamo alcune delle specifiche tecniche (2.20):

Product	Portable Air Quality Monitor
Model No.	HH-150
VOC	0.000 ~ 10.000ppm
CO ₂	400 ~ 10,000ppm
PM2.5 & PM10	0 ~ 500 $\mu\text{g}/\text{m}^3$
Power Supply	Li-Polymer Battery
Charge Mode	USB micro 5 Pin
Operating Temperature	0 ~ 40°C (32 ~ 104°F)
Storage Temperature	-20 ~ 60°C (-4 ~ 140°F)
Dimension	91 X 41 X 37mm (3.6 X 1.6 X 1.5 inches)
Weight	60g (0.13 pound)
Certifications	CE FCC KC

Figura 2.20: Specifiche tecniche Huma-i 150

2.3.3 Tera NEXT-PM



Figura 2.21: Tera NEXT-PM

Questo sensore è in grado di effettuare misurazioni di PM_1 , $PM_{2.5}$ e PM_{10} restituendo risultati sia in pcs/L che in $\mu g/m^3$, oltre che di temperatura ($^{\circ}C$) ed umidità percentuale. Inoltre è possibile cambiare il suo stato di lavoro in modo da risparmiare energia.

Dalle sue specifiche tecniche possiamo evincere (2.22) come i produttori di questo sensore garantiscano un'efficienza di rilevamento delle particelle più piccole rilevabili dal sensore maggiore del 50%, come l'errore di ripetibilità sia inferiore al 3% e l'errore di linearità inferiore al 5%. Per questo il sensore risulta essere molto affidabile e quindi un buon compromesso agli spettrometri di aerosol dal prezzo eccessivo.

Per quanto riguarda le specifiche elettriche, le dimensioni ed i protocolli di comunicazione facciamo invece riferimento alla figura (2.23).

Designation	Values	Units
GENERAL		
Technology	Optical	-
Targeted pollutants	Particulate Matter	-
Outputs	PM1 PM2.5 PM10 Temperature ¹ Relative Humidity ¹	$\mu g/m^3$ & pcs/L $^{\circ}C$ %
Airflow	2.5	L/min
Size	Annex 1	mm / Inches
Lifetime	>10 000	hour
PERFORMANCE		
Particle Size detection range	0.3 - 10	μm diameter
Detection efficiency with 0.3 μm diameter particles	> 50	%
Concentration detection range / PM_{10} - $PM_{2.5}$ - PM_1	0 - 1000	$\mu g/m^3$ (Arizona dust A1 equivalent)
Detection Limit	<1	$\mu g/m^3$ (Arizona dust A1 equivalent)
Linearity error	<5	%
Repeatability error ²	<3	%
Refresh rate	1 / 10 / 60	sec.
Warm-up time	10	sec.
Temperature influence 0 $^{\circ}C$ to 30 $^{\circ}C$ -20 $^{\circ}C$ to 0 $^{\circ}C$ 30 $^{\circ}C$ to 70 $^{\circ}C$	0 < +1.0 < -0.8	%/ $^{\circ}C$

Figura 2.22: Tera NEXT-PM specifiche tecniche

Designation	Values	Units
ELECTRIC SPECIFICATIONS		
Power supply	5.0	VDC
Power consumption in operation	<80 300	mA mA (Maximum)
Power consumption in Sleep Mode	<20	mA
COMMUNICATION		
UART / Modbus (RS485) ²	Download NextPM User Guide for more informations	
OTHER		
Operating conditions	-20 ÷ +70 253 to 343	°C K
	0 - 95 uncondensed	%
	500 ÷ 1500	hPa
Storage conditions	-20 ÷ +70	°C
	0 - 95 uncondensed	%
	500 ÷ 1500	hPa
Certifications	CE	
	RoHS compliant	
Dimensions and weight	L 62 mm x l 52 mm x H 23 mm 45 g L 2.4 / W 2.07 / H 0.9 inches 1.59 Oz	

Figura 2.23: Tera NEXT-PM specifiche elettriche

Protocolli di comunicazione

Il sensore in esame presenta la possibilità di comunicare con esso tramite due protocolli diversi. Il primo è un protocollo di comunicazione seriale molto utilizzato in molte applicazioni: *UART* (Universal Asynchronous Receiver / Transmitter(2.24)).

Si tratta di un protocollo di comunicazione bidirezionale ed asincrono che basa il suo funzionamento sull'utilizzo di due linee dati chiamate TX (per la trasmissione) ed RX (per la ricezione) e una linea di terra per discriminare il segnale digitale e consentire la comunicazione. La comunicazione si dice asincrona quando non dipende da un segnale di clock sincronizzato tra i due dispositivi che comunicano insieme. Il sistema UART funziona in full-duplex: ciò significa che i dati possono essere trasmessi e ricevuti contemporaneamente da entrambi i lati della comunicazione.

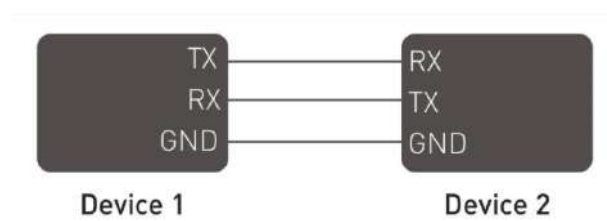


Figura 2.24: Protocollo UART

Il secondo è anch'esso un protocollo seriale molto utilizzato in ambiti industriali : *Modbus*. Tale protocollo (2.25) è ancora oggi uno dei più diffusi nel settore dell'automazione industriale nonostante le sue origini risalgano al 1979. La sua solida e duratura presenza nella vasta

panoramica dei protocolli industriali è dovuta al fatto che il Modbus è un protocollo semplice e diretto. Inoltre non è soggetto royalty ed è flessibile per cui può essere liberamente implementato ed adattato alle più svariate applicazioni.

Il protocollo Modbus è di tipo Master/Slave e quindi nella rete è presente sempre e solo un dispositivo Master che gestisce la comunicazione nei confronti di uno o più dispositivi Slave. Ogni scambio di informazioni è originato dal Master il quale invia un frame di bytes sul bus di campo contenente una particolare richiesta, normalmente un comando di lettura o di scrittura delle informazioni contenute in uno degli Slave. Tutti gli Slave sono normalmente in ricezione ed ascoltano le richieste del Master. Solo lo specifico Slave interrogato cattura le informazioni inviate dal Master, provvede all'esecuzione del comando e risponde al Master inviando a sua volta le proprie informazioni sulla rete.

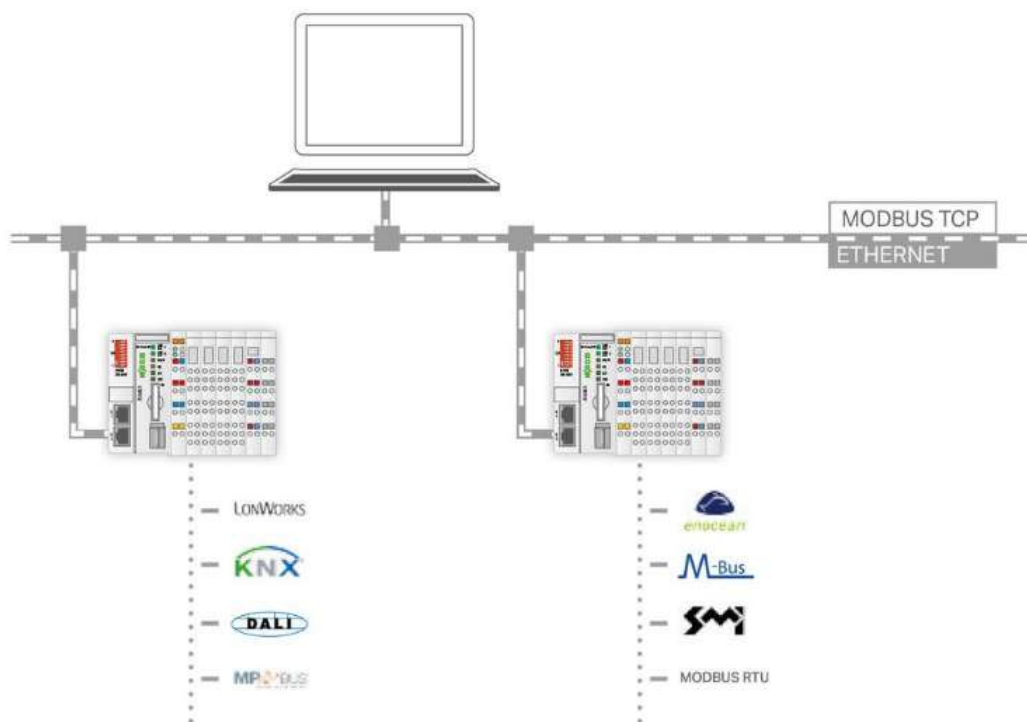


Figura 2.25: Protocollo ModBus

Conversione dei dati

Per l'acquisizione dati dal sensore in questione è stato utilizzato il protocollo UART descritto in precedenza, motivo per cui verrà approfondita unicamente la metodologia utilizzata per la conversione dei dati inerente a questo protocollo.

Come possiamo notare dalla tabella nella prossima figura (2.26), il produttore fornisce tutti i possibili comandi inviabili al sensore ed una loro descrizione:

Cmd_Id	Description	Example
0x11	Concentrations reading's averaged over 10 seconds and updated every 1 second	0x81 0x11 0x6E
0x12	Concentrations reading's averaged over 60 seconds and updated every 10 seconds	0x81 0x12 0x6D
0x13	Concentrations reading's averaged over 900 seconds and updated every 60 seconds	0x81 0x13 0x6C
0x14	Temperature and humidity readings	0x81 0x14 0x6B
0x15	Power on or sleep mode	0x81 0x15 0x6A (note 1)
0x16	Sensor state's readings	0x81 0x16 0x69
0x17	Firmware version readings	0x81 0x17 0x68
0x22	Modbus address read Modbus address 3 write (note 2)	0x81 0x22 0x00 0x5D 0x81 0x22 0x00 0x5A

Figura 2.26: Tera NEXT-PM comandi

Vengono fornite anche le tabelle inerenti le risposte del sensore ai vari comandi. In particolare il sensore risponde ai primi tre comandi mostrati nella precedente figura (2.26) con 16 bytes contenenti le informazioni.

1. **byte 1**: contiene l'indirizzo associato al sensore;
2. **byte 2**: contiene il comando a cui il sensore risponde;
3. **byte 3**: contiene informazioni riguardo lo stato del sensore;
4. **bytes 4 e 5**: informazioni inerenti la misurazione di PM_1 pcs/L;
5. **bytes 6 e 7**: informazioni inerenti la misurazione di $PM_{2.5}$ pcs/L;
6. **bytes 8 e 9**: informazioni inerenti la misurazione di PM_{10} pcs/L;
7. **bytes 10 e 11**: informazioni inerenti la misurazione di PM_1 $\mu\text{g}/\text{m}^3$;
8. **bytes 12 e 13**: informazioni inerenti la misurazione di $PM_{2.5}$ $\mu\text{g}/\text{m}^3$;
9. **bytes 14 e 15**: informazioni inerenti la misurazione di PM_{10} $\mu\text{g}/\text{m}^3$;
10. **bytes 16**: byte checksum per il controllo della buona riuscita della trasmissione dati.

Viene inoltre fornita la tabella (2.27) in cui sono riportati i fattori di conversione dei risultati ottenuti dal sensore:

	2 bytes data	Factor	Results
PM_1 pcs/L	0x32E7	1	13031
$PM_{2.5}$ pcs/L	0x32F5	1	13045
PM_{10} pcs/L	0x32F8	1	13048
PM_1 $\mu\text{g}/\text{m}^3$	0x006A	0.1	10.6
$PM_{2.5}$ $\mu\text{g}/\text{m}^3$	0x0072	0.1	11.4
PM_{10} $\mu\text{g}/\text{m}^3$	0x0085	0.1	13.3

Figura 2.27: Tera NEXT-PM conversione

Le concentrazioni di particolato sono quindi calcolate a partire da 2 bytes (b1 e b2) moltiplicando il risultato per il fattore (f), mostrato in tabella, come si evince dall'equazione:

$$PM = f(256b1 + b2)$$

Per quanto riguarda l'ultimo byte esso viene calcolato in maniera tale che la somma di tutti i byte, checksum compreso, sia un multiplo di 256 (0x100). In caso negativo i dati ottenuti non sono veritieri.

Il comando 0x14 invece è necessario all'ottenimento delle misurazioni di temperatura ed umidità. Il sensore risponde al comando con 8 bytes:

1. **byte 1**: contiene l'indirizzo associato al sensore;
2. **byte 2**: contiene il comando al quale il sensore sta rispondendo;
3. **byte 3**: contiene informazioni riguardo lo stato del sensore;
4. **bytes 4 e 5**: valori di temperatura;
5. **bytes 6 e 7**: valori di umidità;
6. **bytes 8**: checksum.

I bytes 1, 2, 3 ed 8 sono gli stessi inerenti al comando precedentemente discusso. Per quanto riguarda il calcolo di temperatura (T) ed umidità (H) vengono fornite le equazioni:

$$\begin{aligned} T &= 0.9754x_1 - 4.2488 \\ H &= 1.1768x_2 - 4.727 \\ x_1 &= (256b4 + b5)/10 \\ x_2 &= (256b6 + b7)/10 \end{aligned}$$

dove b4, b5, b6 e b7 sono i rispettivi bytes ricevuti in risposta dal sensore.

È reso disponibile inoltre un comando per la lettura dello stato del sensore (se il sensore riceve comandi mentre è in sleep mode risponderà con il codice del suo stato interno) al quale il sensore risponderà con 4 byte:

1. **byte 1**: contiene l'indirizzo associato al sensore;
2. **byte 2**: contiene il comando al quale il sensore sta rispondendo;
3. **byte 3**: contiene informazioni riguardo lo stato del sensore;
4. **bytes 4**: checksum.

Il byte significativo è il terzo ed ogni suo bit corrisponde ad un errore diverso secondo la figura 2.28:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Laser Error	Memory Error	Fan Error	T/RH Error	Heat Error	Not Ready	Degraded State	Sleep State

Figura 2.28: Tera NEXT-PM stato

2.3.4 Omron B5W-LD0101



Figura 2.29: Omron B5W-LD0101

Questo trasduttore fornisce due uscite di tipo PWM dal quale è possibile risalire al valore di concentrazione di particolato tramite l'utilizzo di un metodo sottrattivo che verrà descritto in seguito.

L'alimentazione supportata varia in un range di 4.5-5.5 V DC con un consumo di corrente massimo dichiarato di 90 mA.

I produttori forniscono un grafico di correlazione tra i dati ricavati dal sensore e quelli ottenuti da un Dust Monitor (TSI DUSTTRAK MODEL8530) circa i valori di particolato generati dal fumo di incenso, oltre che la configurazione utilizzata per il rilevamento (2.30):

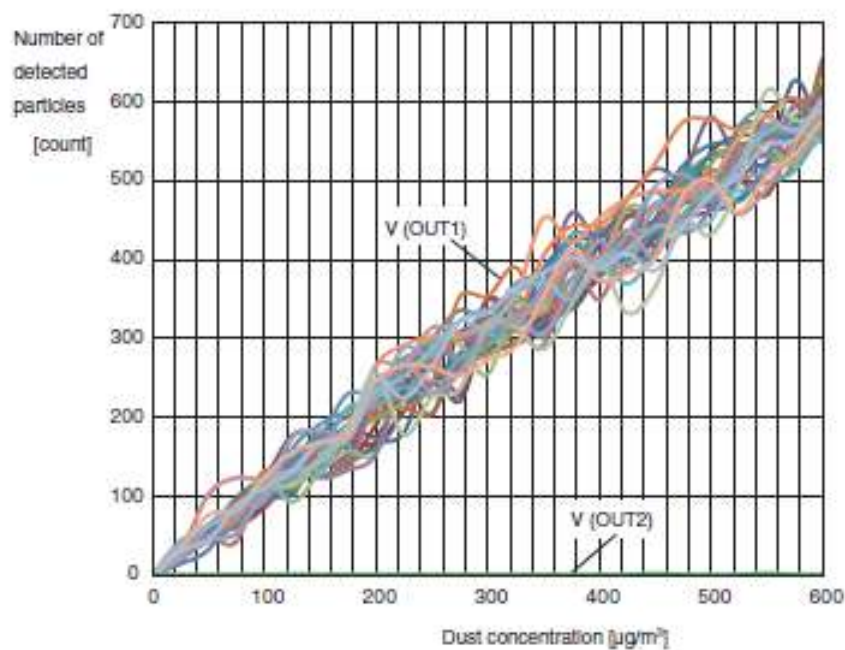


Figura 2.30: Omron B5W-LD0101 incense

Conversione dei dati

L'Omron B5W-LD0101 è predisposto a ricevere, tramite uno dei suoi 5 pin, una tensione detta di threshold attraverso il quale si può variare la sensibilità del trasduttore a particelle di dimensioni diverse.

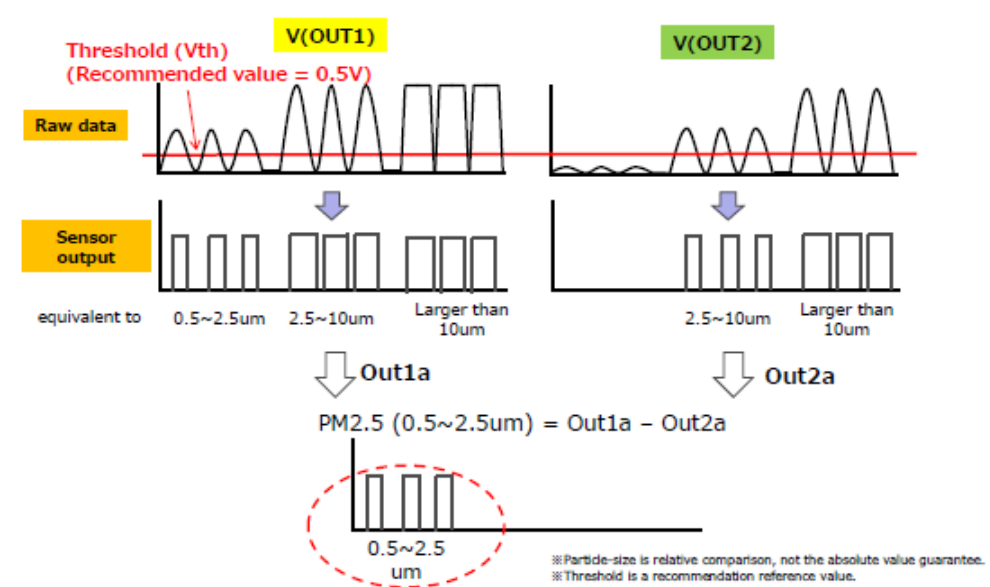


Figura 2.31: Omron B5W-LD0101 PM_{2.5} conversion

Come possiamo notare dalla figura 2.31, i circuiti interni che generano le due uscite del trasduttore presentano sensibilità diverse alle dimensioni delle particelle e l'uscita varia a seconda della tensione di threshold fornita al trasduttore. Tramite la sottrazione dei conteggi ottenuti è possibile ottenere il valore di concentrazione di particolato.

Facendo variare il valore della tensione di threshold (0.5 V per le PM_{2.5} e 3 V per le PM₁₀) è possibile ottenere una stima dei valori di PM₁₀ come mostrato in figura 2.32.

Si tratta di una stima di questo valore in quanto le misurazioni vanno effettuate in due momenti diversi perchè bisogna prima conteggiare l'uscita V1 con una tensione di threshold di 0.5 V per poi sottrarre il conteggio ottenuto da V2 con una tensione di threshold di 3 V.

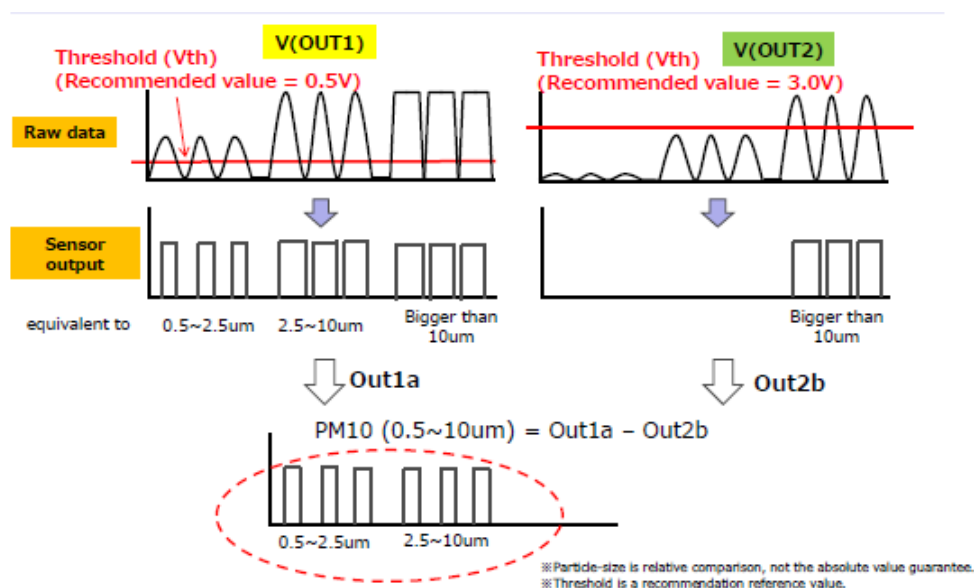


Figura 2.32: Omron B5W-LD0101 PM₁₀ conversion

2.4 Confronto fra sensori e scelta sensore da implementare in DoNet

Una volta effettuata la caratterizzazione dei sensori selezionati per il rilevamento delle concentrazioni di particolato ambientale, si necessita, al fine di selezionare il più adatto per l'implementazione all'interno di DoNet, della realizzazione di un box in compensato che permettesse la raccolta dei dati in maniera uniforme.

In questo paragrafo verrà discusso, inoltre, il setup realizzato per effettuare le misurazioni contenente il microcontrollore e i sensori selezionati.

2.4.1 Sistema di raccolta dati

Per quanto riguarda il sistema atto alla raccolta ed all'invio dei dati, possiamo vedere lo schema dei collegamenti in figura 2.35. Per i sensori Omron B5W-LD0101 e SMUART-04L sono state prese alcune precauzioni nei collegamenti.

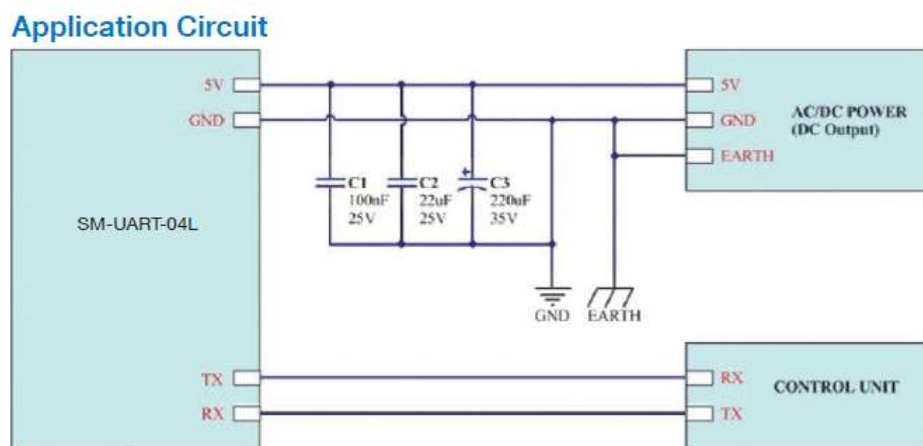


Figura 2.33: SM-UART-04L alimentazione

Come mostrato in figura 2.33, il sensore necessita di 3 condensatori in parallelo per garantire una stabilità della tensione di alimentazione.

Oltre a questo, lo stesso sensore presenta un pin di reset che per essere azionato (per riavviare il sensore) deve essere ad un livello logico basso. Per questo è stato inserito un tasto con una resistenza di pullup che mantiene alto il livello logico del pin in questione fino a che il pulsante non viene premuto, portando il livello logico del pin a 0V.

Per quanto riguarda il sensore Omron B5WLD0101 l'uscita PWM che fornisce presenta segnali da 5V. Il microcontrollore ESP32 supporta ingressi di massimo 3.6V continui per cui, onde evitare di danneggiarlo, si è optato di utilizzare un convertitore logico come quello in figura 2.34:



Figura 2.34: Convertitore logico bidirezionale

Per questa applicazione non si necessitava di un convertitore bidirezionale in quanto bisogna abbassare unicamente il livello logico dell'uscita del sensore, si è preferito utilizzare questa tipologia unicamente perché già presente in azienda. Non si hanno problemi per quanto riguarda le uscite degli altri 2 sensori essendo entrambe UART TTL a 3.3V.

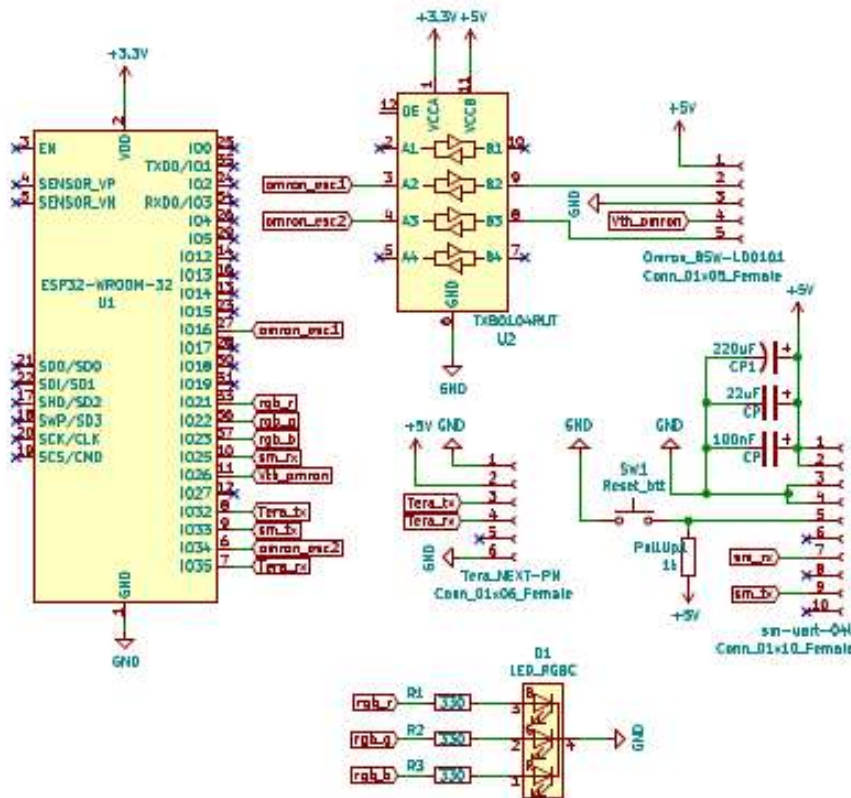


Figura 2.35: Schema dei collegamenti

Inoltre, come possiamo vedere dallo schematico in figura 2.35, è stato aggiunto un led RGB per la visualizzazione dello stato di funzionamento del sistema. Di seguito vengono descritti i codici colore utilizzati:

1. **Luce rossa fissa:** inizializzazione delle variabili / sistema attualmente non connesso alla rete;
2. **Luce blu lampeggiante:** connessione alla rete ed al broker MQTT;

3. **Luce verde, lampeggio rapido:** connessione eseguita con successo, seguita da luce spenta, inizializzazione dei sensori;
4. **Luce verde fissa:** sistema in funzione, acquisizione dati;
5. **Luce viola, lampeggio rapido:** invio dati al topic "esp32/data" riuscito;
6. **Luce rossa, lampeggio rapido:** invio dati fallito, dati corrotti.

Nella prossima figura (2.36) possiamo vedere due fotografie del device realizzato per la raccolta e invio dei dati:

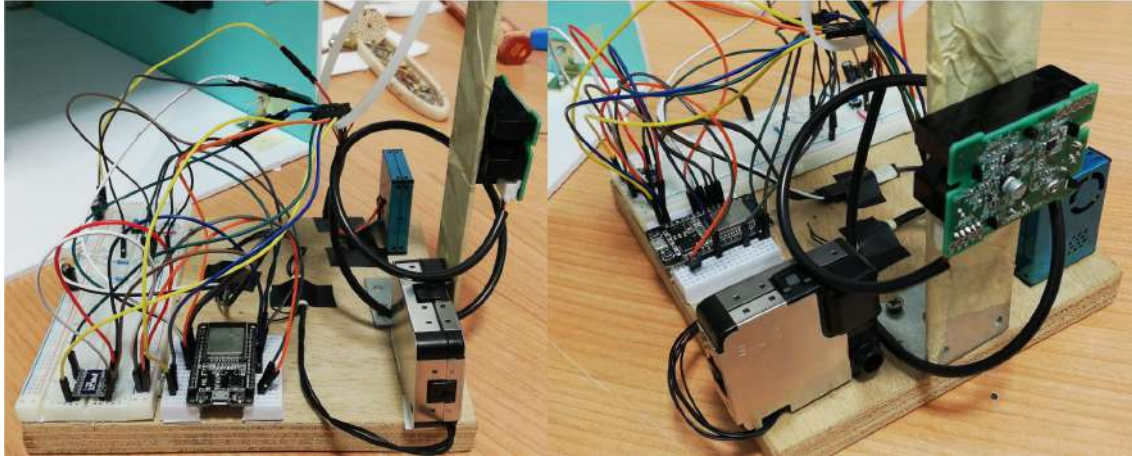


Figura 2.36: Device realizzato

2.4.2 Raccolta dati

Una volta realizzato il device necessario per la raccolta dei dati, al fine di individuare quale sia il sensore migliore da implementare in DoNet, facendo riferimento alle caratteristiche elencate all'interno della sezione 2.2.1, sono stati realizzati i seguenti grafici:

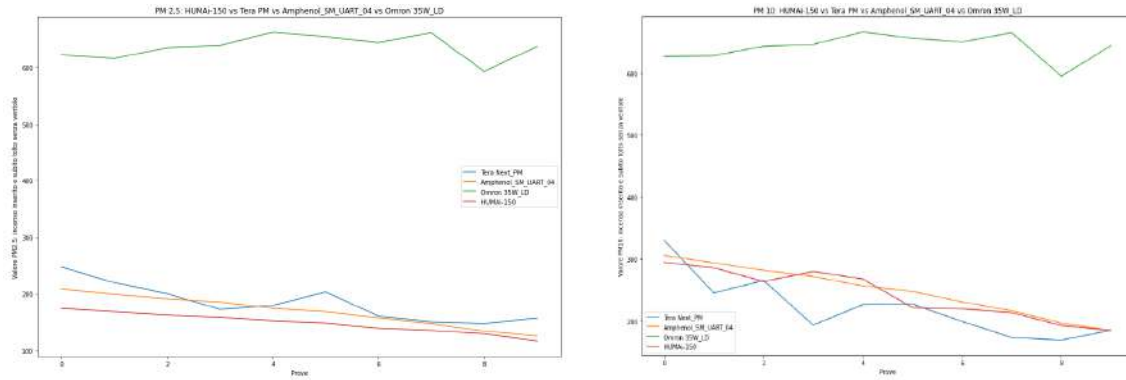


Figura 2.37: Misurazioni effettuate con incenso introdotto all'interno dell'ambiente e subito rimosso, senza però utilizzare la ventilazione artificiale

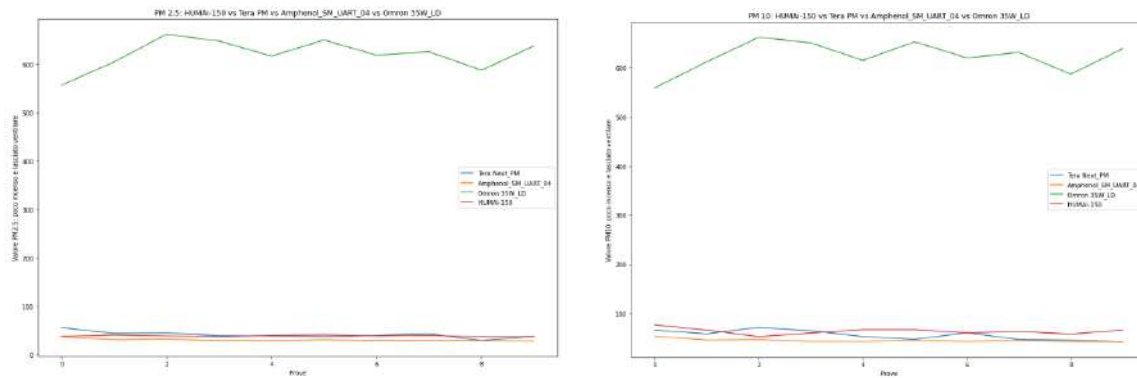


Figura 2.38: Misurazioni effettuate con incenso introdotto all'interno dell'ambiente e subito rimosso, utilizzando però in questo caso la ventilazione artificiale

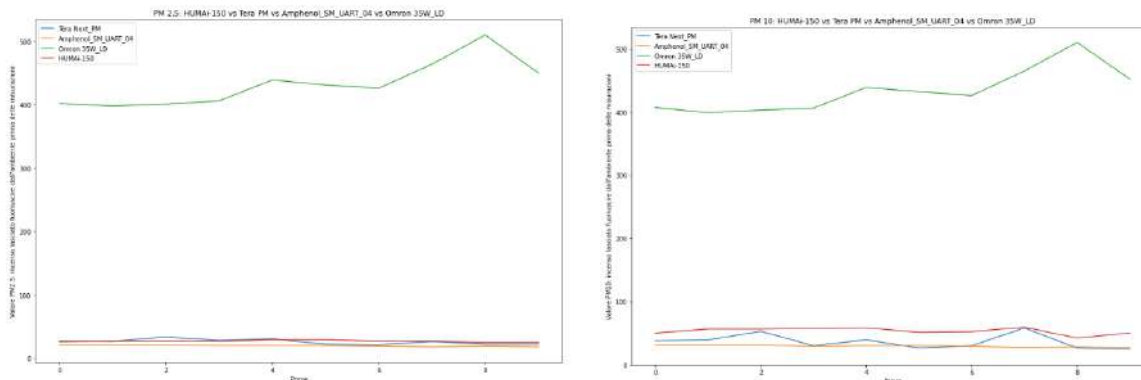


Figura 2.39: Misurazioni effettuate con incenso lasciato fuoriuscire dall'ambiente prima del rilevamento

2.4.3 Scelta del sensore

Prima di procedere con la scelta del sensore da implementare all'interno di DoNet, verrà riportata una tabella (2.40) contenente le principali caratteristiche, i pro e i contro di ogni sensore, da prendere in considerazione per la scelta di uno dei quattro dispositivi proposti all'interno della sezione 2.3:

Tipologia di sensore	Costo	Range	Pro	Contro	Limite di saturazione
Tera_Next:	Circa 95 euro	particelle del diametro compreso tra: $0.3 \mu m \leq d \leq 10 \mu m$	-Il più affidabile tra quelli descritti - per la comunicazione dei dati è predisposto all'utilizzo del protocollo <u>ModBus</u> oltre all'UART - il protocollo ModBus lo rende che lo rende particolarmente adatto per alcune applicazioni in ambito industriale - Fornisce anche temperatura e umidità	- Molto Costoso - Più dispendioso in termini energetici: consuma in working mode dai 80 mA ai 300mA.	Non specificato
Amphenol (SM_uart):	Circa 20 euro	particelle del diametro compreso tra: $0.3 \mu m \leq d \leq 10 \mu m$	- Alta accuratezza - Elevata sensibilità - Compatto - Montatura flessibile - Poco dispendioso in termini energetici 60mA - Forte linearità tra i dati per cui il sensore risulta essere affidabile	- Tecnologia laser con generazione del flusso d'aria da campionare tramite ventola (peggiore rispetto a omron). -Non è predisposto per l'utilizzo del protocollo ModBus	999 $\mu g/m^3$
Omron 35W_LD:	Circa 11 euro	particelle del diametro maggiore di: $d \geq 0.5 \mu m$	-Utilizza un laser ad infrarossi che rende più precise le misurazioni, - Alta sensibilità ($100 \mu g/m^3$) - Dispone di una struttura unica del percorso del flusso che aspira efficacemente l'aria - Di piccole dimensioni grazie al suo sistema ottico compatto	- Non presenta limite superiore -Presenta un limite inferiore più alto perdendo quindi la possibilità di rilevare le particelle più piccole -Mediamente dispendioso in termini energetici: 90mA	Non specificato
Humai-150:	Circa 125 euro	Non specificato	-Stessa precisione Tera -Compatto e leggero -Portatile e ricaricabile -Capace di misurare anche VOC e CO ₂	-Range davvero limitato -Non presenta possibilità di esportare dati -Misurazioni molto sensibili -Presenta numerose vibrazioni causate dalla ventola	500 $\mu g/m^3$

Figura 2.40: Tabella caratteristiche sensori

Dopo aver analizzato i grafici proposti all'interno della sezione 2.4.2 e la precedente tabella, per l'implementazione in DoNet è stato quindi scelto l'**SM-UART-04L**. Questo perché, confrontando i pro e i contro precedentemente proposti e i risultati derivanti dai grafici, esso risulta essere il sensore più adatto per un applicazione di tipo commerciale dato che:

1. Fornisce misure stabili e accurate (come si può evincere dai grafici);
2. Possiede un'elevata sensibilità;
3. Compatto;
4. Basso costo;
5. Facilmente implementabile;
6. Rappresenta un punto d'incontro fra il Tera (riferimento) e l'Omron35W_LD (come si può evincere dai grafici).

2.5 Implementazione SM-UART-04L su DoNet

Scelto il sensore, non resta che implementare sia dal punto di vista hardware che dal punto di vista software tale dispositivo all'interno di DoNet.

Il DoNet 2.41 progettato e sviluppato da IDEA Soc. Coop. è **coperto da brevetto**.

2.5.1 Implementazione Hardware

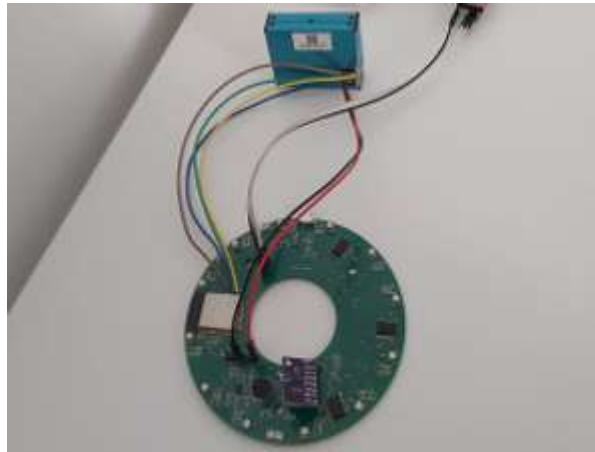


Figura 2.41: Implementazione hardware SM-UART-04L su DoNet

La connessione fra DONET e SM-UART-04l è stata effettuata saldando al modulo ESP-WROOM-32 i seguenti cavi Arduino (2.41):

1. **Cavo bianco:** TX connessione TTL;
2. **Cavo grigio:** RX connessione TTL;
3. **Cavi neri:** GND ESP32 e GND SM-UART-04L;
4. **Cavo rosso:** alimentazione SM-UART-04L.

I cavi saldati direttamente al modulo ESP-WROOM-32 sono:

1. **Cavo marrone:** Sleep SM-UART-04L;
2. **Cavo blu:** Reset SM-UART-04L;
3. **Cavo verde:** UART2 Tx SM-UART-04L;
4. **Cavo giallo:** UART2 Rx SM-UART-04L.

2.5.2 Implementazione Software

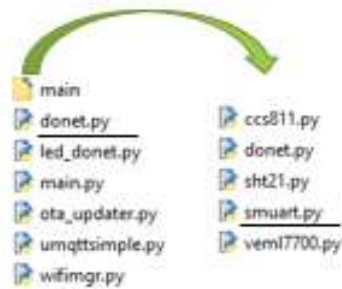


Figura 2.42: Implementazione software SM-UART-04L su DoNet

Dal punto di vista software, invece, oltre ad aver aggiunto nella cartella main la libreria (*smtuart.py* 2.5.2.1) che garantisce il corretto funzionamento del sensore SM-UART-04L, il file che è stato modificato all'interno del firmware DoNet è: *donet.py*.

Dopo aver aggiornato il firmware, è stata creata su Github una nuova versione OTA, la V4.0 che potrà essere scaricabile direttamente dal DONET grazie al file *ota_updater.py* seguendo una specifica procedura.

Praticamente cosa succede?

A metà dell'intervallo di misurazione selezionato dall'utente, l'SM-UART-04L andrà ad accendersi in modo tale che poi alla fine di tale intervallo, i dati forniti dal DoNet siano corretti. Una volta finite le acquisizioni, cioè al termine di ogni ciclo di misurazione (2.43), L'SM-UART-04L viene spento, per poi essere riacceso a metà periodo, sfruttando le sue funzionalità di reset e sleep, in modo tale da evitare inutili sprechi energetici.

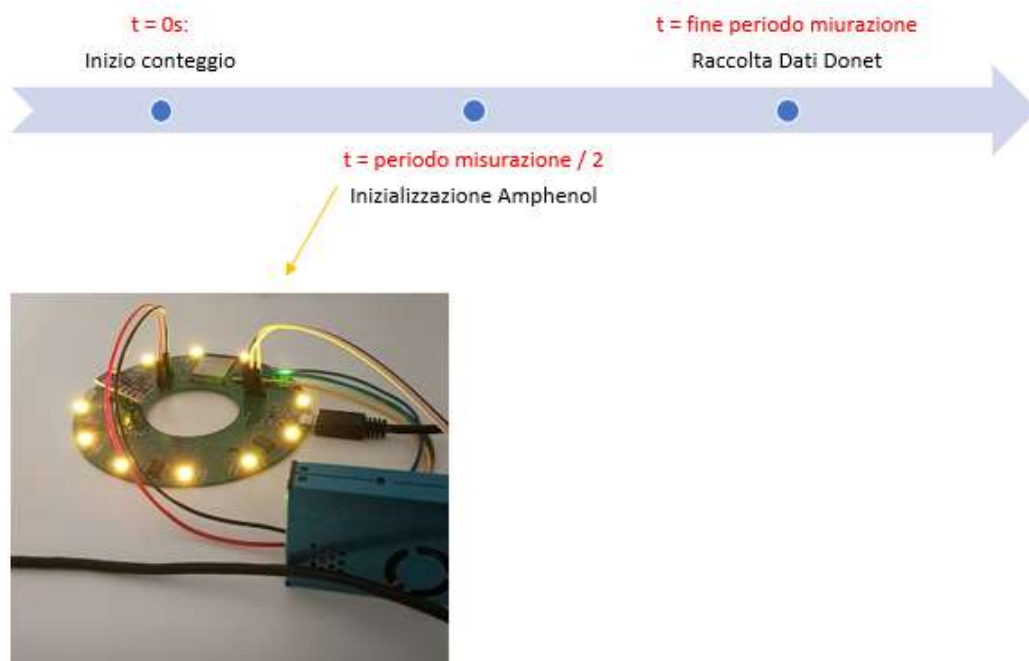


Figura 2.43: Ciclo di funzionamento DoNet con SM-UART-04L

2.5.2.1 Listati

```
1
2 from machine import UART
3 from time import sleep
4 from machine import Pin
5 from utime import sleep_ms
6
7 class SmUart04l:
8     def __init__(self, tx, rx, n_device):
9
10         self.__uart = UART(n_device, baudrate = 9600, bits = 8, parity = None,
11 stop = 1, tx = tx, rx = rx)
12         self.__pm_value = [0, 0, 0] # PM 1, 2.5, 10
13
14         header = 0x42 + 0x4d
15         crc_get_reading = header + 0xe2
16         crc_answer_mode = header + 0xe1
17         crc_direct_mode = header + 0xe1 + 0x01
18         crc_standby = header + 0xe4
19         crc_work = header + 0xe4 + 0x01
20
21         self.get_reading = bytearray([0x42, 0x4d, 0xe2, 0x00, 0x00,
22 crc_get_reading>>8, crc_get_reading & 0xff])
23         self.answer_mode = bytearray([0x42, 0x4d, 0xe1, 0x00, 0x00,
24 crc_answer_mode>>8, crc_answer_mode & 0xff])
25         self.direct_mode = bytearray([0x42, 0x4d, 0xe1, 0x00, 0x01,
26 crc_direct_mode>>8, crc_direct_mode & 0xff])
27         self.standby_mode = bytearray([0x42, 0x4d, 0xe4, 0x00, 0x00,
28 crc_standby>>8, crc_standby & 0xff])
29         self.working_mode = bytearray([0x42, 0x4d, 0xe4, 0x00, 0x01,
30 crc_work>>8, crc_work & 0xff])
31
32         sleep(2)
33         self.__uart.write(self.standby_mode)
34         self.status = 'sleeping'
35         sleep(1)
36         self.__uart.read()
37         print("SMUART04L set-up done")
38     def start(self):
39         reset = Pin(18, Pin.OUT)
40         reset.value(1)
41         self.__uart.read()
42         if self.status == 'running':
43             print('SM-UART-04L: Also running')
44         else:
45             self.__uart.write(self.working_mode)
46             while self.__uart.any() == 0:
47                 pass
48             self.status = 'running'
49             self.__uart.read()
50             self.__uart.write(self.answer_mode)
51
52     def stop(self):
53         self.__uart.read()
54         if self.status == 'sleeping':
55             print('SM-UART-04L: Also sleeping')
56         else:
57             self.__uart.write(self.standby_mode)
58             while self.__uart.any() == 0:
```

```

53     pass
54     self.status = 'sleeping'
55     self.__uart.read()
56     reset = Pin(18, Pin.OUT)
57     reset.value(0)
58 def get_pm(self):
59     if self.status != 'running':
60         print('SM-UART-04L: Not running')
61     else:
62         self.__uart.read()
63         self.__uart.write(self.get_reading)
64         while self.__uart.any() < 32:
65             pass # busy waiting
66         datas = bytearray(32)
67         self.__uart.readinto(datas)
68         crc = (datas[30]<<8)+datas[31]
69         sumv = sum(datas[:-2])
70         if crc != sumv:
71             return False # crc error
72         else:
73             self.__pm_value[0] = (datas[4]<<8)+datas[5]
74             self.__pm_value[1] = (datas[6]<<8)+datas[7]
75             self.__pm_value[2] = (datas[8]<<8)+datas[9]
76             return True
77
78 def measure(self):
79     if self.get_pm() == True:
80         print(self.__pm_value)
81         return self.__pm_value
82     else:
83         print('SM-UART-04L: something gone wrong measuring')
84         return None

```

Elenco dei listati 1: smuart.py

Di seguito, invece, le parti modificate all'interno dello script donet.py:

```

1 self.__smuart = smuart.SmUart04l(tx = sm_tx, rx = sm_rx, n_device = 2)
2 self.__dati = []

```

Elenco dei listati 2: prima parte introdotta da linea 83 a linea 84 donet.py

```

1 values = OrderedDict([('mac.address', '{}'.format(json_list[0])),
2                       ('temperature.celsius', '{}'.format(str(json_list[1]))),
3                       ('relative.humidity', '{}'.format(str(json_list[2]))),
4                       ('light.lux', '{}'.format(str(json_list[3]))),
5                       ('co2.ppm', '{}'.format(str(json_list[4]))),
6                       ('tvc.ppb', '{}'.format(str(json_list[5]))),
7                       ('Sound', '{}'.format(str(json_list[6]))),
8                       ('PM2_5', '{}'.format(str(json_list[7]))),
9                       ('PM10', '{}'.format(str(json_list[8])))]

```

Elenco dei listati 3: modifica valori da inviare a database in donet.py

```

1 PM = self.__smuart.measure()
2 self.__dati.append(self.__smuart.measure())
3 sleep_ms(2000)
4 self.__smuart.stop()
5 data = [self.g_macAddress, temp, hum, lux, co2, tvoc, 0 ,PM[1], PM[2]]

```

Elenco dei listati 4: terza modifica effettuata da linea 345 a linea 349 in donet.py

2.6 Risultati Ottenuti

Usufruendo del servizio hosting Netsons, dopo aver modificato adeguatamente la struttura del vecchio database, sono state effettuate alcune prove per testare il reale funzionamento del sensore per il particolato ambientale all'interno di DoNet.











				Time_stamp	ID_Measure	MAC_Address	Temperature	Humidity	Luminosity	CO2	Particulate	Sound	PM2_5	PM10			
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:50:36	2907586	4c:11:ae:fd:f3:4c	23.9	53.2	59	575	26	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:49:30	2907585	4c:11:ae:fd:f3:4c	23.9	53.2	103.6	596	29	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:48:24	2907584	4c:11:ae:fd:f3:4c	23.9	53.9	98.3	706	46	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:47:19	2907583	4c:11:ae:fd:f3:4c	23.8	54.2	81.8	581	27	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:46:13	2907582	4c:11:ae:fd:f3:4c	23.8	54.1	79.3	666	40	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:45:08	2907581	4c:11:ae:fd:f3:4c	23.8	54.3	80.5	672	41	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:44:03	2907580	4c:11:ae:fd:f3:4c	23.7	54.1	87.2	640	36	0	6	9
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:35:21	2907579	4c:11:ae:fd:f3:4c	23.8	54.1	95.3	564	24	0	6	10
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:33:11	2907578	4c:11:ae:fd:f3:4c	23.7	54	106.3	581	27	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:15:17	2907601	4c:11:ae:fd:f3:4c	23.9	52	74.7	656	39	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:14:12	2907600	4c:11:ae:fd:f3:4c	23.9	51.9	77.8	626	34	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:13:04	2907599	4c:11:ae:fd:f3:4c	23.9	51.6	67.7	629	34	0	4	7
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:11:58	2907598	4c:11:ae:fd:f3:4c	23.9	51.6	76.7	666	40	0	5	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:10:52	2907597	4c:11:ae:fd:f3:4c	23.9	51.7	82.5	665	40	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:09:47	2907596	4c:11:ae:fd:f3:4c	23.9	51.8	83.5	640	36	0	5	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:08:42	2907595	4c:11:ae:fd:f3:4c	23.9	52	83.6	600	30	0	6	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:07:36	2907594	4c:11:ae:fd:f3:4c	23.9	52	75.7	591	29	0	5	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:06:31	2907593	4c:11:ae:fd:f3:4c	23.9	52.2	73.2	596	29	0	5	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:05:26	2907592	4c:11:ae:fd:f3:4c	23.9	52.1	78.4	584	28	0	5	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:04:20	2907591	4c:11:ae:fd:f3:4c	24	52	78.9	574	26	0	5	8
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:03:15	2907590	4c:11:ae:fd:f3:4c	23.9	52	84.3	537	20	0	5	7
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 15:02:10	2907589	4c:11:ae:fd:f3:4c	23.8	52.3	86.4	459	8	0	5	7
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:52:45	2907588	4c:11:ae:fd:f3:4c	23.9	53.3	87.1	575	26	0	6	9
<input type="checkbox"/>		Modifica		Copia		Elimina	2021-06-11 12:51:40	2907587	4c:11:ae:fd:f3:4c	23.9	53.2	102	586	28	0	6	10

Figura 2.44: Risultati ottenuti tramite database Netsons

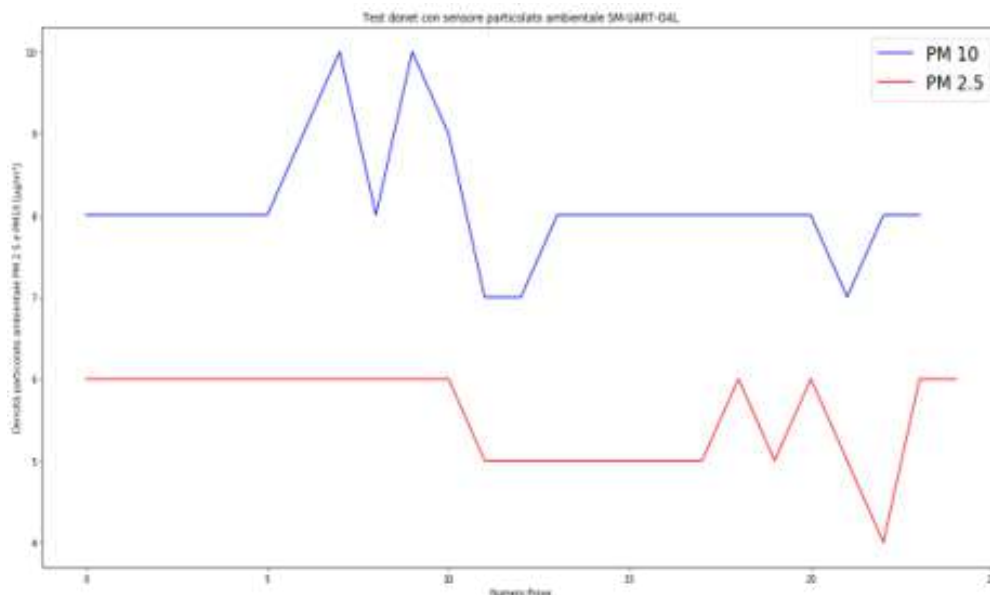


Figura 2.45: Risultati ottenuti PM_{2.5} e PM₁₀ con nuova struttura DoNet

Come si evince dalle precedenti immagini, oltre ai soliti sensori presenti all'interno di DoNet per la misura di: temperatura, umidità, luminosità, CO₂, TVOC, ora è possibile valutare la densità di particolato ambientale presente nell'aria attraverso DoNet.

3 Implementazione embedded di un sensore per la misura della densità di particolato ambientale $PM_{2.5}$ e PM_{10}

Dopo aver studiato ed implementato all'interno di DoNet un sensore per il particolato ambientale già in commercio, si è deciso, all'interno del lavoro di tirocinio, di progettare, creare e implementare da zero un sensore per le polveri sottili $PM_{2.5}$ e PM_{10} .

Questo progetto nasce dall'esigenza di creare un sensore per il particolato ambientale che abbia le seguenti caratteristiche:

1. Capacità di fornire misure accurate e precise;
2. Basso costo;
3. Dimensioni ridotte;
4. Facilmente implementabile all'interno di un device (in questo caso DoNet);
5. Consumi ridotti.

In linea di massima un sensore per il particolato ambientale è costituito da:

- Microcontrollore;
- Diodo laser;
- Fotodiodo;
- Convertitore analogico-digitale + amplificatore;
- Ventola di aspirazione.

L'aria aspirata passa attraverso la camera di rilevamento, dove la luce del laser che colpisce le particelle di polvere sospese, viene diffusa in tutte le direzioni e trasformata in segnali elettrici che saranno amplificati ed elaborati. In uscita sarà presente un segnale digitale proporzionale alla concentrazione di particolato rilevato.

Nella prossima figura (3.1) è proposta una possibile implementazione:

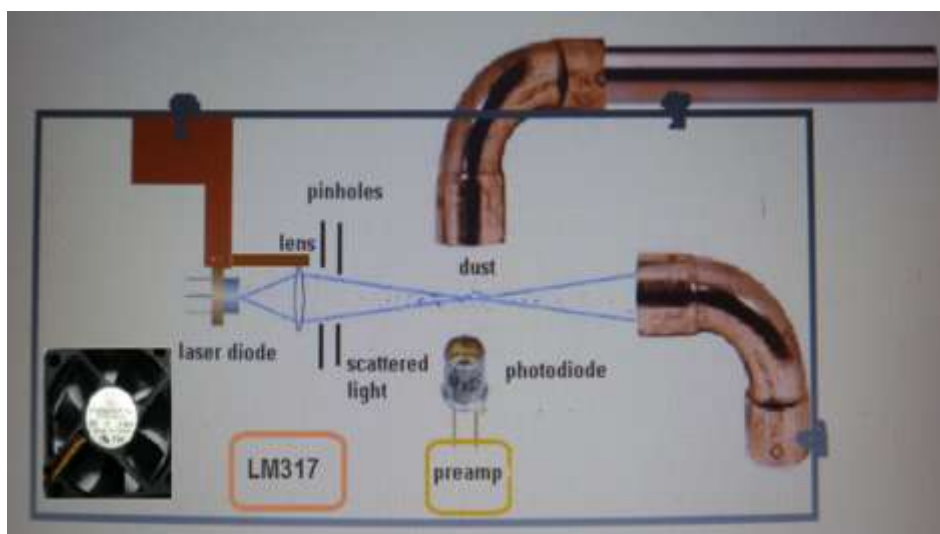


Figura 3.1: Possibile implementazione sensore $PM_{2.5}$ e PM_{10}

3.1 Tecnologie prese in considerazione per l'implementazione del sensore

La tecnologia scelta per l'implementazione del sensore è lo *scattering laser*. Però, grazie a numerosi incontri con alcuni docenti universitari, sono state esplorate anche altre strade prima di prendere la decisione finale. In particolar modo, le tecniche prese in considerazione sono:

- Citofluorimetria (citometria a flusso);
- Nanosight;
- Dynamic light scattering;
- Scattering di Mie;
- Scattering laser.

3.1.1 Citofluorimetria (citometria a flusso)

La CFM è una tecnica che consente la misurazione e caratterizzazione di cellule sospese in un mezzo fluido. Vediamone i principi base:

una sospensione cellulare monodispersa viene iniettata in un sistema fluidico il quale tende, in opportune condizioni idrodinamiche, a trasportare le cellule in maniera separata e ordinata fino al punto di misura, dove incontra il fascio di luce focalizzata proveniente dal laser. L'incontro tra il raggio di luce e ogni singola cellula presente nel flusso cellulare genera dei segnali. Questi segnali sono legati alle caratteristiche fisiche della cellula e alla presenza di molecole fluorescenti. I segnali sono raccolti da un sistema di lenti, specchi e filtri ottici, e inviati ai rispettivi sensori (fotodiodi e fotomoltiplicatori) che ne misurano l'intensità.

Andando più nel dettaglio, alcune limitazioni di questa tecnica sono:

- In questa tecnica può essere presa in considerazione *una* sola particella alla volta;
- Il laser utilizzato di solito è a 488 nm;
- Per avere risoluzioni migliori si deve passare ad un laser con lunghezza d'onda sul blu.

In generale, un citofluorimetro a flusso è costituito da:

1. Sistema fluidico per il trasporto del campione e la sua focalizzazione idrodinamica;
2. Sistema ottico costituito da sorgente di eccitazione e di rilevazione;
3. Sistema elettronico per l'acquisizione, elaborazione e la rappresentazione dei dati.

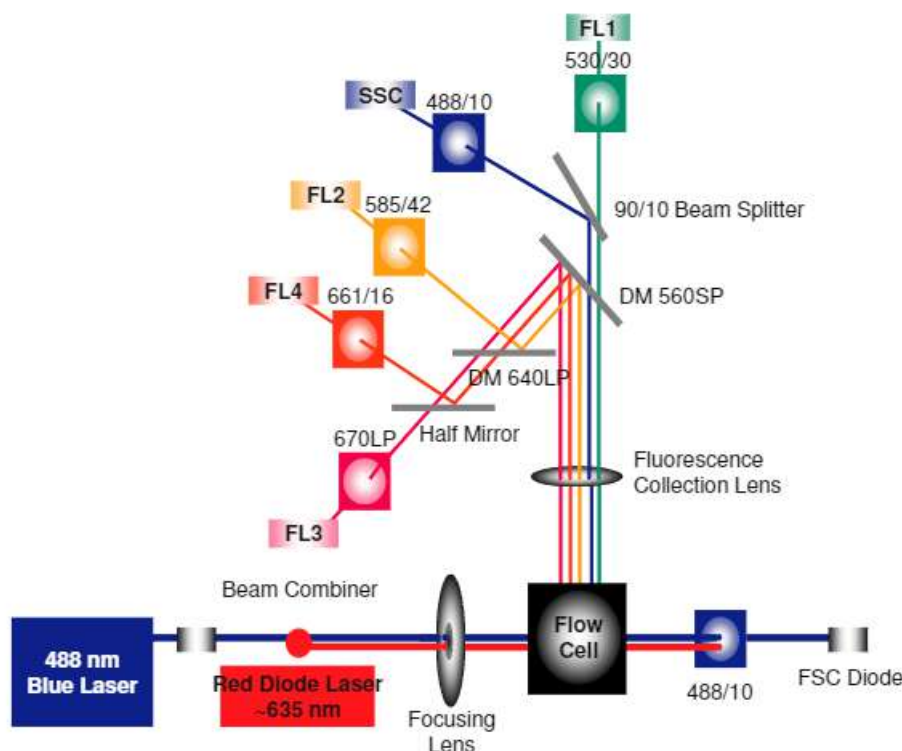


Figura 3.2: Possibile implementazione citometria a flusso

Sistema fluidico

Nel caso si voglia implementare un sensore per il particolato ambientale utilizzando la tecnica della citofluorimetria, il fluido preso in considerazione sarà l'aria e non un normale liquido, quindi, sarà necessario andare ad implementare un tubicino di 100/200 micron che ci permetta di incanalare l'aria in modo da non far passare un flusso enorme e in modo tale da far scorrere una sola particella alla volta.

Sistema elettronico

È il sistema elettronico a valle dell'insieme di filtri e specchi atti a selezionare le opportune lunghezze d'onda, sono posizionati dei sensori elettronici denominati fotomoltiplicatori (PMT) che trasformano i segnali ottici in intensità di corrente elettrica. I PMT svolgono inoltre la funzione di amplificare lo stesso segnale in maniera lineare o logaritmica. I segnali provenienti dai PMT sono impulsi analogici, ovvero segnali tempo continui e variabili nel tempo stesso e proporzionalmente al parametro misurato. Un convertitore analogico-digitale (ADC) converte il segnale analogico continuo in un valore discreto cioè "spezzetta" i valori continui rendendoli discreti a seconda del numero di canali (256,512,1024,ecc.) a disposizione.

Creazione di un impulso elettrico

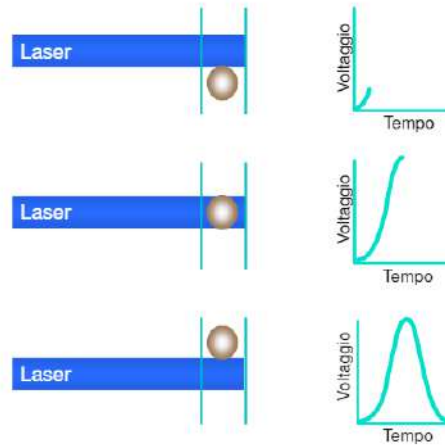


Figura 3.3: Impulso elettrico

Come quantifico poi tale impulso?

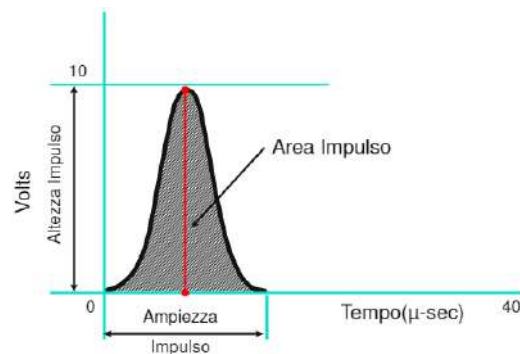


Figura 3.4: Quantizzazione impulso elettrico

Nel processo di conversione analogico-digitale, il parametro prescelto per descrivere discretamente l'impulso originale è l'*altezza* (H, height).

3.1.2 Nanosight

Tale tecnica è utilizzata per rilevare la dimensione e la densità di particelle con grandezza compresa fra 10 nanometri e 1 micrometro. Per questo motivo, in relazione all'applicazione precedentemente descritta non risulta particolarmente adatta.

Vediamone comunque le caratteristiche principali:

il flusso di particelle viene illuminato con un laser sul verde e, per l'acquisizione, si utilizza una telecamera con zoom 10x in modo tale da raccogliere le immagini davanti al raggio laser.

Di solito, viene utilizzato un filmato di 10 secondi che effettua il tracking delle particelle e ne valuta la velocità media.

Ottenuta tale grandezza, attraverso la velocità posso ricavare quella che è la dimensione delle particelle di particolato ambientale secondo la relazione per cui:

- Maggiore è la velocità della particella → minore è la dimensione della particella;
- Minore è la velocità della particella → maggiore è la dimensione della particella.

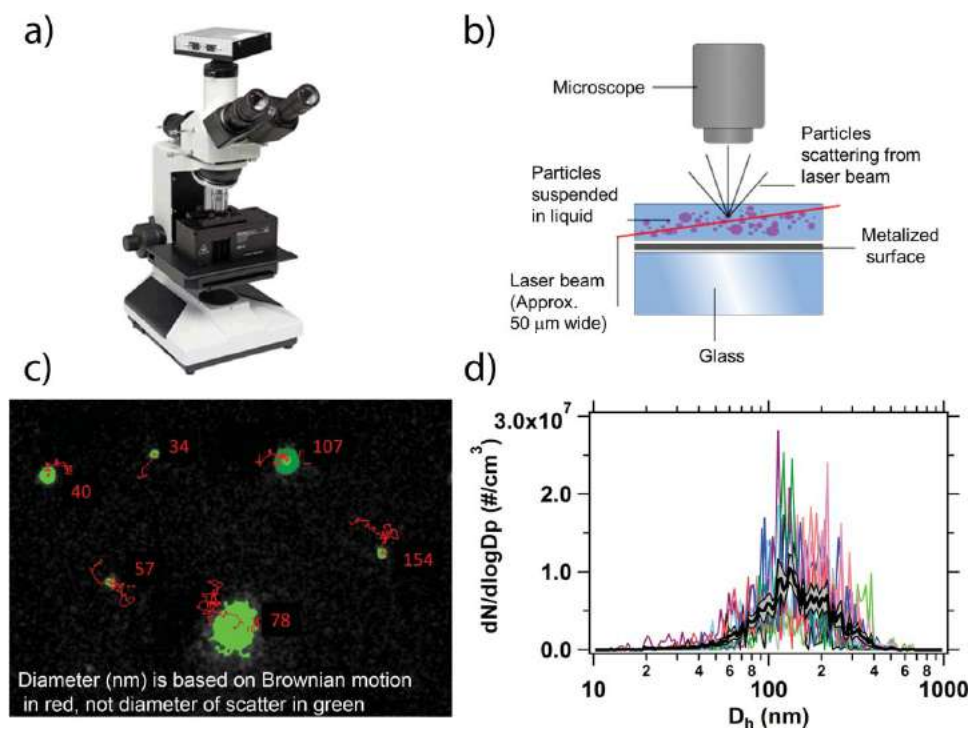


Figura 3.5: Tecnica nanosight

3.1.3 Dynamic light scattering

La diffusione dinamica della luce (DLS) è una tecnica che nella fisica può essere utilizzata per determinare il profilo della distribuzione di piccole particelle in sospensione o polimeri in soluzione. Può anche essere usata per sondare il comportamento di fluidi complessi come le soluzioni di polimero concentrato.

Quando la luce colpisce le piccole particelle si disperde in tutte le direzioni (dispersione di Rayleigh) purché le particelle siano piccole rispetto alla lunghezza d'onda (al di sotto dei 250 nm). Se la sorgente di luce è un laser, e perciò monocromatica e coerente, allora si osserva nell'intensità di dispersione (scattering) una fluttuazione dipendente dal tempo. Queste fluttuazioni sono dovute al fatto che le piccole molecole nelle soluzioni sono sottoposte al moto browniano e così la distanza tra gli spargimenti (scatterers) nella soluzione è in continua evoluzione nel tempo. Questa luce diffusa, viene sottoposta a un'interferenza o costruttiva o distruttiva da parte delle particelle circostanti e all'interno di questa fluttuazione d'intensità le informazioni sono contenute riguardo alla scala del tempo del movimento degli spargimenti.

Un importante osservazione da fare è che i sistemi DLS funzionano solo per particelle di dimensioni fino a 10 μm e sono sempre più imprecisi sopra 1 μm .

I componenti che costituiscono un sistema per la misurazione della diffusione dinamica della luce (DLS) sono riassunti nella prossima figura (3.6):

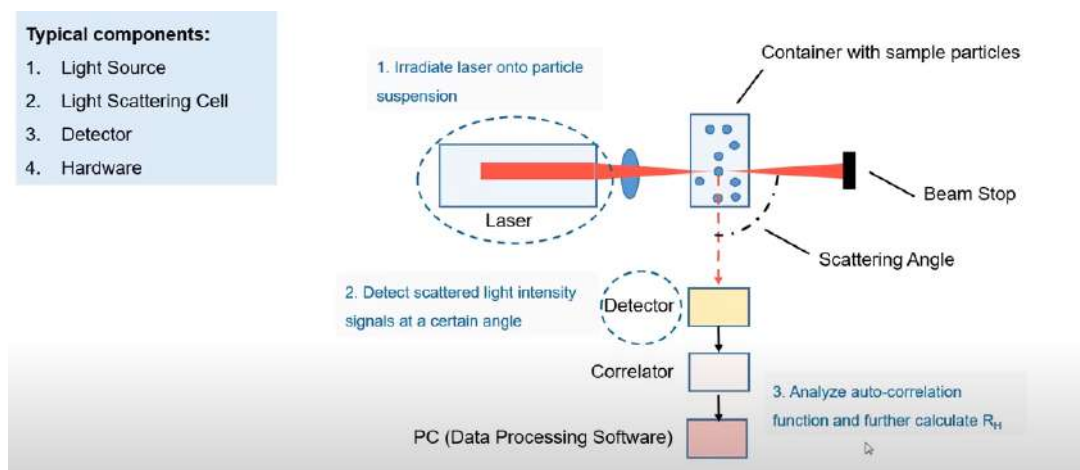


Figura 3.6: Dynamic light scattering

3.1.3.1 Step implementativi

1- Implementazione del sistema hardware come mostrato all'interno della figura 3.6

L'angolo fra il laser e il fotorilevatore (detection angle) deve essere sempre più grande di 20°, tipicamente a 90°!

2- Calcolo dell'intensità di luce diffusa attraverso il fotodetector

Da che cosa dipende l'intensità della luce?

$$I = f(q, t)$$

con

$$q = \frac{4\pi n}{\lambda \sin(\frac{\theta}{2})}, \text{dove :}$$

- n = indice di rifrazione medio;
- λ = lunghezza d'onda del raggio laser incidente al flusso;
- θ = angolo di rilevamento;
- q = vettore di scattering.

L'intensità I non avrà unità di misura, o meglio avrà un'unità di misura arbitraria in base a ciò che viene visualizzato dall'output del fotoricevitore, dato che:

- n è adimensionale;
- λ [nm];
- Il seno è adimensionale.

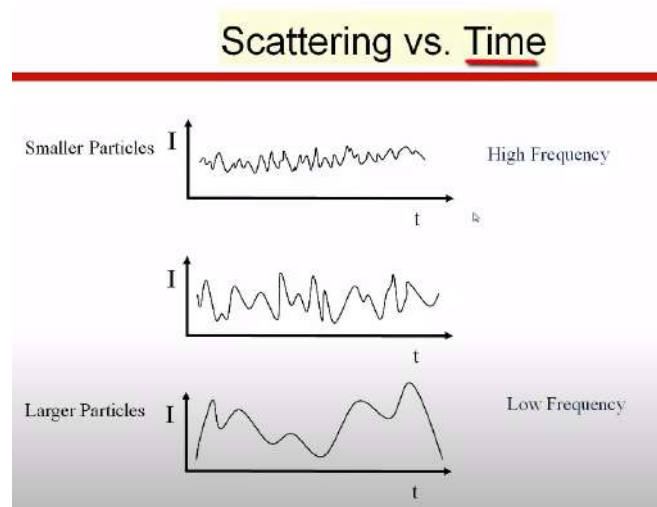


Figura 3.7: Scattering

3- Individuazione della tipologia di flusso da analizzare

- Flusso con particelle **monodisperse** = uniformi in grandezza, forma, peso;
- Flusso con particelle **polidisperse** = non uniformi in grandezza, forma, peso (caso preso in considerazione).

In base alla diversa tipologia cambia il modo di calcolare la funzione di correlazione.

4- Calcolo funzione di correlazione e coefficiente diffusione D

Un correlatore è fondamentalmente un comparatore di segnali che ci permette di calcolare:

$g_2(q, \tau)$ = funzione di autocorrelazione d'intensità del secondo ordine, con:

$$g_2(q, \tau) = \frac{I(q, t)I(q, t + \tau)}{\langle I(q, t)^2 \rangle}$$

dove $I(q, t)I(q, t + \tau)$ rappresenta il prodotto fra l'intensità scatterata di luce al tempo t e l'intensità di luce scatterata al tempo $t + \tau$, dove τ è il ritardo di correlazione che può essere scelto arbitrariamente fra i nanosecondi e i microsecondi dato che non influenza i risultati.

Mentre $\langle I(q, t)^2 \rangle$ rappresenta l'intensità media al quadrato calcolata sul periodo di campionamento.

Tale funzione è legata alla funzione di correlazione di primo ordine del campo elettrico ($g_1(q, \tau)$) attraverso la relazione di *Siebert*:

$$g_2(\tau) = 1 + b g_1(\tau)^2$$

Dove b è il fattore di coerenza dello strumento di misura che dipende da:

1. Dall'area del rivelatore;
2. Dall'allineamento ottico;
3. Dalle proprietà di diffusione del sistema.

$g_1(q, \tau)$ inoltre ha un decadimento che va da 1 a 0; dove 0 sta a significare che non c'è correlazione fra il segnale al tempo t e il segnale al tempo $t + \tau$, mentre 1 sta a significare che c'è perfetta correlazione.

Per i sistemi polidispersi, i quali saranno poi oggetto di studio, ogni particella contribuisce ponderata per la sua efficienza di dispersione e quindi:

$$g_1(\tau) = \int_0^\infty G(\Gamma) e^{-\Gamma \tau} d\Gamma$$

dove $\Gamma = q^2 D \tau$, con Γ che rappresenta il rate di decadimento e D identifica il coefficiente di diffusione.

Nel caso di particelle polidisperse, due principali algoritmi matematici sono utilizzati per risolvere tale integrale:

1. Il primo, e più comune, è il **metodo cumulativo**, che comporta la risoluzione dell'espansione di Taylor della funzione di autocorrelazione. Tuttavia, il metodo cumulativo è valido solo con campioni che hanno polidispersità di piccole dimensioni e con rate di decadimento monomodale;
2. Il secondo è l'algoritmo Contin. Tale metodo può calcolare direttamente la distribuzione del raggio idrodinamico per i campioni che sono ampiamente dispersi. Si tratta di un metodo matematico relativamente complicato che coinvolge la regolarizzazione.

Analizziamo, in linea di massima, il metodo *cumulativo*:

È uno dei metodi più semplici che permette di ricavare il tasso di decadimento $G(\Gamma)$. Tale metodo si basa su due relazioni:

1. Una tra $g_1(\tau)$ e la funzione generatrice di momenti delle distribuzioni;
2. L'altra tra il logaritmo della funzione $g_1(\tau)$ e la funzione generatrice di tale metodo.

Per quanto riguarda la prima relazione, esprimendo con $M(-\tau, \Gamma)$ la funzione generatrice possiamo scrivere che:

$$M(-\tau, \Gamma) = \int_0^\infty G(\Gamma) e^{-\Gamma \tau} d\Gamma = g_1(\tau)$$

l' m -esimo momento della distribuzione $m_m(\Gamma)$, invece, è dato dalla derivata m -esima di $M(-\tau, \Gamma)$ rispetto a τ

$$m_m(\Gamma) = \frac{d^m M(-\tau, \Gamma)}{d(-\tau)^m} \Big|_{-\tau=0} = \int_0^\infty G(\Gamma) \Gamma^m e^{-\Gamma \tau} d\Gamma \Big|_{-\tau=0}$$

per quanto riguarda la seconda relazione similmente possiamo scrivere che:

$$K(-\tau, \Gamma) = \ln[M(-\tau, \Gamma)] = \ln[g_1(\tau)]$$

dove l'm-esimo cumulante della distribuzione si calcola come:

$$K_m(\Gamma) = \frac{d^m K(-\tau, \Gamma)}{d(-\tau)^m} \Big|_{-\tau=0}$$

Sfruttando il fatto che i cumulanti, tranne per il primo, sono invarianti sotto un cambio di origine, si possono scrivere i cumulanti in termini di momenti intorno alla media come:

$$K_1(\Gamma) = \int_0^\infty G(\Gamma) \Gamma d\Gamma = \bar{\Gamma}$$

$$K_2(\Gamma) = \mu_2$$

$$K_3(\Gamma) = \mu_3$$

$$K_4(\Gamma) = \mu_4 - 3(\mu_2)^2$$

Dove μ_m rappresenta i momenti calcolati come:

$$\mu_m = \int_0^\infty G(\Gamma) (\Gamma - \bar{\Gamma})^m d\Gamma$$

Il primo cumulante descrive il tasso di decadimento medio della distribuzione. Il secondo e il terzo cumulante corrispondono direttamente ai momenti appropriati intorno alla media: il secondo momento corrisponde alla varianza, e il terzo fornisce una misura dell'asimmetria della distribuzione. I primi due cumulanti devono essere positivi, ma il terzo può essere positivo o negativo.

La base dell'espansione cumulativa che viene solitamente utilizzata nell'analisi dei dati nella DLS consiste nell'espansione del logaritmo di $g_1(\tau)$ in termini di cumulanti della distribuzione.

Questa relazione segue dal fatto che l'm-esimo cumulante è il coefficiente di $-\frac{\tau^m}{m!}$ nella espansione di Taylor di $K(-\tau, \Gamma)$ con $\tau = 0$, come dato da:

$$\ln[g_1(\tau)] = K(-\tau, \Gamma) = -\bar{\Gamma}\tau + \frac{k_2}{2!}\tau^2 - \frac{k_3}{3!}\tau^3 + \frac{k_4}{4!}\tau^4 + \dots$$

Per sfruttare al meglio questa forma sarà necessario usare i metodi lineari ai minimi quadrati per adattare questa funzione ai dati richiesti. Deve essere fatta quindi un'assunzione chiave sui dati:

si deve assumere che la baseline del grafico di correlazione sia esattamente uguale ad uno ($B=1$); pertanto, sostituendo a g_1 la relazione $g_2(\tau) = 1 + bg_1(\tau)^2$:

$$\ln[g_2(\tau) - 1] = K(-\tau, \Gamma) = \ln\left(\frac{\beta}{2}\right) - \bar{\Gamma}\tau + \frac{k_2}{2!}\tau^2 - \frac{k_3}{3!}\tau^3 + \dots$$

tale equazione è definita *funzione di adattamento*.

Anche se la maggior parte dei moderni correlatori fa un eccellente lavoro di misurazione della linea di base, piccole quantità di rumore possono portare a piccolissime deviazioni dall'unità ($B \neq 1$).

Le routine di adattamento non lineare permettono la possibilità di adattare i dati direttamente a $g_2(\tau)$ otteniamo quindi dall'equazione precedente:

$$g_2(\tau) = B + \beta e^{-2\bar{\Gamma}\tau + k_2\tau^2 - \frac{k_3}{3}\tau^3 + \dots}$$

usando tale forma è possibile calcolare la baseline B dai dati.

Tale equazione ha però dei problemi di stabilità per τ grandi. Si può anche esprimere la funzione di correlazione del campo in termini di momenti intorno alla media, riscrivendo $e^{(-\Gamma\tau)}$ in termini del suo valore medio:

$$e^{(-\Gamma\tau)} = e^{(-\bar{\Gamma}\tau)} e^{-(\Gamma-\bar{\Gamma})\tau}$$

sostituendo tale relazione all'interno di: $g_1(\tau) = \int_0^\infty G(\Gamma) e^{-\Gamma\tau} d\Gamma$ otteniamo:

$$g_1(\tau) = e^{-\bar{\Gamma}\tau} \int_0^\infty G(\Gamma) e^{-(\Gamma-\bar{\Gamma})\tau} d\Gamma$$

la funzione esponenziale all'interno dell'integrale può essere espressa con un'espansione in serie:

$$g_1(\tau) = e^{-\bar{\Gamma}\tau} \int_0^\infty G(\Gamma) \left[1 - (\Gamma - \bar{\Gamma})\tau + \frac{(\Gamma - \bar{\Gamma})^2}{2!} \tau^2 - \frac{(\Gamma - \bar{\Gamma})^3}{3!} \tau^3 + \dots \right] d\Gamma$$

questa espansione è esatta finché tutti i termini dell'espansione sono inclusi. Dopo l'integrazione, la funzione di correlazione può essere espressa in termini di valore medio:

$$g_1(\tau) = e^{-\bar{\Gamma}\tau} \left(1 + \frac{\mu_2}{2!} \tau^2 - \frac{\mu_3}{3!} \tau^3 + \dots \right)$$

da tale relazione, applicando la legge di Seigert:

$$g_2(\tau) = B + \beta e^{-2\bar{\Gamma}\tau} \left(1 + \frac{\mu_2}{2!} \tau^2 - \frac{\mu_3}{3!} \tau^3 + \dots \right)^2$$

tale forma permette l'adattamento diretto di B e ha il vantaggio di eliminare i problemi di stabilità per grandi τ emersi nel caso precedente.

Calcolate le funzioni di correlazione, conoscendole già attraverso il correlatore, possiamo andare a calcolare inversamente quello che è il coefficiente di diffusione D.

5- Calcolo raggio particolare ambientale.

Una volta ottenuta la misura del coefficiente di diffusione, il raggio idrodinamico può essere calcolato attraverso la legge di Stokes-Einstein:

$$R_H = \frac{K_b T}{6\pi\eta D} = \left[\frac{\text{Area}}{\text{Tempo}} \right]$$

dove:

- **D** = coefficiente di diffusione;
- η = viscosità del fluido;
- K_B = costante di Boltzman = $1,380649 \times 10^{-23}$ J/K;
- **T** = temperatura [K];
- R_H = raggio della particella da calcolare (Hydrodynamic Diameter) [μm].

3.1.3.2 Perché non utilizzare la DLS per la misura della densità del particolato ambientale

L'ultima tecnica esaminata pur essendo molto precisa, risulta essere poco adatta alla misura della densità del particolato ambientale.

Di seguito sono riportati alcuni motivi per cui tale tecnica non verrà utilizzata per l'implementazione del sensore embedded per il calcolo del particolato ambientale:

- Tale tecnica ci permette di ottenere le dimensioni della particella di particolato presa in esame e non la sua densità all'interno di una specifica area;
- i sistemi DLS funzionano solo per particelle di dimensioni non superiori ai 10 μm e sono sempre più imprecisi sopra 1 μm (il nostro caso);
- La DLS è adatta per misurare particelle dell'ordine dei nanometri;
- Fa uso di un correlatore;
- Onerosa dal punto di vista dei costi;
- Difficile da implementare in ambito portatile.

3.1.4 Scattering di Mie

Il principio su cui si base la misurazione della concentrazione di particolato ambientale con il metodo della diffusione della luce è il seguente:

1. la luce diffusa è prodotta quando le particelle di particolato sono irradiate dalla luce;
2. In determinate condizioni del sistema ottico e delle proprietà del particolato, l'intensità della luce diffusa è proporzionale alla concentrazione delle polveri sottili;
3. Misurando l'intensità della luce diffusa, la concentrazione di massa delle polveri può essere ottenuta.

La teoria dello scattering di Mie descrive la soluzione matematica rigorosa dell'equazione di Maxwell per particelle uniformi in mezzi uniformi che sono irradiati da un'onda monocromatica piana.

Secondo la teoria della diffusione di Mie, l'intensità della luce diffusa nel punto p dalla particella r può essere scritta come segue:

$$I_{\text{sca}} = I_0 g \frac{\lambda^2}{8\pi^2 r^2} gI(\theta, \Phi)$$

dove:

$$I(\theta, \Phi) = |S_1(\theta)|^2 \sin^2(\phi) + |S_2(\theta)|^2 \cos^2(\phi)$$

con:

- I_{sca} = intensità di luce scatterata;
- I_0 = intensità di luce incidente;
- λ = lunghezza d'onda del laser;
- θ = angolo di scatter;
- ϕ = angolo di polarizzazione della luce polarizzata;
- $S_1(\theta)$ e $S_2(\theta)$ = funzioni di ampiezza.

$$S_1(\theta) = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} [a_n \tau_n + b_n \pi_n]$$
$$S_2(\theta) = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} [a_n \tau_n - b_n \pi_n]$$

dove:

- a_n e b_n sono funzioni legate alle funzioni di Bessel;
- τ_n e π_n sono Polinomi di Legendre che sono soltanto collegati all'angolo di scattering θ .

a_n e b_n possono essere calcolati come segue:

$$a_n = \frac{\phi_n(a) \phi'_n(ma) - m \phi'_n(a) \phi_n(ma)}{\epsilon_n(a) \phi'_n(ma) - m \epsilon'_n(a) \phi_n(ma)}$$
$$b_n = \frac{m \phi_n(a) \phi'_n(ma) - \phi'_n(a) \phi_n(ma)}{m \epsilon_n(a) \phi'_n(ma) - \epsilon'_n(a) \phi_n(ma)}$$

dove i parametri mancanti rappresentano:

- $\phi_n(a)$ e $\epsilon_n(a)$ = funzioni di Bessel;
- a = diametro adimensionale;
- D = diametro della particella di particolato;
- m = indice di rifrazione della particella scatterata rispetto a quelle circostanti.

Secondo le equazioni mostrate finora, la distribuzione dell'intensità della luce diffusa dalla particella è legata:

1. alla lunghezza d'onda del laser emettitore;
2. all'indice di rifrazione;
3. all'angolo di diffusione;
4. alla dimensione della particella.

Per esaminare l'effetto dell'angolo di diffusione e delle dimensioni delle particelle sull'intensità della luce diffusa, sono state condotte simulazioni numeriche con Matlab ([7]) in combinazione con un'equazione di ricorsione e una frazione continua.

Sulla base degli studi esistenti, si può sapere che quando la lunghezza d'onda della luce è intorno a 632,8 nm, l'intensità della luce diffusa è più forte.

Nei dati (3.8) della simulazione sono stati utilizzati:

1. indice di rifrazione $m = 1,7 + 0,5i$;
2. lunghezza d'onda ottica = 633nm.

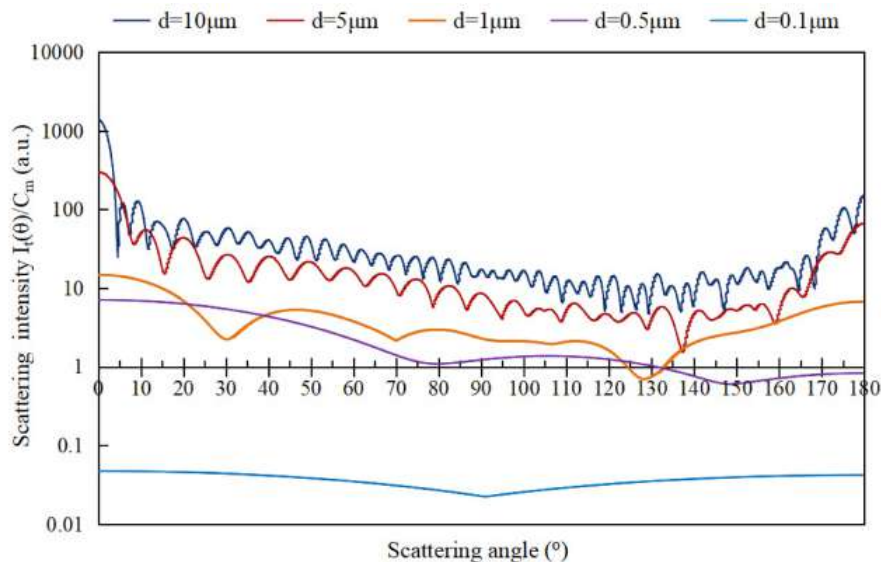


Figura 3.8: Relazione fra intensità di luce scatterata, dimensione della particella e angolo di scattering.

Tale grafico indica che l'intensità della luce diffusa è evidentemente aumentata con l'aumentare della dimensione delle particelle, e l'asimmetria della distribuzione dell'intensità della luce è cresciuta. La tendenza centrale della luce diffusa in avanti era apparentemente maggiore di quella della luce diffusa all'indietro, e sono state osservate evidenti caratteristiche oscillatorie.

Vediamo ora una possibile implementazione di tale tecnica

Il modello matematico è stato stabilito sulla base dei risultati della simulazione numerica e della relazione tra la dimensione delle particelle di polvere e l'intensità della luce diffusa. La relazione derivata successivamente tra il flusso luminoso e la concentrazione di polveri sottili fornisce una base importante per il calcolo dei dati del sensore. La prossima figura (3.9) presenta la diffusione in avanti di una singola particella di particolato:

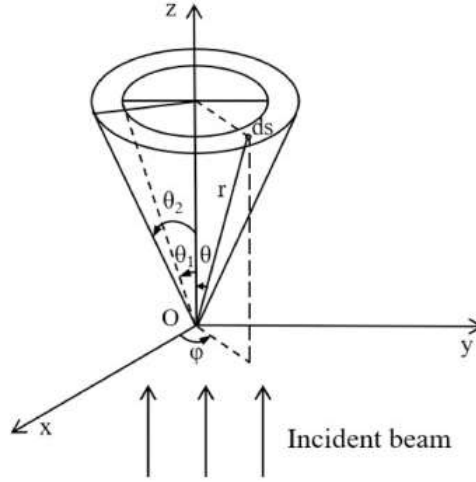


Figura 3.9: Grafico scattering in avanti di una singola particella di particolato

Il flusso luminoso in un angolo solido tra θ_1 e θ_2 può essere calcolato come segue:

$$F = \frac{\lambda^2 I_0}{4\pi} \int_{\theta_1}^{\theta_2} [i_1(\theta) + i_2(\theta)] \sin(\theta) d\theta$$

quando N singole particelle di particolato in un volume unitario di gas subiscono la diffusione Mie, il flusso luminoso può essere calcolato come segue:

$$F = N \frac{\lambda^2 I_0}{4\pi} \int_{\theta_1}^{\theta_2} [i_1(\theta) + i_2(\theta)] \sin(\theta) F_N(D) d\theta$$

dove:

- $i_1(\theta)$ e $i_2(\theta)$ sono le funzioni d'intensità polarizzata verticale e orizzontale della luce di diffusa;
- D denota il diametro del flusso luminoso della particella di particolato sferico;
- $F_N(D)$ rappresenta la funzione di distribuzione del numero delle particelle di polvere;

Per rilevare la concentrazione di massa di particolato, l'equazione presentata in precedenza deve essere convertita in un'equazione di calcolo del flusso luminoso e della massa di particolato. Quando V denota il volume dell'area fotosensibile e C denota la concentrazione di particolato, la massa della particella di particolato sferico equivalente può essere scritta come segue:

$$M(D) = \frac{\pi}{3} D^3$$

La relazione tra la funzione di distribuzione della massa e la funzione di distribuzione del numero di particelle può essere espressa come segue:

$$F_N(D) = \frac{\overline{M}}{M(D)} F_M(D)$$

con \overline{M} e $F_M(D)$ sono rispettivamente la massa media delle particelle e la funzione di distribuzione della massa delle polveri sottili.

Dunque, la relazione tra il numero di particelle di particolato N e la massa media di particolato è la seguente:

$$N = \frac{CV}{\overline{M}}$$

sostituendo queste due equazioni all'interno di $F = N \frac{\lambda^2 I_0}{4\pi} \int_{\theta_1}^{\theta_2} [i_1(\theta) + i_2(\theta)] \sin(\theta) F_N(D) d\theta$, otteniamo la relazione tra il flusso luminoso F e la concentrazione di particolato C :

$$\frac{F}{C} = V \frac{\lambda^2 I_0}{4\pi} \int_{\theta_1}^{\theta_2} [i_1(\theta) + i_2(\theta)] \sin(\theta) \frac{F_M(D)}{M(D)} d\theta$$

quando il sistema di percorso della luce e lo sciame di particelle di particolato sono dati, il flusso luminoso F è linearmente correlato con la concentrazione di particolato C e la tensione di uscita T del sensore. Inoltre, considerando che la concentrazione di particolato C calcolata si discosta leggermente dal valore reale, il coefficiente di proporzionalità K_0 è stato introdotto per determinare l'uscita effettiva del sensore:

$$C'' = K_0 C$$

Vediamone ora una delle tante possibili reali implementazioni:

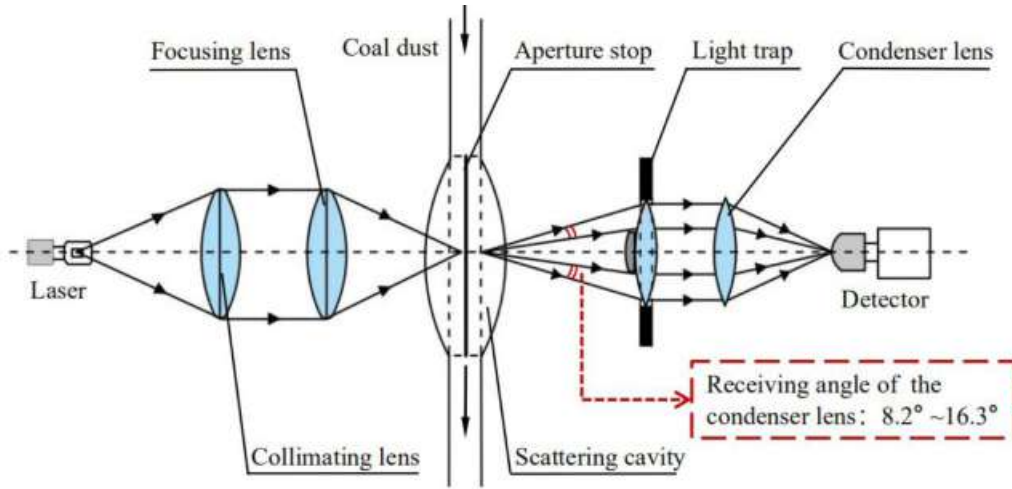


Figura 3.10: Possibile implementazione principio scattering Mie

3.1.5 Tecnologia scelta per l'implementazione del sensore: scattering laser basato sulla teoria di Mie

Durante la diffusione della luce laser (conosciuta anche come diffrazione laser, diffrattometria laser, diffrazione di Fraunhofer o diffusione di Mie), un raggio laser passa attraverso un campione di particelle disperse, le particelle grandi diffondono la luce ad angoli piccoli rispetto al raggio laser e le particelle piccole diffondono la luce ad angoli grandi. La dimensione delle particelle viene quindi determinata dalla variazione misurata dell'intensità della luce diffusa in funzione dell'angolo di diffusione, utilizzando la teoria di Mie della diffusione della luce. Rispetto alla diffusione dinamica della luce (DLS), il movimento delle particelle è meno importante poiché l'intensità della diffusione in funzione dell'angolo è una funzione diretta della dimensione delle particelle.

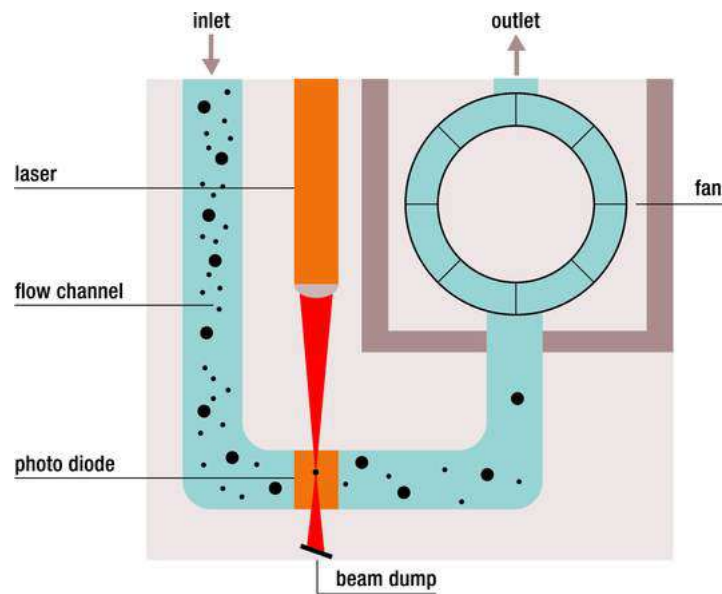


Figura 3.11: Primo esempio di sensore per il particolato ambientale che utilizza come tecnica di misurazione della densità di particolato lo scattering laser

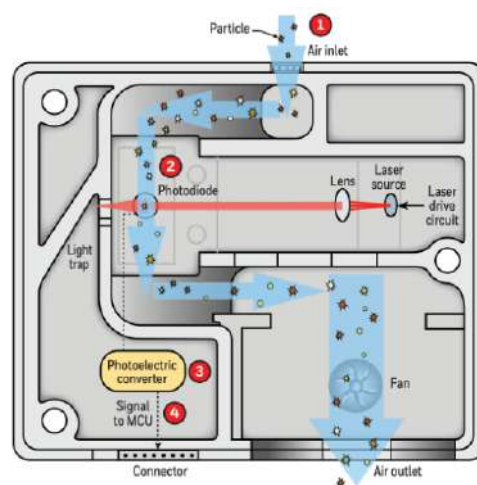


Figura 3.12: Secondo esempio di sensore per il particolato ambientale che utilizza come tecnica di misurazione della densità di particolato lo scattering laser

3.2 Strumenti utilizzati per l'implementazione del sensore

Dopo aver analizzato i principi teorici su cui si basa il funzionamento del sensore per il particolato ambientale, andiamo ad elencare, all'interno di tale sezione, gli strumenti utilizzati per la sua reale implementazione all'interno del lavoro di tirocinio.

3.2.1 Microcontrollore: Esp32



Figura 3.13: Esp32

Per la sua descrizione e caratterizzazione si rimanda al capitolo 2.1.1.

3.2.2 Laser: SYD1230 (Rosso)



Figura 3.14: SYD1230

Questo è il laser SYD1230.

Esso è caratterizzato da una luce laser di lunghezza d'onda 650nm, e quindi, sul rosso. Risulta essere molto facile da installare. Grazie alle sue caratteristiche di forte durata e soprattutto portabilità, risulta perfetto per l'utilizzo all'interno di un sensore portatile per la misura della densità del particolato ambientale.

Due osservazioni importanti prima di analizzare le specifiche più rilevanti:

- L'intensità del punto di messa a fuoco può essere regolata con l'aiuto della testa ad anello dell'obiettivo;
- La lente di focalizzazione può anche essere sostituita in base all'utilizzo che ne deve essere fatto.

SPECIFICATIONS	
Optical Power	5mW
Wavelength	650nm(red laser)
Voltage	DC 3-5V
Current	<40mA
Beam Shape	line
Emitting Angle	120°
Power Cord Length	135mm(5.31")
Lens Material	plastic
Size	Φ12X35mm
This laser module has high output power,please wear suitable goggles when testing or using.	

Figura 3.15: Specifiche laser SYD1230

3.2.3 Fotorilevatore: SFH 2401



Figura 3.16: SFH 2401

Il fotorilevatore SFH 2401 è l'ultimo membro della famiglia *Silicon PIN Photodiode* che fornisce una maggiore sensibilità alla luce laser per supportare le applicazioni di carattere industriale.

Cosa rende il prodotto unico?

- fotodiodo ad alta sensibilità con elevata linearità;
- eccezionale nelle applicazioni di rilevamento ad alte prestazioni come l'automazione industriale.

Quali sono i suoi punti di forza?

- tempo di risposta veloce;
- ideale per applicazioni a bassa luminosità quando è richiesto un alto rapporto segnale/rumore;
- rilevatore affidabile e ad alta sensibilità, eccezionale nelle applicazioni consumer, industriali e automobilistiche.

Riportiamo ora il grafico che descrive la sua sensibilità spettrale relativa (3.17), e cioè, la relazione che è presente fra l'intensità di luce scatterata e la lunghezza d'onda del laser utilizzato:

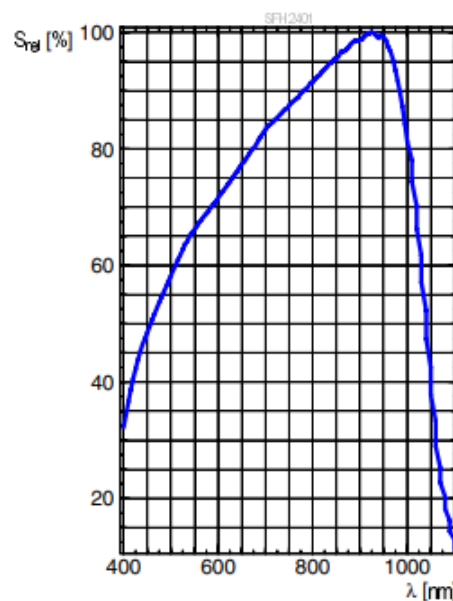


Figura 3.17: Sensibilità spettrale relativa SFH 2401

Vediamone rapidamente alcune delle più importanti caratteristiche:

Maximum Ratings

Parameter	Symbol	Values
Operating Temperature	T_{op}	min. -40 °C max. 85 °C
Storage temperature	T_{stg}	min. -40 °C max. 85 °C
Reverse voltage	V_R	max. 16 V
Total power dissipation	P_{tot}	max. 150 mW
ESD withstand voltage acc. to ANSI/ESDA/JEDEC JS-001 (HBM, Class 2)	V_{ESD}	max. 2 kV

Figura 3.18: Maximum ratings SFH 2401

Characteristics

$T_A = 25\text{ °C}$

Parameter	Symbol	Values
Spectral sensitivity $V_R = 5\text{ V}$; Std. Light A, $T = 2856\text{ K}$	S	typ. 76 nA/lx
Wavelength of max sensitivity	$\lambda_{S\text{ max}}$	typ. 950 nm
Spectral range of sensitivity	$\lambda_{10\%}$	typ. 300 ... 1100 nm
Radiant sensitive area	A	typ. 8.12 mm ²
Dimensions of active chip area	L x W	typ. 2.85 x 2.85 mm x mm
Half angle	φ	typ. 60 °
Dark current $V_R = 10\text{ V}$	I_R	typ. 1 nA max. 25 nA
Spectral sensitivity of the chip $V_R = 5\text{ V}$; $\lambda = 400\text{ nm}$	S_λ	typ. 0.2 A / W
Spectral sensitivity of the chip $V_R = 5\text{ V}$; $\lambda = 530\text{ nm}$	S_λ	typ. 0.42 A / W
Spectral sensitivity of the chip $V_R = 5\text{ V}$; $\lambda = 850\text{ nm}$	S_λ	typ. 0.66 A / W
Open-circuit voltage $E_v = 1000\text{ lx}$; Std. Light A, $T = 2856\text{ K}$	V_O	min. 300 mV typ. 350 mV
Short-circuit current $E_v = 1000\text{ lx}$; Std. Light A, $T = 2856\text{ K}$	I_{sc}	typ. 76 μA
Rise time $V_R = 5\text{ V}$; $R_L = 50\text{ }\Omega$; $\lambda = 850\text{ nm}$	t_r	typ. 0.04 μs
Fall time $V_R = 5\text{ V}$; $R_L = 50\text{ }\Omega$; $\lambda = 850\text{ nm}$	t_f	typ. 0.04 μs
Forward voltage $I_F = 100\text{ mA}$; $E = 0$; 0	V_F	typ. 1.2 V
Capacitance $V_R = 0\text{ V}$; $f = 1\text{ MHz}$; $E = 0$	C_O	typ. 65 pF
Temperature coefficient of voltage	TC_V	typ. -2.6 mV / K
Noise equivalent power $V_R = 5\text{ V}$; $\lambda = 400\text{ nm}$	NEP	typ. 0.0856 pW / Hz ^{1/2}
Noise equivalent power $V_R = 5\text{ V}$; $\lambda = 530\text{ nm}$	NEP	typ. 0.0429 pW / Hz ^{1/2}

Figura 3.19: Caratteristiche SFH 2401

3.2.4 Convertitore analogico-digitale: DollaTek ADS1115

L'ESP32 ha due moduli ADC incorporati, cioè ADC0 e ADC1 e ogni canale è di 12 bit. Il problema principale dell'ADC ESP32 è che ha caratteristiche non lineari; inoltre, non può differenziare tra i segnali di 1mv e 2mv, il che significa che offre una risoluzione molto bassa. Per risolvere questi problemi, possiamo usare un ADC esterno programmabile e ad alta risoluzione. Per lo sviluppo del sensore per la misura della densità di $PM_{2.5}$ e PM_{10} è stato preso in considerazione l'**ADS1115** (3.20).

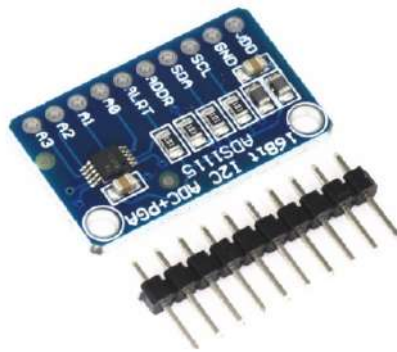


Figura 3.20: ADS1115

La risoluzione di questo ADC esterno è calcolata dividendo 6.144 con 32767 (16 bit) che è uguale a 0.1875mV.

Quindi la tensione analogica minima che possiamo misurare con l'ADC esterno I2C ADS1115 è 0,1875mV che è quasi il 50% di precisione in più rispetto al modulo convertitore analogico-digitale integrato dell'ESP32.

Vediamone rapidamente alcune delle più importanti caratteristiche:

ABSOLUTE MAXIMUM RATINGS⁽¹⁾

	ADS1113, ADS1114, ADS1115	UNIT
VDD to GND	-0.3 to +5.5	V
Analog input current	100, momentary	mA
Analog input current	10, continuous	mA
Analog input voltage to GND	-0.3 to VDD + 0.3	V
SDA, SCL, ADDR, ALERT/RDY voltage to GND	-0.5 to +5.5	V
Maximum junction temperature	+150	°C
Storage temperature range	-60 to +150	°C

Figura 3.21: Maximum ratings ADS1115

ELECTRICAL CHARACTERISTICS

All specifications at -40°C to $+125^{\circ}\text{C}$, $\text{VDD} = 3.3\text{V}$, and Full-Scale (FS) = $\pm 2.048\text{V}$, unless otherwise noted. Typical values are at $+25^{\circ}\text{C}$.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115			UNIT
		MIN	TYP	MAX	
ANALOG INPUT					
Full-scale input voltage ⁽¹⁾	$V_{IN} = (\text{AIN}_P) - (\text{AIN}_N)$		$\pm 4.096/\text{PGA}$		V
Analog input voltage	AIN_P or AIN_N to GND	GND		VDD	V
Differential input impedance			See Table 2		
Common-mode input impedance	FS = $\pm 6.144\text{V}^{(1)}$		10		MΩ
	FS = $\pm 4.096\text{V}^{(1)}$, $\pm 2.048\text{V}$		6		MΩ
	FS = $\pm 1.024\text{V}$		3		MΩ
	FS = $\pm 0.512\text{V}$, $\pm 0.256\text{V}$		100		MΩ
SYSTEM PERFORMANCE					
Resolution	No missing codes	16			Bits
Data rate (DR)			8, 16, 32, 64, 128, 250, 475, 860		SPS
Data rate variation	All data rates	-10		10	%
Output noise		See Typical Characteristics			
Integral nonlinearity	DR = 8SPS, FS = $\pm 2.048\text{V}$, best fit ⁽²⁾			1	LSB
Offset error	FS = $\pm 2.048\text{V}$, differential inputs		± 1	± 3	LSB
	FS = $\pm 2.048\text{V}$, single-ended inputs		± 3		LSB
Offset drift	FS = $\pm 2.048\text{V}$		0.005		LSB/°C
Offset power-supply rejection	FS = $\pm 2.048\text{V}$		1		LSB/V
Gain error ⁽³⁾	FS = $\pm 2.048\text{V}$ at 25°C		0.01	0.15	%
Gain drift ⁽³⁾	FS = $\pm 0.256\text{V}$		7		ppm/°C
	FS = $\pm 2.048\text{V}$		5	40	ppm/°C
	FS = $\pm 6.144\text{V}^{(1)}$		5		ppm/°C
Gain power-supply rejection			80		ppm/V
PGA gain match ⁽³⁾	Match between any two PGA gains		0.02	0.1	%
Gain match	Match between any two inputs		0.05	0.1	%
Offset match	Match between any two inputs		3		LSB
Common-mode rejection	At dc and FS = $\pm 0.256\text{V}$		105		dB
	At dc and FS = $\pm 2.048\text{V}$		100		dB
	At dc and FS = $\pm 6.144\text{V}^{(1)}$		90		dB
	$f_{CM} = 60\text{Hz}$, DR = 8SPS		105		dB
	$f_{CM} = 50\text{Hz}$, DR = 8SPS		105		dB
DIGITAL INPUT/OUTPUT					
Logic level					
V_{IH}		0.7VDD		5.5	V
V_{IL}		GND - 0.5		0.3VDD	V
V_{OL}	$I_{OL} = 3\text{mA}$	GND	0.15	0.4	V
Input leakage					
I_H	$V_{IH} = 5.5\text{V}$			10	μA
I_L	$V_{IL} = \text{GND}$	10			μA

Figura 3.22: Caratteristiche ADS1115

3.2.5 Ventola: AERZETIX

Tale ventola ha la caratteristica di essere di tipo assiale ed in materiale termoplastico. Possiede un motore senza spazzole alimentato a 5V DC. Integrato al suo interno è presente un cuscinetto Vapo. Fa parte della classe di isolamento A e presenta una resistenza di isolamento minima di 10M Ω .

Le dimensioni del corpo sono le seguenti: 20x20x8mm.

Inoltre, presenta 2 cavi di lunghezza 150mm per l'alimentazione.



Figura 3.23: Ventola AERZETIX 20x20x8

3.3 Firmware implementato

Il firmware realizzato per il corretto funzionamento del sensore è il seguente:

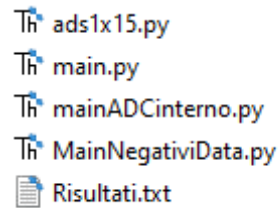


Figura 3.24: Firmware sensore particolato ambientale

dove:

- Il file **ads1x15.py** è la libreria realizzata per implementare il convertitore analogico/digitale *ADS1115*;
- Il file **main.py** è il cuore del firmware, dove vengono gestite tutte le operazioni per il corretto funzionamento del sensore;
- **Gli ultimi due file**, invece, sono dei file test realizzati durante la fase di sviluppo del sensore.

Tutti i dati provenienti dal dispositivo vengono poi salvati all'interno del file "*Risultati.txt*" (3.25). Attenzione però, una volta inizializzata un'ulteriore misurazione della densità del particolato ambientale, i nuovi dati si andranno a sovrascrivere ai vecchi.

```
----- INTENSITÀ LUCE SCATTERATA -----  
array('h', [55, 55, 55, 87, 87, 87, 36, 36, 36, 124, 124, 21, 21, 21, 21, 134, 23, 23, 23, 113, 52, 52, 52, 66, 66, 66, 66, 90, 90, 39, 39, 39, 39, 116, 13, 13,  
, 19, 19, 19, 133, 133, 19, 19, 86, 86, 86, 86, 69, 69, 69, 54, 54, 54, 97, 24, 24, 24, 129, 129, 16, 16, 128, 128, 128, 22, 22, 87, 87, 87, 87, 88, 88, 88, 49,  
129, 17, 17, 127, 127, 127, 34, 34, 87, 87, 87, 87, 84, 84, 84, 50, 50, 106, 23, 23, 23, 133, 133, 20, 20, 122, 122, 122, 39, 39, 79, 79, 79, 79, 81, 81, 1  
, 106, 106, 57, 57, 74, 74, 74, 74, 87, 87, 87, 45, 45, 45, 131, 20, 20, 20, 139, 139, 26, 26, 121, 121, 121, 53, 53, 68, 68, 68, 68, 91, 41, 41, 41, 126, 20, 2  
3, 103, 34, 34, 34, 123, 21, 21, 21, 129, 129, 29, 29, 89, 89, 89, 89, 70, 70, 70, 58, 58, 58, 101, 26, 26, 26, 128, 128, 15, 15, 125, 125, 125, 40, 40, 89, 89  
-----PM2.5 e PM10-----  
La media di scatter di PM2.5 è:113.378  
La media di scatter di PM10 è:134.0  
----- LA DENSITÀ STIMATA DI PM2.5 È:-----  
3  
----- LA DENSITÀ STIMATA DI PM10 È:-----  
3
```

Figura 3.25: File Risultati.txt

Tale procedura di scrittura su un file .txt è resa possibile attraverso tali linee di codice implementate al termine del file *main.py*:

```

1 with open("Risultati.txt", "w") as txt_file:
2     txt_file.write(" ----- INTENSITA LUCE SCATTERATA ----- \n")
3     for line in str(data):
4         txt_file.write(" ".join(line)) # works with any number of elements in a
         line
5     txt_file.write(" \n
        -----PM2
        .5 e PM10
        -----")
6     txt_file.write("\n La media di scatter di PM2.5 e':")
7     for line in str(int(media25)):
8         txt_file.write(" ".join(line))
9     txt_file.write("\n La media di scatter di PM10 e':")
10    for line in str(int(media10)):
11        txt_file.write(" ".join(line))
12    txt_file.write("\n ----- LA DENSITA' STIMATA DI PM2.5 e':----- \
        n")
13    for line in str(densitaPm25):
14        txt_file.write(" ".join(line))
15    txt_file.write("\n ----- LA DENSITA' STIMATA DI PM10 e':----- \n
        ")
16    for line in str(densitaPm10):
17        txt_file.write(" ".join(line))

```

Elenco dei listati 5: main.py txt file

3.3.1 Listati

In tale sezione, sono rese disponibili le parti salienti dei listati che compongono il firmware del sensore per la densità di particolato ambientale.

```

1 def sample(x, adc = ads.read_rev, data=data, timestamp = timestamp):
2     global index_put, irq_busy, Pm10, Pm25, data_reali_25, data_reali_10
3     if irq_busy:
4         return
5     irq_busy = True
6     while index_put < _BUFFERSIZE:
7         timestamp[index_put] = ticks_us()
8         if adc() < 0:
9             data[index_put] = 0
10            index_put += 1
11            irq_busy = False
12        if adc() >=0 and adc() <= 90:
13            data[index_put] = adc()
14            data_reali_25[index_put] = 0
15            data_reali_10[index_put] = 0
16            index_put += 1
17            irq_busy = False
18        if adc() > 90 and adc() < 130:
19            data[index_put] = adc()
20            data_reali_25[index_put] = adc()
21            data_reali_10[index_put] = 0
22            index_put += 1
23            Pm25 += 1
24            irq_busy = False
25        else:
26            data[index_put] = adc()
27            data_reali_25[index_put] = 0
28            data_reali_10[index_put] = adc()

```



```

29     index_put += 1
30     Pm10 += 1
31     irq_busy = False
32     irq_busy = False

```

Elenco dei listati 6: Interrupt service routine for data acquisition called by a timer interrupt

```

1
2 index_put = 0
3
4 #ADS1115 data conversion rate
5 ADC_RATE = 4
6
7 #Optimal RATE and RATE_SET_CONV = 4.
8 start_time = time.time()
9
10 #Start the first conversion
11 ads.set_conv(4, 0)
12 ads.read_rev()
13
14 sleep_ms(ADC_RATE)
15 tim = Timer(-1)
16 tim.init(period=ADC_RATE, mode=Timer.PERIODIC, callback=sample)
17
18 while index_put < _BUFFERSIZE:
19     pass
20
21 fine = time.time() - start_time
22
23 tim.deinit()
24
25 print("--- %s seconds ---" % (fine))
26
27 data_reali_25 = check_25(data_reali_25)
28 data_reali_10 = check_10(data_reali_10)
29
30 #passing PM10 data into an empty vector where the scattered light values can
    be averaged
31 while j < _BUFFERSIZE:
32     if data_reali_10[j] != 0:
33         data_reali_10_nonnulli[h] = data_reali_10[j]
34         j += 1
35         h += 1
36     else:
37         j += 1
38
39 #passing PM2.5 data into an empty vector where the scattered light values
    can be averaged
40 while k < _BUFFERSIZE:
41     if data_reali_25[k] != 0:
42         data_reali_25_nonnulli[g] = data_reali_25[k]
43         k += 1
44         g += 1
45     else:
46         k += 1

```

Elenco dei listati 7: Main part of the code main.py

```

1 class ADS1115:
2     def __init__(self, i2c, address=0x48, gain=1):
3         self.i2c = i2c
4         self.address = address

```

```

5     self.gain = gain
6     self.temp2 = bytearray(2)
7
8     def _write_register(self, register, value):
9         self.temp2[0] = value >> 8
10        self.temp2[1] = value & 0xff
11        self.i2c.writeto_mem(self.address, register, self.temp2)
12
13    def _read_register(self, register):
14        self.i2c.readfrom_mem_into(self.address, register, self.temp2)
15        return (self.temp2[0] << 8) | self.temp2[1]
16
17    def raw_to_v(self, raw):
18        v_p_b = _GAINS_V[self.gain] / 32767
19        return raw * v_p_b
20
21    def set_conv(self, rate=4, channel1=0, channel2=None):
22        """Set mode for read_rev"""
23        self.mode = (_CQUE_NONE | _CLAT_NONLAT |
24                    _CPOL_ACTVLOW | _CMODE_TRAD | _RATES[rate] |
25                    _MODE_SINGLE | _OS_SINGLE | _GAINS[self.gain] |
26                    _CHANNELS[(channel1, channel2)])
27
28    def read(self, rate=4, channel1=0, channel2=None):
29        """Read voltage between a channel and GND.
30        Time depends on conversion rate."""
31        self._write_register(_REGISTER_CONFIG, (_CQUE_NONE | _CLAT_NONLAT |
32                                                _CPOL_ACTVLOW | _CMODE_TRAD | _RATES[rate] |
33                                                _MODE_SINGLE | _OS_SINGLE | _GAINS[self.gain] |
34                                                _CHANNELS[(channel1, channel2)]))
35        while not self._read_register(_REGISTER_CONFIG) & _OS_NOTBUSY:
36            time.sleep_ms(1)
37        res = self._read_register(_REGISTER_CONVERT)
38        return res if res < 32768 else res - 65536
39
40    def read_rev(self):
41        """Read voltage between a channel and GND. and then start
42        the next conversion."""
43        res = self._read_register(_REGISTER_CONVERT)
44        self._write_register(_REGISTER_CONFIG, self.mode)
45        return res if res < 32768 else res - 65536
46
47    def alert_start(self, rate=4, channel1=0, channel2=None,
48                   threshold_high=0x4000, threshold_low=0, latched=False):
49        """Start continuous measurement, set ALERT pin on threshold."""
50        self._write_register(_REGISTER_LOWTHRESH, threshold_low)
51        self._write_register(_REGISTER_HITHRESH, threshold_high)
52        self._write_register(_REGISTER_CONFIG, _CQUE_1CONV |
53                            _CLAT_LATCH if latched else _CLAT_NONLAT |
54                            _CPOL_ACTVLOW | _CMODE_TRAD | _RATES[rate] |
55                            _MODE_CONTIN | _GAINS[self.gain] |
56                            _CHANNELS[(channel1, channel2)])
57
58    def conversion_start(self, rate=4, channel1=0, channel2=None):
59        """Start continuous measurement, trigger on ALERT/RDY pin."""
60        self._write_register(_REGISTER_LOWTHRESH, 0)
61        self._write_register(_REGISTER_HITHRESH, 0x8000)
62        self._write_register(_REGISTER_CONFIG, _CQUE_1CONV | _CLAT_NONLAT |
63                            _CPOL_ACTVLOW | _CMODE_TRAD | _RATES[rate] |
64                            _MODE_CONTIN | _GAINS[self.gain] |

```

```
65         _CHANNELS[(channel1, channel2)])
66
67     def alert_read(self):
68         """Get the last reading from the continuous measurement."""
69         res = self._read_register(_REGISTER_CONVERT)
70         return res if res < 32768 else res - 65536
71
```

Elenco dei listati 8: class ADS1115

4 Prototipi

Dopo aver analizzato i principi teorici su cui si basa il funzionamento del sensore per il particolato ambientale e gli strumenti che verranno utilizzati all'interno di esso, per valutare se tali principi e strumenti siano adatti oppure no alla reale misurazione delle polveri sottili, sono stati creati in collaborazione con IDEA due prototipi.

Il primo prototipo è servito come test per poter implementare il principio dello scattering laser. Il secondo segna, invece, una netta evoluzione rispetto al primo.

4.1 Prototipo 1

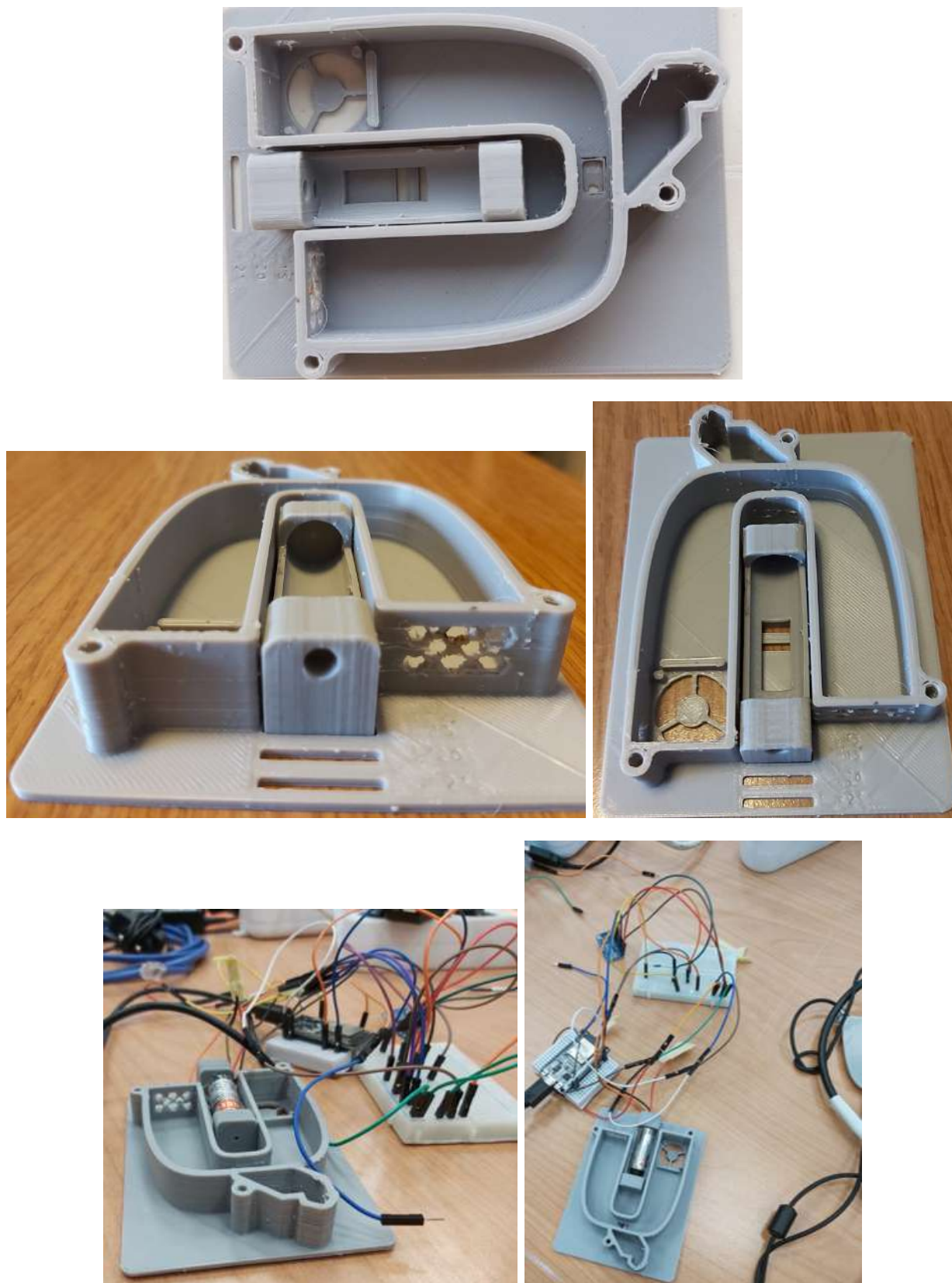


Figura 4.1: Prototipo 1 sensore particolato ambientale

Tale prototipo, come già detto, è stato realizzato per testare il principio della scattering laser fino ad ora studiato solo dal punto di vista teorico.

Dopo aver verificato il suo reale funzionamento, sono state discusse alcune sue caratteristiche costruttive.

Infatti, nell'andare a creare il secondo ed ultimo prototipo, sono stati modificati i seguenti punti:

- Diminuita la lunghezza dei canali di 20mm;
- Diminuita la larghezza del canale di ingresso di 10mm;
- Modificata la parte d'uscita dell'aria nel seguente modo: vista la presenza di una ventola assiale e non radiale, il foro che ora funge da uscita per l'aria, posizionato sul pavimento del sensore, nel secondo prototipo, è stato trasformato in un foro di ingresso dell'aria, posizionato sulla parete finale del canale;
- Aumentata la dimensione dell'alloggiamento del laser;
- Rialzato l'alloggiamento del fotorecettore in modo tale che si trovi a solo 1.5mm di distanza dal raggio del laser;
- Sostituito il foro d'ingresso della camera di scattering con un'apertura;
- Aumentata di mezzo millimetro la dimensione dei fori calibrati.

4.2 Prototipo 2 - finale

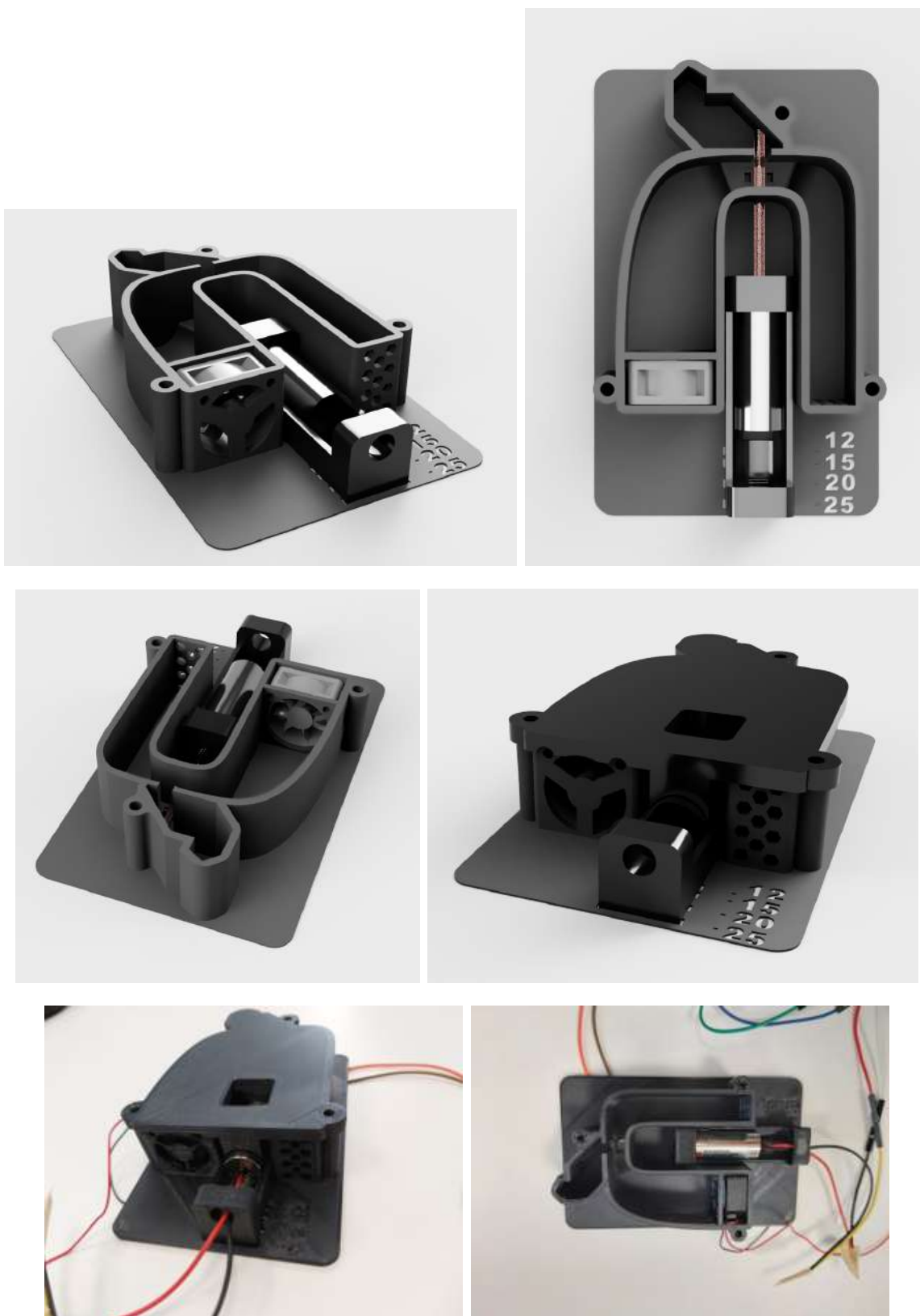


Figura 4.2: Secondo ed ultimo prototipo sensore particolato ambientale

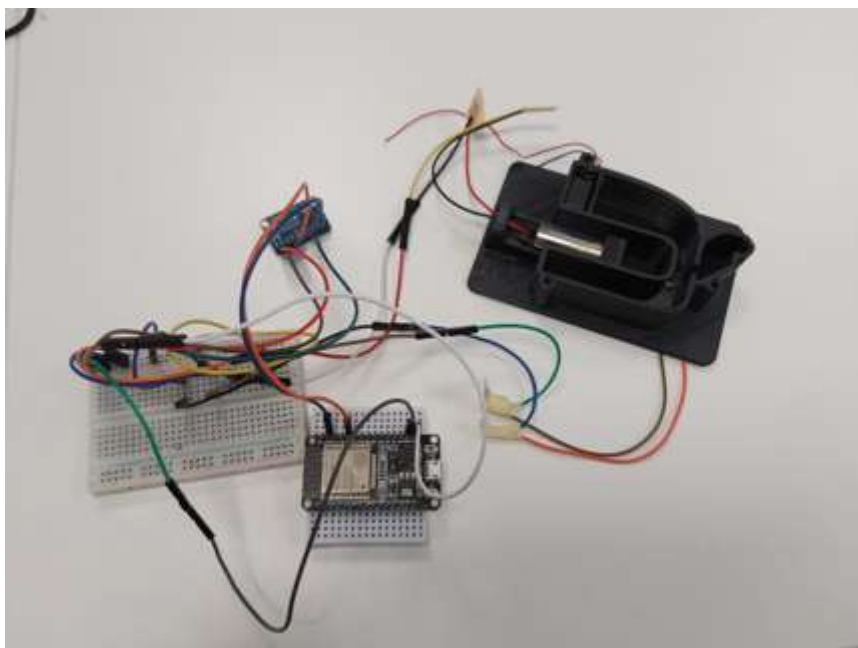


Figura 4.3: Sistema finale realizzato per l'implementazione del sensore per la misurazione della densità di particolato ambientale

4.3 Relazione fra intensità di luce scatterata e dimensione di particolato PM_{2.5} e PM₁₀

Per derivare una funzione di correlazione che leghi lo scattering della luce e la densità di particolato sono state effettuate moltissime prove incrociate fra:

1. Il sensore per la misura della densità di particolato ambientale sviluppato all'interno di IDEA;
2. Il sensore SM-UART-04L implementato in DoNet.

Prima di parlare della funzione di correlazione che lega l'intensità di luce scatterata con la densità di PM_{2.5} e PM₁₀, dobbiamo parlare di come legare tale intensità con la dimensione del particolato PM_{2.5} e PM₁₀.

L'intensità di luce scatterata, in base alle prove effettuate, è stata suddivisa in due gruppi:

- Se Valore_Rilevato_ADC > 90 && Valore_Rilevato_ADC < 130 → PM_{2.5};
- Se Valore_Rilevato_ADC > 130 → PM₁₀;
- Se Valore_Rilevato_ADC < 90 → valore non di picco e quindi trascurabile dato che la curva di rilevamento è una gaussiana. A noi interessa solo il valore di picco(4.4);

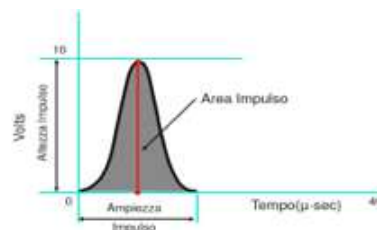


Figura 4.4: Gaussiana rilevamento particolato

- Se Valore_Rilevato_ADC < 0 → piccolo errore di misurazione → imposta il valore a 0.

Di seguito vengono riportate le linee di codice utilizzate per effettuare tale suddivisione in gruppi:

```
1 if adc() < 0:
2     data[index_put] = 0
3     index_put += 1
4     irq_busy = False
5 if adc() >=0 and adc() <= 90:
6     data[index_put] = adc()
7     data_reali_25[index_put] = 0
8     data_reali_10[index_put] = 0
9     index_put += 1
10    irq_busy = False
11 if adc() > 90 and adc() < 130:
12     data[index_put] = adc()
13     data_reali_25[index_put] = adc()
14     data_reali_10[index_put] = 0
15     index_put += 1
16     Pm25 += 1
17     irq_busy = False
18 else:
19     data[index_put] = adc()
20     data_reali_25[index_put] = 0
```

```
21 data_reali_10[index_put] = adc()  
22 index_put += 1  
23 Pm10 += 1  
24 irq_busy = False
```

Elenco dei listati 9: Funzione di correlazione dimensione particolato-intensità luce scatterata

4.4 Relazione fra intensità di luce scatterata di PM_{2.5} e PM₁₀ e densità di particolato PM_{2.5} e PM₁₀

Per ricavare poi la relazione fra l'intensità di luce scatterata e la densità di particolato ambientale, all'interno degli intervalli definiti nello scorso paragrafo per il PM_{2.5} e PM₁₀ è stata calcolata la media dello scattering delle particelle PM_{2.5} e PM₁₀.

In base a tale media all'interno di un intervallo di 1024 campioni è stata definita la correlazione fra l'intensità di luce scatterata e la densità di particolato:

```
1 estremosx25 = 110
2 estremox25 = 111
3
4 estremosx10 = 130
5 estremox10 = 131
6
7 if media10 > 250:
8     densitaPm10 = 999
9     densitaPm25 = 999
10    print("Raggiunto il valore di saturazione del sensore \n")
11 elif media10 <= 0 and media25 > 0:
12     densitaPm10 = 999
13     densitaPm25 = 999
14    print("Raggiunto il valore di saturazione del sensore")
15 elif media25 <= 0 and media10 > 0:
16     densitaPm10 = 999
17     densitaPm25 = 999
18    print("Raggiunto il valore di saturazione del sensore")
19 elif media10 > 240 and media25 < 130:
20     densitaPm10 = 999
21     densitaPm25 = 999
22    print("Raggiunto il valore di saturazione del sensore")
23 elif media25 > 140:
24     densitaPm10 = 999
25     densitaPm25 = 999
26    print("Raggiunto il valore di saturazione del sensore")
27 elif media25 >= 105 and media25 < 106:
28     densitaPm25 = 1
29 elif media25 >= 106 and media25 < 107:
30     densitaPm25 = 2
31 elif media25 >= 107 and media25 < 108:
32     densitaPm25 = 3
33 elif media25 >= 108 and media25 < 109:
34     densitaPm25 = 4
35 elif media25 >= 109 and media25 < 110:
36     densitaPm25 = 5
37 else:
38     while True:
39         densitaPm10 += 1
40         estremosx10 += 1
41         estremox10 += 1
42         if(densitaPm10 == 999 or densitaPm25 == 999):
43             break
44         if (media10 >= estremosx10 and media10 <= estremox10):
45             break
46     while True:
47         densitaPm25 += 1
48         estremosx25 += 1
49         estremox25 += 1
50         if(densitaPm10 == 999 or densitaPm25 == 999):
51             break
```

```

52  if (media25 >= estremosx25 and media25 <= estremodx25):
53      break
54
55
56  if int(media10) >= 200 and int(media10) <= 250:
57      densitaPm10 = 0
58      estremosx10 = 200
59      estremodx10 = 201
60      while True:
61          densitaPm10 += 1
62          estremosx10 += 1
63          estremodx10 += 1
64  if(densitaPm10 == 999 or densitaPm25 == 999):
65      densitaPm10 == 999
66      densitaPm25 == 999
67      break
68  if (media10 >= estremosx10 and media10 <= estremodx10):
69      break
70
71  if int(media10) == 210:
72      densitaPm10 = 45
73      densitaPm25 = 30

```

Elenco dei listati 10: Funzione di correlazione densità particolato-intensità luce scatterata

I valori numerici di correlazione appena proposti all'interno di tale listato, al variare di alcune condizioni, che ora verranno elencate, potrebbero cambiare:

- tipologia di laser;
- tipologia di fotorilevatore;
- tipologia di ventola;
- distanza fra rilevatore e laser;
- posizione laser;
- intensità del raggio laser.

5 Risultati finali

In questo paragrafo vengono riportati i valori della densità di particolato ambientale ottenuti andando ad utilizzare il prototipo finale numero due del nuovo sensore progettato all'interno del progetto di tirocinio e tesi.

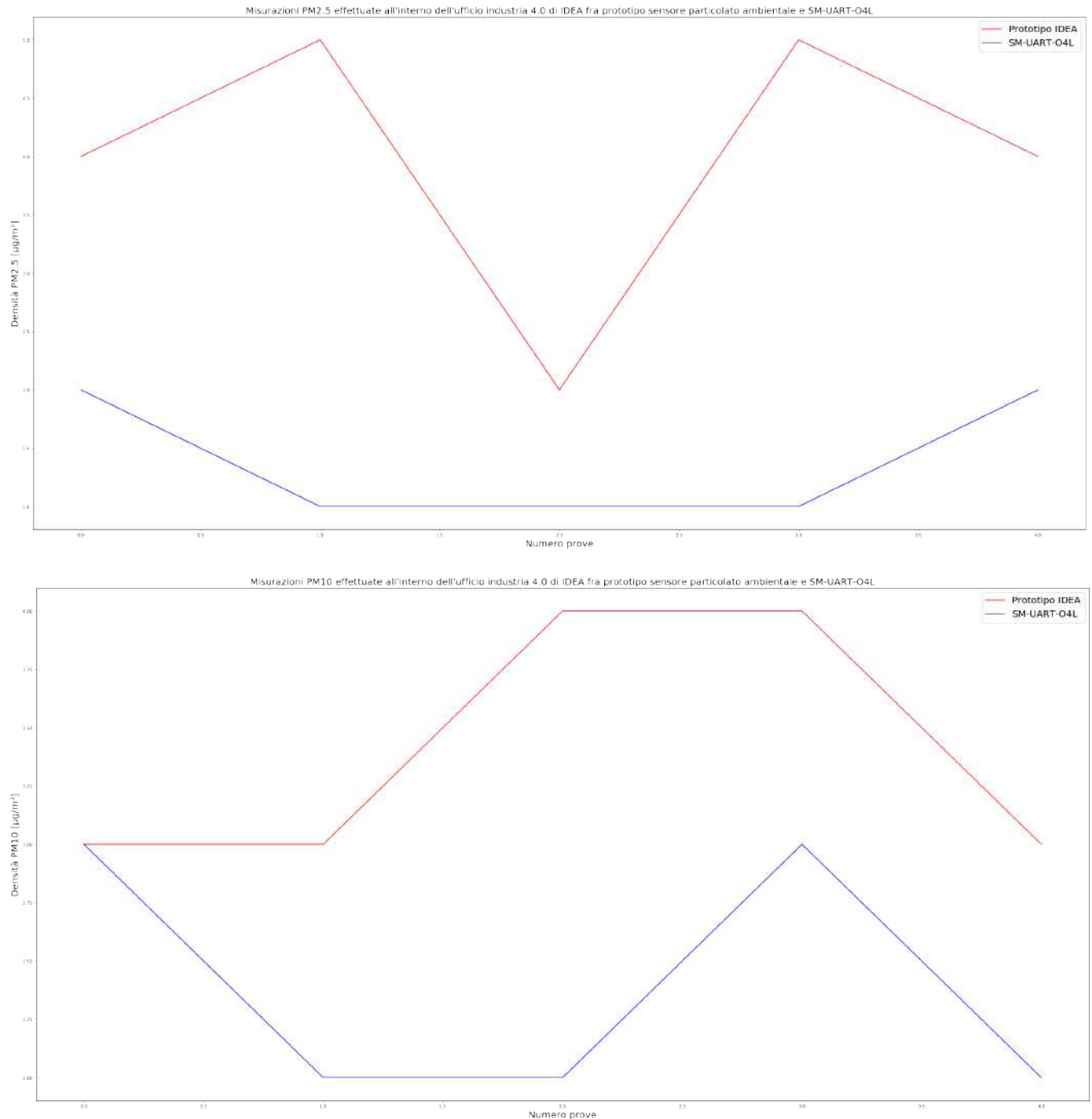


Figura 5.1: Confronto fra risultati ottenuti sensore IDEA e SM-UART-04L in condizioni normali all'interno dell'ufficio industria 4.0 di IDEA con ricircolo aria

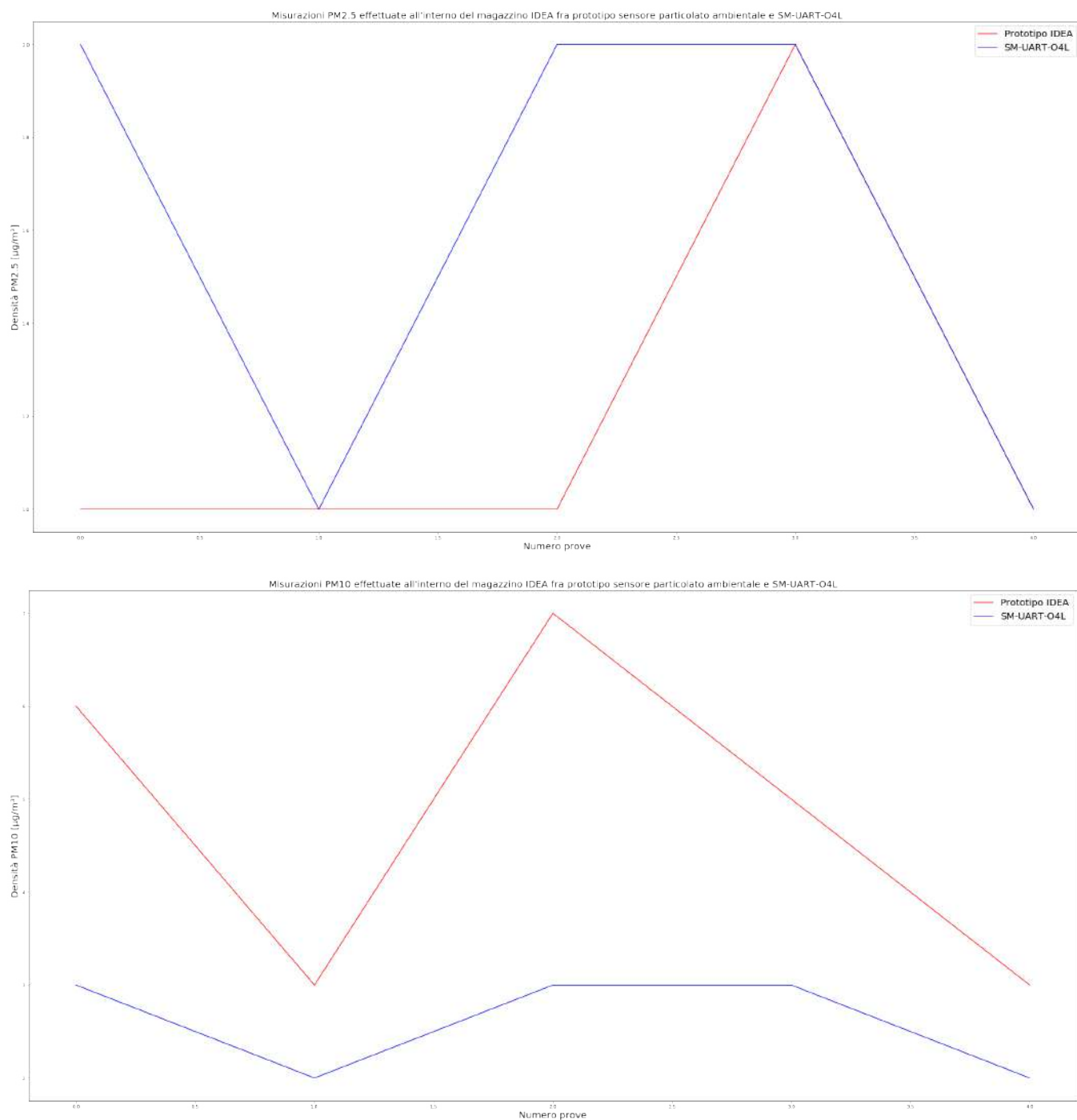


Figura 5.2: Confronto fra risultati ottenuti sensore IDEA e SM-UART-04L in condizioni normali all'interno del deposito al primo piano di IDEA senza ricircolo aria

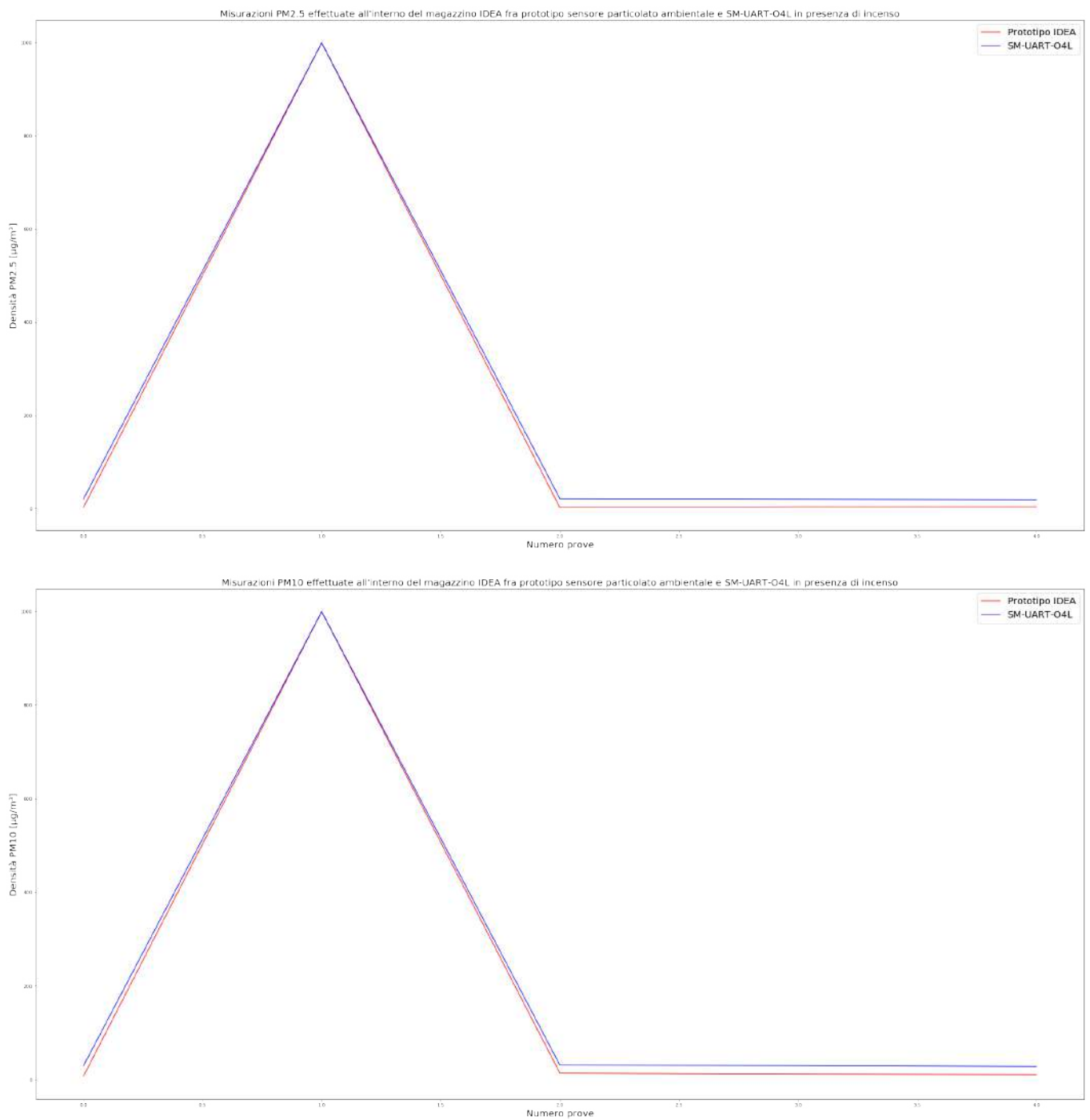


Figura 5.3: Confronto fra risultati ottenuti sensore IDEA e SM-UART-04L in condizioni di medio inquinamento dell'aria simulato con incenso all'interno del deposito al primo piano di IDEA senza ricircolo aria

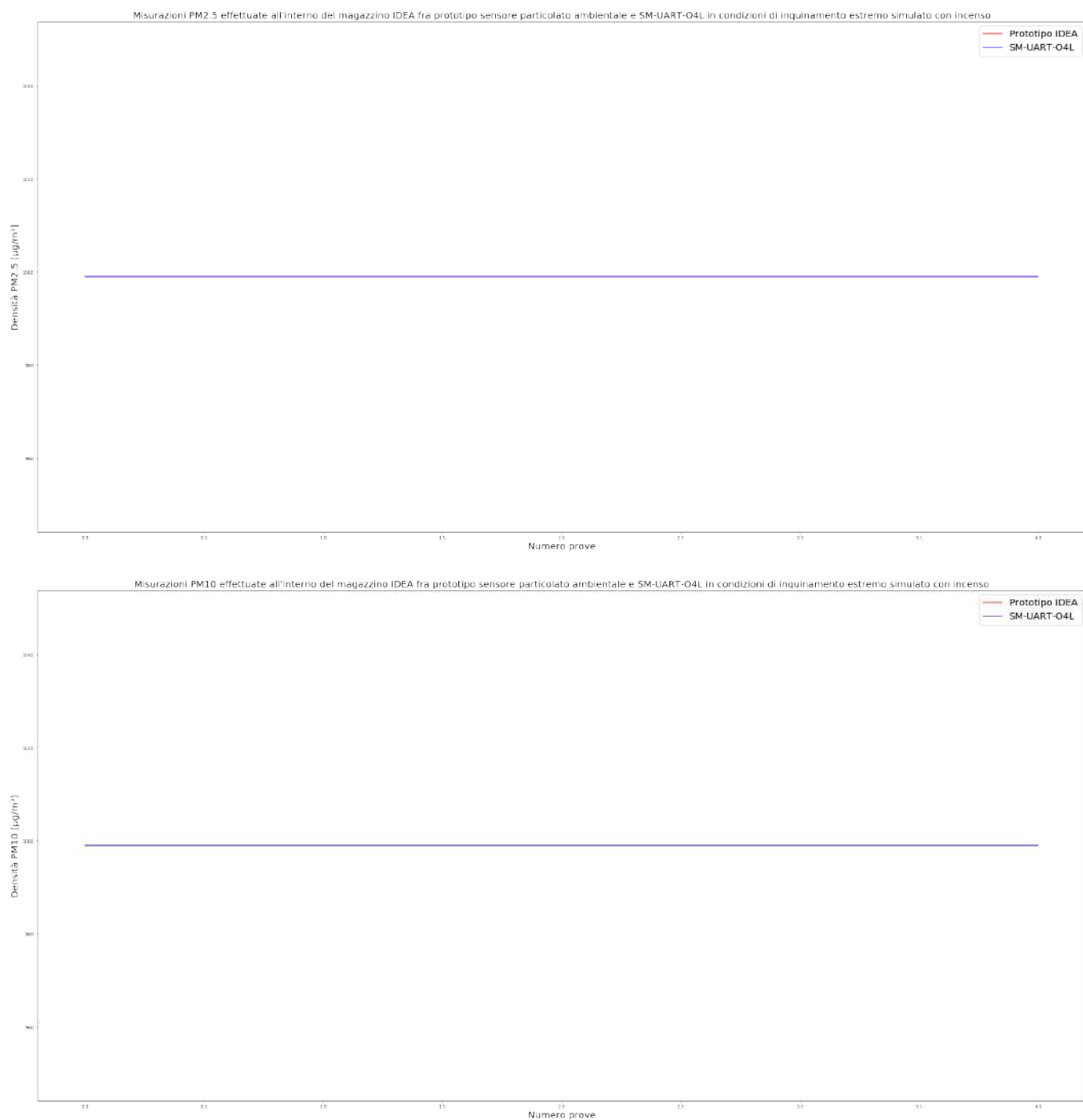


Figura 5.4: Confronto fra risultati ottenuti sensore IDEA e SM-UART-04L in condizioni di estremo inquinamento dell'aria simulato con incenso all'interno del deposito al primo piano di IDEA senza ricircolo aria

6 Conclusioni

In questa tesi si è visto sia come realizzare da zero un sensore per la misura della densità del particolato ambientale, sia come implementarne uno già esistente in commercio all'interno di un dispositivo.

In particolare, nella prima parte del progetto di tesi, è stato utilizzato il dispositivo DoNet progettato dall'azienda *IDEA Soc. Coop.*, in grado di monitorare la temperatura, l'umidità relativa, la luminosità, l'anidride carbonica, i composti organici volatili nella stanza in cui è installato, e ora, grazie all'implementazione del sensore per polveri sottili SM-UART-04L al suo interno, anche il PM_{2.5} e PM₁₀.

Il dispositivo è connesso alla rete Internet tramite un router WiFi locale, pubblicando i dati attraverso il protocollo di comunicazione MQTT in un gateway di raccolta, il quale si occupa d'inviare le misure a un database e a un sito web, per controllare gli andamenti delle grandezze. In un secondo momento DoNet si è evoluto, introducendo un servizio di aggiornamento del firmware Over The Air in modo da distribuire gli aggiornamenti da remoto senza dover intervenire in ogni singolo dispositivo della rete wireless. OTA si serve di un repository privato su GitHub nel quale caricare le versioni del codice, quindi DoNet è in grado di accedere attraverso un token di autenticazione e scaricare gli aggiornamenti.

Un'altra considerazione che va assolutamente fatta riguarda la manutenzione del dispositivo. Questa tipologia di sensori presenta sempre una camera in cui viene generato il flusso d'aria e in cui vengono per forza di cose trasportate particelle che potrebbero depositarsi sulle lenti e sugli ingressi e uscite dell'aria dalla camera. L'utilizzo continuativo del sensore porterebbe quindi, successivamente al deposito di particelle, a un inevitabile calo delle prestazioni dello stesso. Si necessita per cui di una manutenzione periodica.

Nella seconda parte del progetto di tesi, invece, è stato realizzato da zero, partendo dalla progettazione arrivando alla creazione di due prototipi, un sensore per la misurazione della densità del particolato ambientale PM_{2.5} e PM₁₀.

Naturalmente lo sviluppo di tale dispositivo non è ancora completo. Il prototipo, però, mostra risultati soddisfacenti e in linea con gli esiti attesi.

6.1 Sviluppi futuri

Il dispositivo DoNet è attualmente ancora in fase di sviluppo. La prossima miglioria potrebbe riguardare sicuramente la modifica della custodia in modo tale da rendere sicura e affidabile l'implementazione già effettuata del SM-UART-04L.

È prevista anche l'aggiunta di un display per la visualizzazione delle letture o per la segnalazione di messaggi di alert. Con questi sensori nuovi si potrebbe valutare la realizzazione di più versioni di DoNet differenti dal punto di vista hardware ma allo stesso tempo identici, predisponendo la versione base a ospitare comunque tutti i sensori, mentre per il firmware, può risultare importante l'implementazione dell'aggiornamento OTA. Si potrebbe anche valutare la modifica del sistema di alimentazione utilizzando una batteria che riesca ad alimentare DoNet per un tempo soddisfacente. Per quanto riguarda le migliorie nel software, le idee sono tante. Si sta lavorando sull'integrazione di DoNet con Alexa e Google Home. Si vorrebbe integrare il dispositivo anche con elementi domotici della casa, per poi implementare algoritmi di machine learning e d'intelligenza artificiale, offrendo un'ulteriore analisi dei dati raccolti dai dispositivi come per esempio la valutazione della qualità dell'aria e dei consumi energetici. Infine, si potrebbe migliorare la piattaforma IoT, con l'aggiunta di funzionalità in più nell'applicazione web e la realizzazione di un'applicazione per smartphone.

Per quanto riguarda, invece, il sensore per il particolato ambientale, in secondo momento si potrebbe:

- variare la strumentazione utilizzata in modo tale da migliorare la precisione della misura;
- studiare i flussi d'aria che si vengono a creare all'interno del sensore in modo tale da migliorare il risultato finale;
- una volta scelta la strumentazione in maniera definitiva, applicare dei filtri all'ingresso e all'uscita del sensore in modo tale da caratterizzare in maniera univoca la relazione che c'è fra intensità di luce scatterata e densità di particolato ambientale;
- se possibile, introdurre il sensore all'interno di un ambiente controllato (in cui si conoscono le dimensioni delle particelle di particolato ambientale) in modo tale da relazionare in maniera precisa e accurata intensità di luce scatterata e densità di particolato.

Riferimenti bibliografici

- [1] Inquinamento e COVID-19: cosa sappiamo. <https://www.hsr.it/news/2020/maggio/inquinamento-coronavirus>, 2020.
- [2] Francesca Costabile Daniele Contini. Does Air Pollution Influence COVID-19 Outbreaks? atmosphere, 2020.
- [3] Chip ESP32. <https://www.espressif.com/en/products/socs/esp32>, 2020.
- [4] MicroPython documentation. <http://docs.micropython.org/en/latest/>.
- [5] Python download website. <https://www.python.org/>.
- [6] MicroPython download website. <https://micropython.org/>.
- [7] API GitHub. <https://developer.github.com/v3/>, 2020.
- [8] Metodi di misura. <https://www.chiamamilano.it/metodi-di-misura.html>.
- [9] Criteri per l’allineamento dei misuratori PM2.5-PM10 low-cost del progetto “Che Aria Tira?” ai misuratori di riferimento Qbit LCT-14. 2019.
- [10] Abigail R. Lara. *Patologie correlate all’inquinamento atmosferico*, 2018.
- [11] Minna Aurela Kimmo Teinilä Samu Varjonen Óscar González Hilkka Timonen Joel Kuula, Timo Mäkelä. *Laboratory evaluation of particle sizeselectivity of optical low-cost particulate matter sensors*. Atmospheric Measurement Techniques discussion, 2020.
- [12] Hao Zhang, Wen Nie, Yu Liang, Jiange Chen, Huitian Peng, *Development and performance detection of higher precision optical sensor for coal dust concentration measurement based on Mie scattering theory*, China, 2021.
- [13] M. Letizi. *Sviluppo di un servizio OTA per la programmazione di un sensore ambientale custom*. PhD thesis, Università Politecnica delle Marche, 2019-2020.
- [14] A. Rubechini. *Progettazione di una Wireless Sensor Network per la gestione del benessere termico igrometrico in un edificio*. PhD thesis, Università Politecnica delle Marche, 2018-2019.
- [15] Renliang Xu, *Particle Characterization: Light Scattering Methods*, KLUWER ACADEMIC PUBLISHERS New York, Boston, Dordrecht, London, Moscow, 2002.
- [16] Chee F., Angelo S., Kiu S., Justin S. and Jackson C., *REAL TIME PARTICULATE MATTER CONCENTRATION MEASUREMENT USING LASER SCATTERING*, ARPN Journal of Engineering and Applied Sciences, 2018.
- [17] Xiaoyu Yu, Yunbo Shi, Tian Wang, Xu Sun, *Dust-concentration measurement based on Mie scattering of a laser beam*, Lakshminarayana Polavarapu, Ludwig-Maximilians-Universitat Munchen, GERMANY, 2017.
- [18] Jörg Stetefeld, Sean A. McKenna, Trushar R. Patel, *Dynamic light scattering: a practical guide and applications in biomedical sciences*, International Union for Pure and Applied Biophysics (IUPAB) and Springer-Verlag Berlin Heidelberg, 2016.

- [19] F. De Stefani. *Sistema di monitoraggio ambientale tramite WSN*. Master's thesis Università degli Studi di Pavia, 2008.
- [20] J. Bauwens, P. Ruckebusch, S. Giannoulis, I. Moerman, and E. D. Poorter. *Over-the-Air Software Updates in the Internet of Things: An Overview of Key Principles*. IEEE Communications Magazine, 58(2):35–41, 2020.