



**UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA**

Corso di Laurea in Ingegneria elettronica

**Studio e Sviluppo di un Sensore
NB-IoT**

Relatore:

Prof. Gambi Ennio

Tesi di laurea di:

Cipriani Laura

Correlatore:

Ing. De Santis Adelmo

A.A. 2019/2020

INDICE

Capitolo 1 Introduzione.....	5
Capitolo 2 Architettura di sistema	9
2.1 Scheda SODAQ SARA R412M	12
2.2 Sensori	15
2.2.1 Sensore MQ-137	15
2.2.2 Sensore DHT22.....	17
2.2.3 Sensore Laser SDS018	19
2.2.4 Sensore MH-Z14 NDIR a infrarossi.....	21
Capitolo 3 Tecnologia NB-IoT.....	23
3.1 LPWA.....	25
3.2 Caratteristiche NB-IoT	27
3.3 Struttura Frame	32
3.3.1 Downlink	32
3.3.2 Uplink	39
Capitolo 4 MQTT.....	46
Capitolo 5 Implementazione sketch Arduino.....	48
Capitolo 6 Conclusioni.....	57

CAPITOLO 1

Introduzione

Negli ultimi tempi sono nate sempre più discussioni relative all'inquinamento dell'aria (sia per quanto riguarda agenti inquinanti emessi dalle industrie sia per quelli normalmente prodotti dalla popolazione nella vita quotidiana) e in particolar modo viene spesso presa in considerazione la qualità dell'aria negli ambienti più frequentati, come ad esempio case, centri commerciali e uffici.

L'inquinamento dell'aria è definito come la presenza di agenti inquinanti in ambienti non industriali [1]. Questi agenti sono un rischio per la salute della popolazione e possono comprometterla sotto molti punti di vista: gli effetti infatti vanno dai più piccoli fastidi fino ad arrivare nei casi più estremi a causare seri problemi degli apparati respiratorio, cardiovascolare, gastro-intestinale ecc.

Secondo l'organizzazione mondiale della sanità (OMS) la popolazione, in media, passa il 90% della giornata in ambienti chiusi, di cui il 70% è spesso nel posto di lavoro e il 20% in casa, inoltre i dati raccolti dall'EPA (Environmental Protection Agency) dimostrano che gli agenti inquinanti in ambienti indoor sono 2-5 volte superiori che in quelli outdoor [1].

Gli effetti sull'uomo sono diversi a seconda del luogo e del tempo di esposizione ma dipendono anche dallo stato di salute del soggetto (eventuali allergie o patologie preesistenti), dall'età (i giovani sono meno soggetti a sviluppare effetti gravi), dalla classe sociale, dal sesso e anche dal tipo di nutrizione.

Gli studi che sono stati condotti hanno preso in considerazione diversi tempi di esposizione a seconda dell'agente inquinante valutato, ad esempio per l'anidride carbonica (CO₂) sono stati rilevati dati nel lungo periodo mentre per il diossido di azoto (NO₂) che presenta brevi picchi di valore elevato sono stati scelti tempi di esposizione ridotti.

Una delle cause principali e più diffuse dell'inquinamento dell'aria nelle abitazioni è l'umidità, questa è generata principalmente dalla condensazione, da uno scarso ricambio d'aria, o anche da infiltrazioni esterne.

Il problema principale legato all'umidità è il possibile sviluppo di muffe che rilasciano nell'aria circostante particelle inquinanti e tossiche per l'uomo che potrebbero portare all'insorgere di problemi respiratori: asma, bronchite e rinite allergica sono solo alcune delle possibili patologie causate dall'eccessiva presenza di umidità.

Quest'ultima inoltre può favorire la crescita di batteri e, si pensa, favorire la sopravvivenza di virus provenienti dall'esterno (teoria ad oggi non ancora ampiamente approvata) [2].

Quella dell'umidità però non è la sola causa dell'inquinamento dell'aria indoor, a essa se ne affiancano molte altre, come ad esempio la presenza di gas radon (derivato da materiali di costruzione e isolanti), le polveri sottili, l'anidride carbonica (CO₂), (dovuta ad esempio ad uno scarso ricambio d'aria di impianti di condizionamento) e i composti organici volatili (VOC). Questi ultimi sono definiti dalla legislazione italiana come "quei composti organici che alla temperatura di circa 20° hanno una pressione pari a 0,01 kPa o superiore".

La lista dei VOC è molto lunga, tra questi troviamo le sostanze presenti nei rivestimenti delle pareti e in particolare nelle vernici e nei solventi (acetone, etil acetato ecc.) ma nell'elenco si colloca anche il benzene il quale viene prodotto in larga parte dal fumo di tabacco [3].

La conseguenza della presenza di questi agenti inquinanti, come già ampiamente trattato, è il possibile sviluppo di patologie, per descrivere le situazioni più gravi (la maggior parte delle persone che frequentano luoghi con un livello di inquinamento dell'aria elevato presentano quasi subito qualche sintomo lieve) si parla di "*sindrome dell'edificio malato*":

Tale definizione indica un quadro sintomatologico ben definito, che si manifesta in un elevato numero di occupanti di edifici moderni o recentemente rinnovati, dotati di impianti di ventilazione meccanica e di condizionamento d'aria globale (senza immissione di aria fresca dall'esterno) e adibiti a uffici, scuole, ospedali, case per anziani, abitazioni civili.

Le manifestazioni cliniche sono aspecifiche, insorgono dopo alcune ore di permanenza in un determinato edificio e si risolvono in genere rapidamente, nel corso di qualche ora o di qualche giorno (nel caso di sintomi cutanei) dopo l'uscita dall'edificio. Sebbene i sintomi siano di modesta entità, i casi di SBS (sick building syndrome) che si verificano in ambienti lavorativi possono portare allo sviluppo di alcune malattie gravi e a prognosi peggiore.

Da numerose indagini in edifici in cui sono stati segnalati problemi di salute o di comfort è emerso che il problema prevalente (in quasi la metà dei casi) era costituito da una ventilazione inadeguata; alcuni studi condotti su uffici e altri edifici ad uso pubblico in diversi paesi, hanno rivelato una frequenza di disturbi tra gli occupanti compresa tra il 15% e il 50% [4].

A queste preoccupazioni però se ne è da poco aggiunta un'altra, generata dal diffondersi del virus Covid-19, il quale è in grado di trasmettersi molto rapidamente da un individuo all'altro per via aerea, è per questa ragione dunque che il miglior modo per proteggersi è quello di mantenere una certa distanza di sicurezza, indossare sempre la mascherina e non affollare luoghi chiusi.

Per questi motivi il progetto che ho scelto di portare avanti è costituito da una board principale connessa a 4 sensori che sono in grado di determinare la quantità di alcuni gas presenti nell'ambiente, tali valori vengono poi inviati tramite protocollo mqtt e studiati per valutare la qualità dell'aria in una determinata situazione. Ho infine effettuato misure in uno stesso ambiente con una o più persone per determinare come variano i valori dei gas esaminati a seconda del numero di persone presenti.

Quest'ultima valutazione si può rivelare utile in ambienti che sono molto frequentati in una situazione di emergenza come quella che stiamo attualmente vivendo, come ad esempio i supermercati, nei quali il numero di presenti deve trovarsi al di sotto di una soglia prestabilita per assicurare il rispetto delle normative imposte per la tutela sia dei consumatori che dei dipendenti. Infatti se i valori di alcuni gas sono elevati essi fungono da campanello d'allarme, ad esempio se in una stanza ci sono troppe persone o non c'è un corretto ricircolo d'aria il valore di CO₂ aumenta sensibilmente.

Il miglior modo per garantire uno spazio sicuro sotto tutti questi punti di vista, è quindi quello di monitorare costantemente i valori dei principali agenti inquinanti dell'aria (sia prodotti dai macchinari sia prodotti dall'uomo). Questo può essere fatto ad esempio tramite dei piccoli sensori integrati su schede elettroniche progettate in modo che siano in grado di inviare periodicamente i dati raccolti ad un secondo dispositivo.

Nel caso specifico la scheda utilizzata è la SODAQ SARA R412M con integrati quattro sensori (Sensore Laser di Polvere e Particelle SDS018, Sensore CO2 MH-Z14 NDIR a infrarossi, Sensore temperature umidità DHT22/AM2302, Sensore di VOC MQ-137) e l'invio dei dati avverrà tramite protocollo mqtt utilizzando il broker online "CloudMQTT".

CAPITOLO 2

Architettura di Sistema

Il sistema è basato su diversi componenti che sono stati assemblati all'interno di una scatola. In seguito saranno descritte le connessioni elettriche e dati realizzate, e così anche il circuito.

I blocchi che costituiscono il prototipo sono:

- Alimentatore;
- Modulo fusibili e regolatori di tensione;
- Sensore PM;
- Sensore CO₂;
- Sensore qualità aria;
- Sensore temperatura ed umidità;
- Scheda SODAQ SARA.
- Ventola

Sensore	Agenti rilevati
Sensore Laser SDS018	Polveri sottili e particelle
Sensore MH-Z14 NDIR a infrarossi	CO ₂
Sensore DHT22/AM2302	Temperatura, umidità
Sensore MQ-137	NH ₃ , NO _x , alcool, Benzene, fumo

Alimentatore

Alimentatore in tensione continua realizzato in tecnologia switching, modello RS-25-12. Fornisce una tensione di uscita continua a 12V con corrente massima di 2.1A.

Usare un alimentatore a commutazione consente di ridurre il peso e l'ingombro del prototipo e la sua dissipazione termica. Gli alimentatori switching dissipano pochissima potenza se non hanno carico. Per questo è stato previsto un interruttore solo sulla sezione a bassa tensione e non sulla 220V, in questo modo, se il prototipo dovrà essere aperto per cambiare fusibili o altre operazioni, il rischio di elettrocuzione è molto ridotto.

La tensione di uscita a 12V ha richiesto l'installazione di regolatori di tensione per ottenere i seguenti valori:

- 5V per alimentare i sensori e la scheda SODAQ SARA;
- 5V lineari per alimentare il sensore PM;

L'assorbimento in corrente (massimo) dei moduli da alimentare è il seguente:

- Sensore PM: 200mA (1W a 5V);
- Sensore CO2: 150mA
- Sensore temperatura ed umidità: 2.5mA
- Sensore qualità aria: 200mA
- SODAQ SARA: 900mA

Corrente massima circa 1450mA a 5V.

Modulo fusibili e regolatori

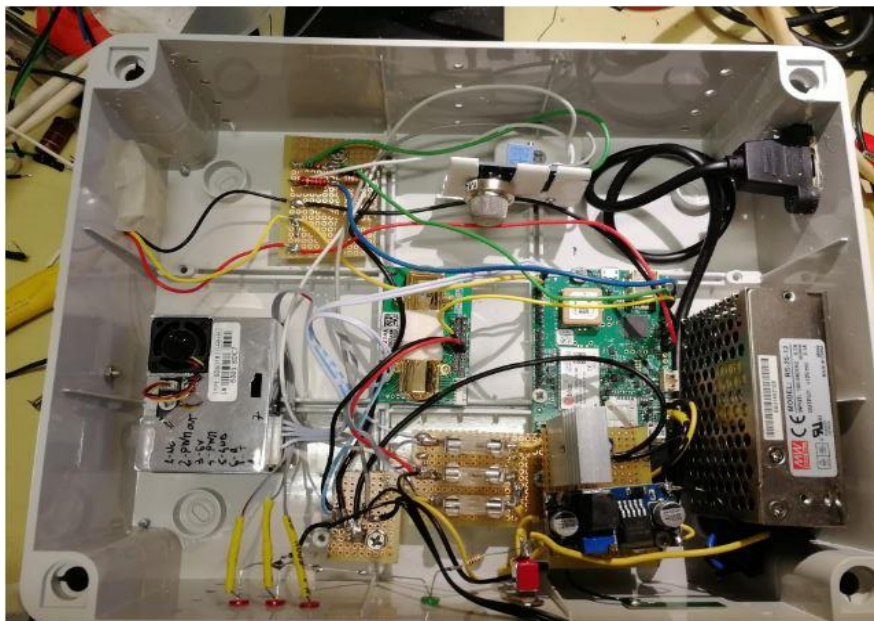
La tensione in uscita dall'alimentatore viene inviata ad un interruttore e quindi a dei regolatori di tensione. Sono utilizzati tre diversi regolatori:

- LM7805: regolatore lineare, deve essere dotato di dissipatore.
- Regolatore switching LM2594: basato sull'integrato omonimo, non necessita di dissipazione;

I regolatori switching basati su LM2594 hanno un ripple in uscita che può arrivare a 100mV. Tale valore non è compatibile con quanto richiesto dal misuratore PM che richiede un massimo ripple di 20mV. È stato dunque necessario utilizzare un regolatore lineare e dotarlo di un dissipatore di calore. La potenza dissipata dal regolatore è pari alla caduta di tensione per la corrente: $7V * 200mA = 1.4W$. Occorre prevedere dei fori per il raffreddamento del dissipatore, sfruttando la circolazione di aria della ventola.

I LED rossi sono installati a valle dei fusibili.

- D1 – SODAQ SARA -> Fusibile 1A;
- D2 – Sensore CO2 – MQ-137 – Ventola -> Fusibile 500mA
- D3 – Sensore PM -> Fusibile 500mA



figural

La parte hardware (figura 1) si occupa dunque di rilevare i valori dei vari agenti chimici e dell'elaborazione dei dati raccolti per determinare la qualità dell'aria e valutare come quest'ultima cambia all'aumentare o al diminuire del numero di persone presenti in una stanza. Tramite il protocollo di comunicazione MQTT (MQ Telemetry Transport) i dati vengono poi inviati al broker (Cloud MQTT) che si occupa della loro visualizzazione.

2.1 SCHEDA SODAQ SARA R412M

La scheda (figura 2) è Arduino M0 compatibile e supporta entrambe le tecnologie NB-IoT e LTE-M, e dunque l'uso delle bande di frequenze ad esse legate[5].

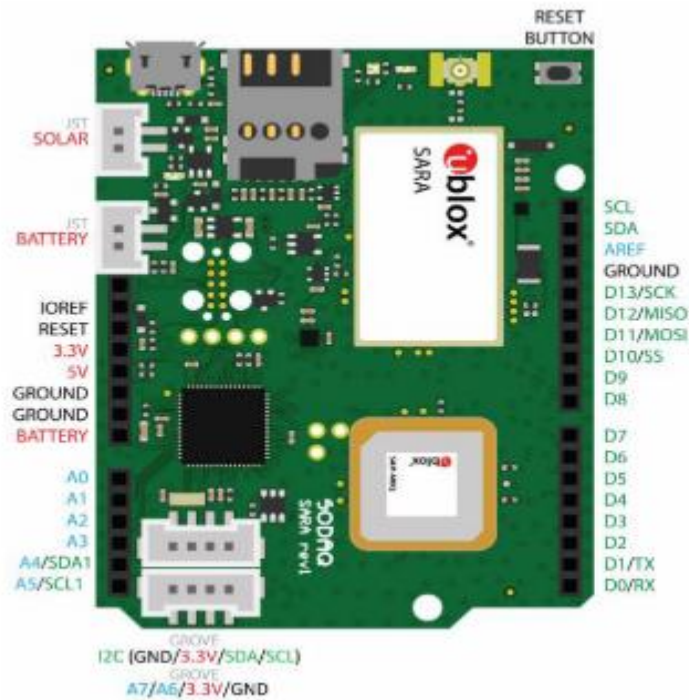


figura 2

I moduli della serie SARA-R4 forniscono una configurabilità multi-banda basata su software, consentendo una copertura multiregionale internazionale nelle tecnologie di accesso radio LTE Cat M1 / NB1 e (E)GPRS. I moduli della serie SARA-R4 offrono comunicazioni dati fino a 375 kb/s su un intervallo di temperatura operativa compreso tra -40 e 85 gradi centigradi, con basso consumo energetico e con un miglioramento della copertura.

Questa scheda offre ottime prestazioni per le applicazioni LPWA (Low Power Wide Area) utilizzata in questo progetto, che permettono l'invio di dati ad una velocità medio-bassa in un'area relativamente estesa senza produrre un eccessivo consumo di batteria.

Interfacce

In figura 3 la rappresentazione delle varie interfacce:

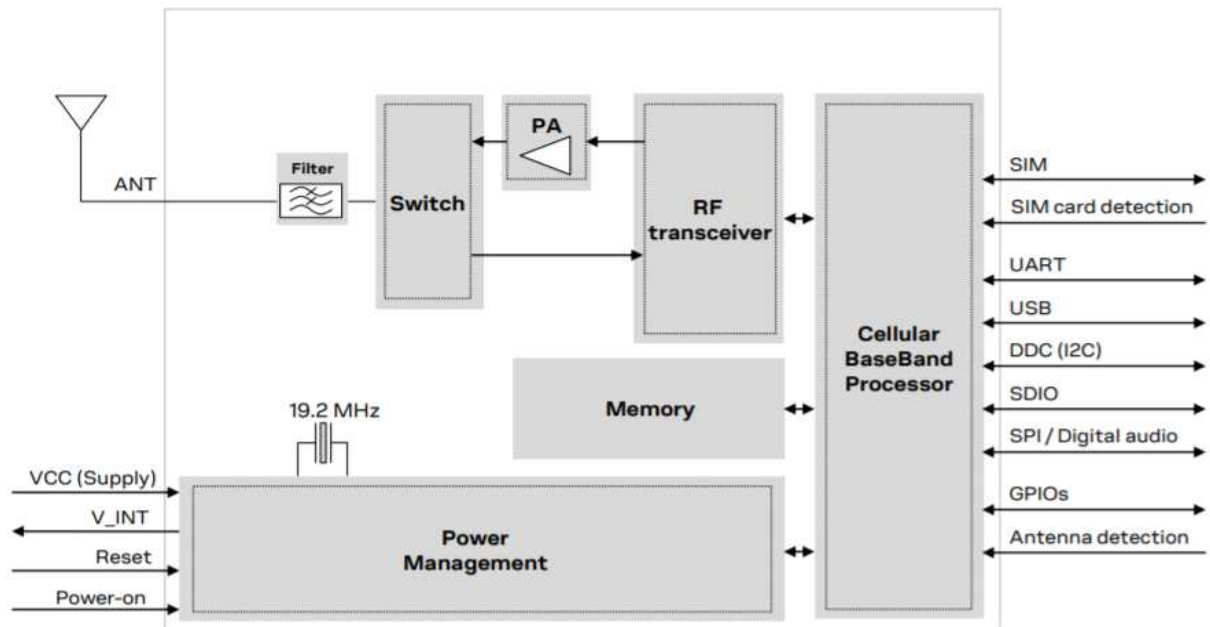


figura 3

-Alimentazione

I moduli della serie SARA-R4 devono essere alimentati tramite i pin VCC da un alimentatore DC e la tensione deve essere stabile, perché durante il funzionamento la corrente prelevata da VCC può variare.

La scheda può essere alimentata in molti modi diversi:

- Tramite USB;
- Tramite Batteria;
- Tramite pannello solare.

Per rendere più agevole il posizionamento della scheda è stato usato il connettore “solar” in quanto, viene specificato che “The solar input is 5V. You can connect any other 5V device to charge the battery”.

-Antenne

Il pin ANT rappresenta l'interfaccia dell'antenna RF del modulo, con una impedenza caratteristica di 50 Ω ;

Il pin ANT_DET è un ingresso ADC (convertitore analogico-digitale) attraversato dalla corrente fornita dai moduli della serie SARA-R4 per rilevare la presenza dell'antenna e valuta inoltre la resistenza dal pin ANT al GND tramite un circuito di rilevamento dell'antenna esterno implementato sulla scheda dell'applicazione.

-Power on

La scheda può essere accesa con:

Livello basso sul pin PWR_ON, che normalmente è impostato alto da un pull-up interno, per un periodo di tempo valido quando la tensione VCC applicata è all'interno dell'intervallo operativo previsto (min. 3.2V max. 4.0V).

-Power off

La serie SARA-R4 può essere disattivata correttamente, in uno dei modi seguenti:

- Comando AT-CPWROFF
- Basso impulso sul pin PWR_ON per un periodo di tempo valido

Si verifica un arresto improvviso sui moduli della serie SARA-R4, senza archiviazione delle impostazioni dei parametri correnti e senza una rete pulita quando:

- L'alimentazione VCC scende al di sotto del limite minimo dell'autonomia operativa
- Viene applicato un livello basso sul perno RESET_N, che normalmente viene impostato alto da un pull-up interno, per un periodo di tempo valido

2.2 SENSORI

2.2.1 Sensore MQ-137



figura 4

Il sensore sopra rappresentato (figura 4, figura 5), è un sensore elettrochimico per la rilevazione della qualità dell'aria. I sensori di questa serie hanno tutti lo stesso schema di principio: sono composti da un riscaldatore e da un elemento sensibile che ha uscita analogica.

Esso viene tipicamente fornito su un piccolo PCB, sul quale è realizzata l'uscita digitale utilizzando un comparatore con soglia, per questo progetto però verrà usata solo l'uscita analogica.

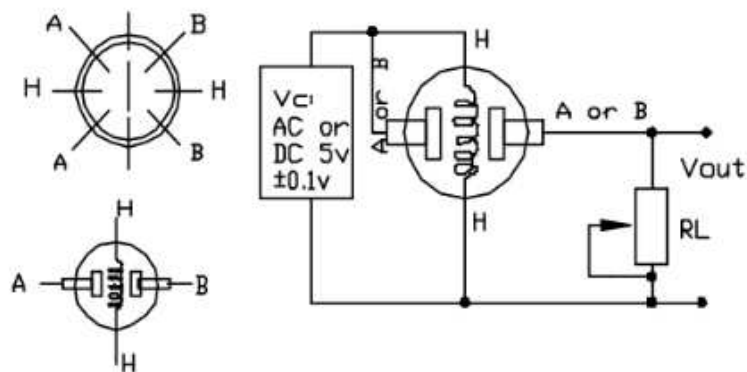


figura 5

L'uscita analogica ha uno swing massimo di 5V [6] che può essere ridotto utilizzando un partitore di tensione (figura 6) realizzato a valle di V_{out} (in parallelo ad R_L). Per le nostre esigenze è stato realizzato con due resistori di valore 180k e 100k.

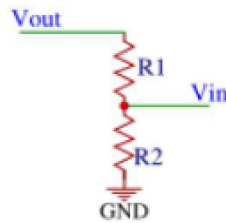


figura 6

L'uscita del sensore è collegata al PIN A1 della scheda SARA. Con questi valori del partitore, la tensione massima è pari a 3.18V.

in grado di rilevare ammonio, anidride carbonica, monossido di carbonio, etanolo, toluene, acetone, presenta 6 pin, 4 dei quali sono utilizzati per la rilevazione dei segnali e 2 per fornire corrente.

Di seguito (figura 7) è riportato il grafico del comportamento del sensore rispetto a variazioni di temperatura e umidità. Notare che R_o è la resistenza del sensore a 100 ppm di NH_3 in aria pulita, R_s è la resistenza del sensore a 100 ppm di NH_3 ma a differenti condizioni di temperatura e umidità, RH è l'umidità relativa (relative humidity).

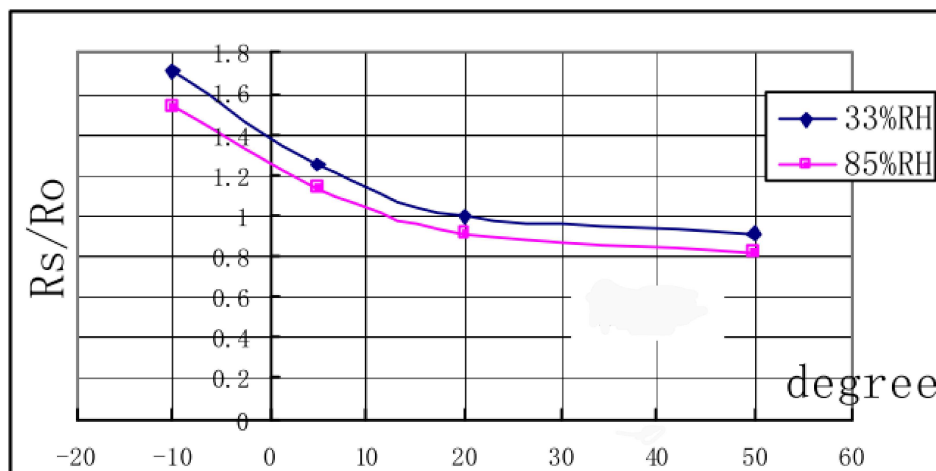


figura 7

2.2.2 Sensore DHT22

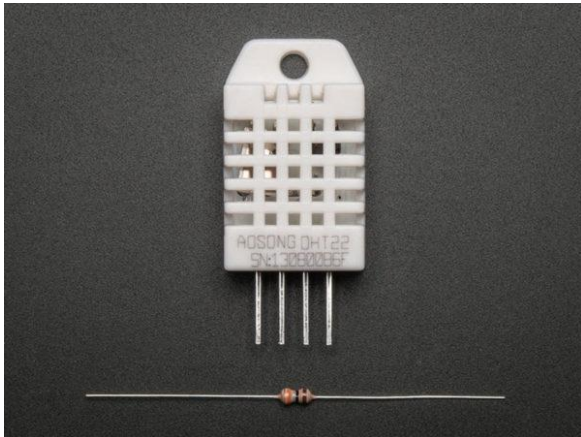


figura 8

Il sensore DHT22 (figura 8) utilizzato per il rilevamento dell'umidità e della temperatura e presenta 4 pin (dei quali solo 3 sono collegati, 2 per l'alimentazione e 1 per i dati):

PIN	FUNZIONE
1	Tensione di alimentazione a 5V
2	Pull-up verso i 5V con resistore da 10k e connessione al pin D2 della scheda SARA
3	Non collegato
4	GND

I suoi elementi di rilevamento sono collegati a un single chip computer a 8 bit.

La tensione dell'alimentazione deve essere 3,3-6V DC [7] e quando l'alimentazione viene fornita, non deve essere inviata alcuna istruzione al sensore per un secondo per passare lo stato instabile.

Quando MCU invia il segnale di avvio, DHT22 passa dalla modalità a basso consumo energetico alla modalità di esecuzione. Quando MCU termina l'invio del segnale di avvio, DHT22 invierà un segnale di risposta di dati a 40 bit che riflettono l'umidità relativa.

Il sensore di umidità e pressione utilizzato è un DHT22 in versione con “enclosure” (figura 9). Si tratta di un guscio plastico che protegge il sensore, rendendolo idoneo alle installazioni che potrebbero degradarne il funzionamento.

La comunicazione avviene con un protocollo “single wire” che rende molto semplice il collegamento al micro. Tale linea richiede un “pull-up” realizzato con un resistore da 10k verso Vcc.



figura 9

A causa della lentezza del dispositivo però, non è possibile richiedere i dati con una frequenza maggiore di 0.5Hz (una richiesta ogni 2 secondi).

2.2.3 Sensore Laser SDS018



figura 10

Questo dispositivo (figura 10) sfrutta il principio di dispersione laser:

la dispersione della luce può essere indotta quando le particelle passano attraverso l'area di rilevamento, la luce diffusa viene trasformata in segnali elettrici e questi segnali saranno amplificati ed elaborati. Il numero e il diametro delle particelle possono essere ottenuti mediante analisi perché la forma d'onda del segnale ha determinate relazioni con il diametro delle particelle. Grazie a questo principio può ottenere la concentrazione delle particelle tra 0,3 e 10 m nell'aria.

Potenze richieste:

- Power Voltage : 4.7~5.3V
- Power supply : >1W
- Supply voltage ripple : <20mV

La vita utile è il parametro chiave del sensore di polvere laser: il diodo laser in questo sensore ha alta qualità e può durare fino a 8000 ore. Se è necessario il rilevamento di dati in tempo reale, è possibile utilizzare la configurazione predefinita che misura alla frequenza di 1 volta al secondo.

Nel caso in cui la domanda in tempo reale non sia elevata (come per filtro, monitoraggio della qualità dell'aria, ecc), è possibile utilizzare il metodo di lavoro discontinuo per prolungare la vita utile (ad esempio, è possibile avviare il sensore per 30 secondi al minuto).

Il sensore particolato è una unità che deve essere alimentata in continua, con un valore di tensione di 5V. Richiede un ripple massimo di 20mV, specifica piuttosto stringente e che deve essere verificata con l'ausilio di un oscilloscopio [8]. Sul sensore è installato un connettore a 7 poli, la cui piedinatura è riportata in figura 11.

No	Name	Comment
1	NC	Not Connect
2	1 μ m	PWM output, see for more details
3	5V	5V Input
4	2.5 μ m	PWM output, see for more details
5	GND	Ground
6	R	RX of UART (TTL) @3.3V
7	T	TX of UART (TTL) @3.3V

figura 11

Nell'unità inserita in questo progetto sono utilizzati solo una parte dei segnali disponibili:

- 3 – 5V input;
- 5 – Ground;
- 6 – UART RX, compatibile 3.3V
- 7 – UART TX, compatibile 3.3V

Le possibilità di interfacciamento del sensore sono quindi limitate alla UART, ovvero all'utilizzo di un protocollo seriale. Per consentire l'interfacciamento con la SODAQ SARA sono stati eseguiti i seguenti collegamenti:

- Pin 6 del sensore (RX) -> pin D1 SARA (TX);
- Pin 7 del sensore (TX) -> pin D0 SARA (RX);

I due dispositivi hanno già la massa in comune, pertanto non è necessario eseguire ulteriori collegamenti.

2.2.4 Sensore MH-Z14 NDIR a infrarossi

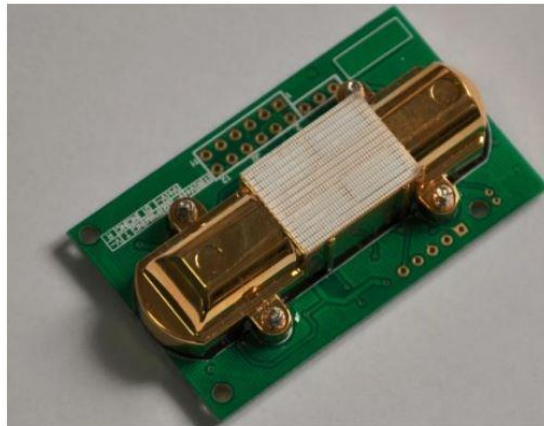
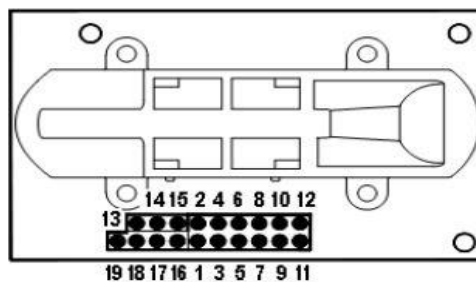


figura 12

L'MH-14 NDIR (figura 12) è un sensore di tipo comune, di piccole dimensioni, che utilizza il principio dell'infrarosso non dispersivo (NDIR) per rilevare l'esistenza di CO₂ nell'aria, con buona selettività, non dipendente dall'ossigeno e caratterizzato da una lunga durata [9]. Ha un sensore di temperatura integrato in grado di effettuare la compensazione della temperatura.

Definizione dei pin:



PIN	Description
Pad1/Pad15/Pad17	Vin (input voltage 4.5V~5.5V)
Pad2/Pad3/ Pad12/Pad16	GND
Pad4	Vout2 (0.4~2V)
Pad5	Vout1 (0~2.5V)
Pad6	PWM
Pad8	HD
Pad7/Pad9	NC
Pad11/Pad14/Pad18	UART (RXD) 0~3.3V input digital
Pad10/Pad13/Pad19	UART (TXD) 0~3.3V output digital

figura 13

Il sensore CO2 ha una forma molto compatta e presenta due connettori per l'interfacciamento. I pin dei connettori sono chiamati PAD ed il loro pinout è riportato in figura 13.

L'interfacciamento con la scheda SARA è stato fatto utilizzando l'uscita analogica e connettendola al pin A0.

Tale GPIO dovrà quindi essere configurato come ingresso e su questo dovrà essere abilitato l'ADC. Notiamo infine come lo swing della tensione di ingresso è tra 0.4V e 2V, pertanto al di sotto della soglia di "sicurezza" del dispositivo.

Le connessioni effettuate sono

- PAD 1 – tensione di ingresso a 5V;
- PAD 2 – GND;
- PAD 4 – al pin A0 della scheda SARA.

CAPITOLO 3

Tecnologia NB-IoT

Il narrowband IoT (NB-IoT) è un nuovo standard di tecnologia cellulare 3GPP in rapida crescita introdotto nella Release 13 che risponde ai requisiti LPWAN (Low Power Wide Area Network) dell'IoT [10].

È stata classificata come tecnologia 5G, standardizzata da 3GPP nel 2016 e sta rapidamente emergendo come la migliore tecnologia LPWAN per consentire una vasta gamma di nuovi dispositivi IoT, tra cui parcheggio intelligente, utility, dispositivi indossabili e soluzioni industriali.

NB-IoT è caratterizzato da:

- eccellente copertura interna,
- supporto di un numero enorme di connessioni,
- efficienza dei costi,
- basso consumo energetico del dispositivo,
- architettura di rete ottimizzata.

Questa tecnologia è in grado di collegare in modo efficiente grandi numeri di dispositivi, fino a 50.000 per cella di rete NB-IoT, riducendo al minimo il consumo energetico e aumentando la portata di copertura in luoghi non serviti dalle tecnologie cellulari convenzionali. La batteria può durare fino a 10 anni (molto più che con le altre tecnologie) pur mantenendo un rate di trasmissione dati elevato (200 byte al giorno).

Poiché NB-IoT è uno standard aperto, l'end device funziona con differenti operatori e frequenze, supporta la trasmissione di dati IP e non è molto esoso in termini di potenza del segnale ricevuto.

La rete di accesso di NB-IoT rimane la stessa della rete LTE legacy con la differenza principale che il nodo NB-IoT contenderà sempre il canale uplink e quindi stabilirà la sessione, mentre l'end device sarà in modalità di connessione in LTE legacy.

3.1 LPWA

LPWA è un termine generico per un gruppo di tecnologie che consentono comunicazioni su vasta area con costi inferiori e un ridotto consumo energetico. È in particolare adatto per le applicazioni IoT che hanno bisogno di trasmettere piccole quantità di informazioni in un lungo raggio.

La larghezza di banda molto limitata delle reti LPWA non è adatta per la maggior parte delle applicazioni commerciali come voce, video, audio o anche messaggi di testo. Di conseguenza, le reti LPWA vengono utilizzate quasi esclusivamente dai dispositivi IoT (Internet of Things) e M2M (machine-to-machine).

La maggior parte dei dispositivi IoT, in particolare quelli nei settori delle città intelligenti e industriali (frigoriferi, lampadine ecc.), non richiedono la stessa velocità e larghezza di banda dei dispositivi cellulari. Tuttavia, hanno bisogno della longevità delle reti cellulari LTE tradizionali.

Le tecnologie LPWAN sono ideali per:

- Contatori intelligenti;
- Smart city;
- Agricoltura intelligente;
- Applicazioni di costruzione intelligenti

La tecnologia LPWAN supporta il trasferimento dei dati in piccoli pacchetti intermittenti di dimensioni comprese tra 10 e 1000 byte. Ciò consente una maggiore efficienza e velocità ottimizzate che vanno da 3 Kbps a 375 Kbps.

Poiché le reti LPWA funzionano con una maggiore efficienza di potenza e larghezza di banda e su un'area più ampia, sono necessari meno infrastruttura e hardware. Queste caratteristiche si traducono in una maggiore efficienza dei costi.

Caratteristiche principali:

- Lungo raggio d'azione

Le tecnologie LPWA sono progettate per una vasta copertura di area e un'eccellente propagazione del segnale in luoghi interni difficili da raggiungere (come gli scantinati). Quantitativamente si stima un guadagno di 20 dB rispetto ai sistemi cellulari legacy. Ciò consente ai dispositivi finali di connettersi alle stazioni di base a una distanza che va da alcuni fino a decine di chilometri a seconda dell'ambiente di distribuzione

Le tecniche di modulazione a banda stretta forniscono un budget di collegamento elevato codificando il segnale in larghezza di banda ridotta (di solito inferiore a 25 kHz). Assegnando ad ogni supporto una banda molto stretta, queste tecniche di modulazione condividono lo spettro complessivo in modo molto efficiente tra più collegamenti. Anche il livello di rumore sperimentato all'interno di una singola banda stretta è minimo.

- Basso consumo

La tecnologia LPWAN consente ai dispositivi IoT di funzionare in modo affidabile per un massimo di 10 anni con una singola carica della batteria.

Questa durata è ideale per soluzioni con mobilità limitata che non dispongono di una fonte di alimentazione affidabile per la ricarica.

- Basso costo

Il successo commerciale delle reti LPWA è legato alla connessione di un gran numero di dispositivi finali, mantenendo il costo dell'hardware al di sotto di 5 dollari.

- Scalabilità

Il supporto per un numero enorme di dispositivi che inviano volumi di traffico bassi è uno dei requisiti chiave per le tecnologie LPWA. Queste utilizzano la comunicazione multicanale e multi-antenna per parallelizzare le trasmissioni da e verso i dispositivi collegati. Inoltre, la comunicazione viene resa resiliente alle interferenze utilizzando più canali ed eseguendo trasmissioni ridondanti.

3.2 CARATTERISTICHE NB-IoT

Nell'ambito della release 13 di 3GPP sono state presentate le tre principali tecnologie per l'IoT a seconda dei requisiti da soddisfare e dei mercati su cui si collocano:

- NB-IoT (*NarrowBand Internet of Things*), rappresenta una soluzione basata su una nuova interfaccia radio, che può essere utilizzata sia in una porzione della banda del segnale LTE (o nella sua banda di guardia), o in porzioni di spettro rese disponibili dal rilascio di frequenze.
- LTE-M (*Long Term Evolution - Machine-Type Communications*), rappresenta un'evoluzione di quanto iniziato a definire nell'ambito della Release 12 del 3GPP, in termini di MTC (*Machine-Type Communications*) in una rete LTE, con l'introduzione di una categoria specifica per i terminali, denominata Cat-0. LTE-M è pertanto noto anche con l'acronimo eMTC (*enhanced-MTC*) e per i terminali è stata introdotta una nuova categoria, denominata Cat-M1.
- EC-GSM-IoT (*Extended Coverage GSM IoT*), rappresenta la soluzione compatibile con una rete GSM/EDGE, di cui riutilizza una porzione della banda e che richiede la disponibilità dell'EGPRS in rete.

Sviluppo delle tecnologie legacy (figura 14) :

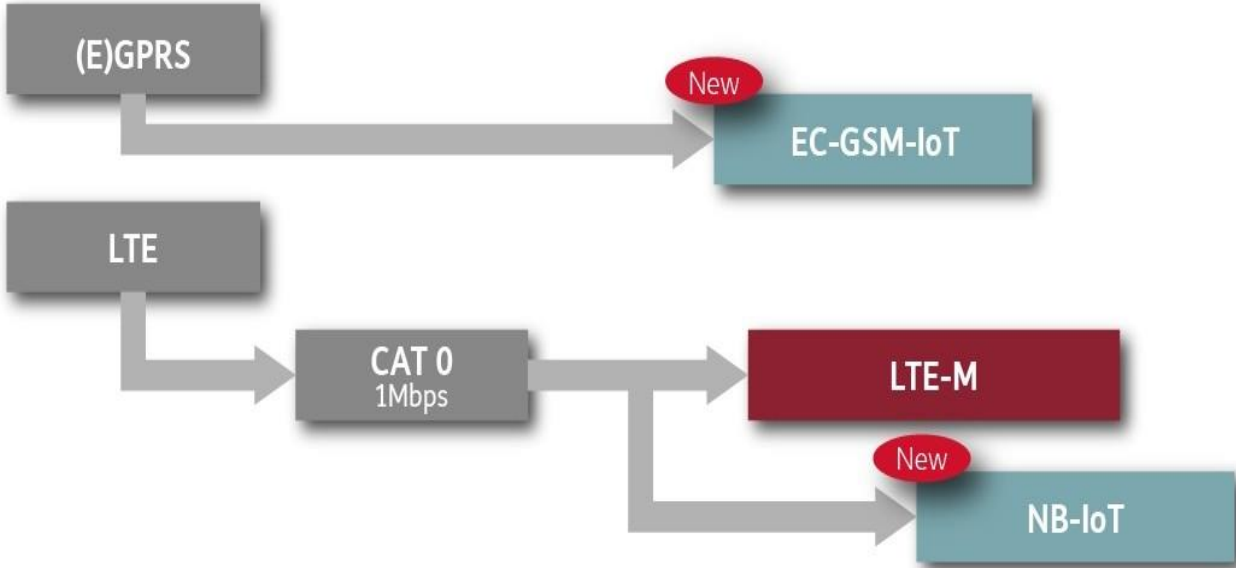


figura 14

NB-IoT nasce come una nuova tecnologia radio (figura 15), tuttavia utilizzabile:

- in una rete LTE nella sua banda utile di dispiegamento “*in-band deployment*”, mediante l’utilizzo di una o più porzioni di spettro da 180 kHz, dette PRB (Primary Resource Block), allocate nella banda utile di LTE
- nella sua banda di guardia “*guard-band deployment*”, mediante l’utilizzo di uno o più PRB di 180 kHz allocati nella banda di guardia di LTE
- in porzioni di spettro rese comunque disponibili di una rete GSM ancora dispiegata in campo, “*stand-alone deployment*”, mediante l’utilizzo di uno o più canali di 200 kHz nominali (180 kHz effettivi).

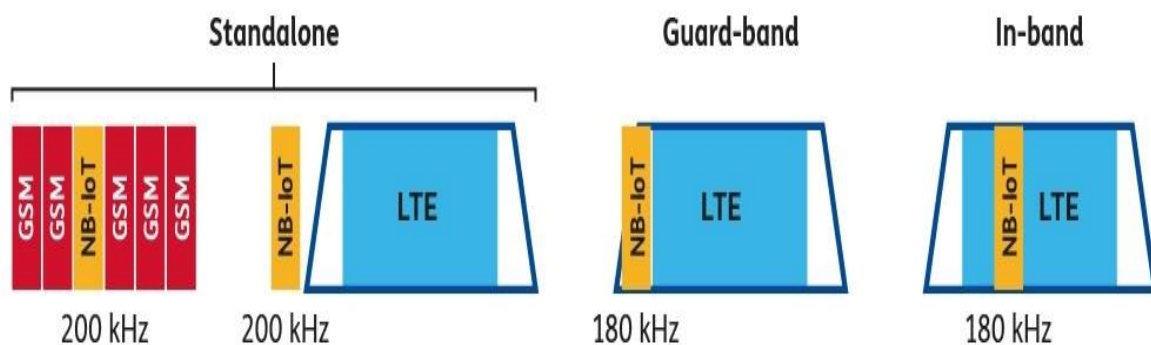


figura 15

Il 3GPP ha definito le seguenti bande di frequenza per NB-IoT nella versione 13:

1, 2, 3, 5, 8, 12, 13, 17, 18, 19, 20, 26, 28 e 66, (vedi tabella) e il limite di connessioni alla stessa cella di 52547. Per il mercato europeo le bande supportate sono:

- B3 (1800 MHz);
- B8 (900 MHz);
- B20 (800 MHz).

In Italia vengono utilizzate le bande B8 e B20.

<i>Numero di Banda</i>	Range di frequenze Uplink [MHz]	Range di frequenze Downlink [MHz]
<i>1</i>	1920 - 1980	2110 - 2170
<i>2</i>	1850 - 1910	1930 - 1990
<i>3</i>	1710 - 1785	1805 - 1880
<i>5</i>	824 - 849	869 - 894
<i>8</i>	880 - 915	925 - 960
<i>12</i>	699 - 716	729 - 746
<i>13</i>	777 - 787	746 - 756
<i>17</i>	704 - 716	734 - 746
<i>18</i>	815 - 830	860 - 875
<i>19</i>	830 - 845	875 - 890
<i>20</i>	832 - 862	791 - 821
<i>26</i>	814 - 849	859 - 894
<i>28</i>	703 - 748	758 - 803
<i>66</i>	1710 - 1780	2110 - 2200

L'NB-IoT è caratterizzato da una banda estremamente ridotta (180 kHz) e facilmente scalabile al crescere del traffico IoT (con allocazioni multiple di canali da 180 kHz) e offre inoltre il miglior MCL (*Maximum Coupling Loss*) rispetto a tutte le altre tecnologie LPWAN: circa 164 dB sia in uplink che in downlink (20 dB superiore a una rete legacy GPRS).

Questa estensione è ottenuta tramite diverse tecniche radio, infatti è prevista una concentrazione della potenza trasmessa su una banda estremamente ridotta, che genera quindi un incremento della PSD (*Power Spectral Density*), vengono inoltre ripetuti per un elevato numero di volte i TB (*Transport Block*): si considerano fino a 2048 ripetizioni in DL e 128 in UL, valori selezionati in modo da poter raggiungere, insieme alle altre funzionalità, i 20 dB di guadagno attesi sulla massima estensione della copertura radioelettrica rispetto al sistema GPRS.

Infine l'impiego di schemi di modulazione efficienti permette di ridurre il PAPR (*Peak-to-Average Power Ratio*), consentendo di utilizzare l'amplificatore in un punto di lavoro prossimo alla saturazione, senza introdurre distorsioni.

La potenza di trasmissione del terminale è impostata a 23 dBm oppure a 20 dBm, valori tali da consentire l'integrazione dell'amplificatore di potenza nel SoC (*System-on-Chip*); a titolo comparativo, un terminale GPRS trasmette a 33 dBm, ossia ad un valore di almeno 10 dB superiore, pur raggiungendo una copertura radioelettrica di 20 dB inferiore rispetto a NB-IoT.

L'NB-IoT non prevede requisiti stringenti in termini di latenza, non presenta comunque ritardi superiori ai 10 secondi nel caso di applicazioni che richiedono l'invio di allarmi da parte di dispositivi collocati in luoghi tali da richiedere la massima estensione di copertura radioelettrica di 20 dB. Il ritardo considerato è valutato tra l'istante in cui si verifica l'evento che determina la segnalazione d'allarme e l'istante in cui tale segnalazione è disponibile alla stazione base per essere inviata alla core network.

Tale tecnologia infine permette di allungare la durata della batteria fino a 10 anni sfruttando alcune funzionalità radio:

- eDRX (*extended Discontinuous Reception*), che consente di ridurre la frequenza con cui sono monitorati i canali di controllo in DL da parte del terminale ed il numero di report di misura inviati dallo stesso terminale alla rete;
- PSM (*Power Saving Mode*), che consente di minimizzare il consumo di potenza da parte del terminale che si trova in tale modalità, in cui il esso risulta comunque irraggiungibile, pur essendo ancora registrato alla rete; il terminale esce da tale modalità con procedure periodiche originate dallo stesso (es. TAU "*Tracking Area Update*").

3.3 STRUTTURA FRAME

Per parlare della struttura del frame è importante distinguere l'invio di dati Uplink e Downlink, infatti l'NB-IoT funziona in modalità HD-FDD, le trasmissioni uplink e downlink utilizzano risorse di frequenza diverse e sono separate da almeno un subframe per alternare le funzioni di trasmissione e ricezione.

Una caratteristica che invece accomuna UL e DL è il data rate ridotto, dell'ordine di alcune decine di kbps, con valori di picco pari a 250 kbps in UL ed a 170/226.7 kbps in DL in in-band/stand-alone deployment.

Gli **slot**, i **sottoframe** e i **frame** hanno durata rispettivamente di **0.5 ms, 1 ms e 10 ms** e sono identici a quelli dell'LTE, inoltre anche il formato dello slot in termini di prefisso ciclico, durata e numero di simboli OFDM è identico all'LTE.

3.3.1 Downlink

Lo Scheduling downlink è simile a quello della rete LTE legacy nel dominio del tempo e ha durata di 10ms, ogni frame è costituito da 10 sottoframe di lunghezza 1 ms e ogni sottoframe è costituito da due slot con una lunghezza di sette simboli OFDM.

Nel dominio della frequenza, è costituito da un PBR con 12 sottoportanti spaziate l'un l'altra di 15 kHz e con prefisso ciclico normale (CP). Una sottoportante e un simbolo costituiscono un resource element (RE), la più piccola unità di trasmissione.

Inoltre, a differenza di LTE, i segnali e i canali fisici downlink NB-IoT sono prima multiplati nel tempo. Nella figura 16 è illustrato il modo in cui i sottoframe NB-IoT vengono allocati in diversi canali fisici e segnali in downlink.

NB-IoT in downlink ha due segnali fisici e tre canali fisici:

- NB Physical Broadcast Channel (NPBCH);
- NB Physical Downlink Shared Channel (NPDSCH);
- NB Physical Downlink Control Channel (NPDCCH);
- NB Primary and Secondary Synchronization Signal (NPSS - NSSS);
- NB Reference Signal (NRS);

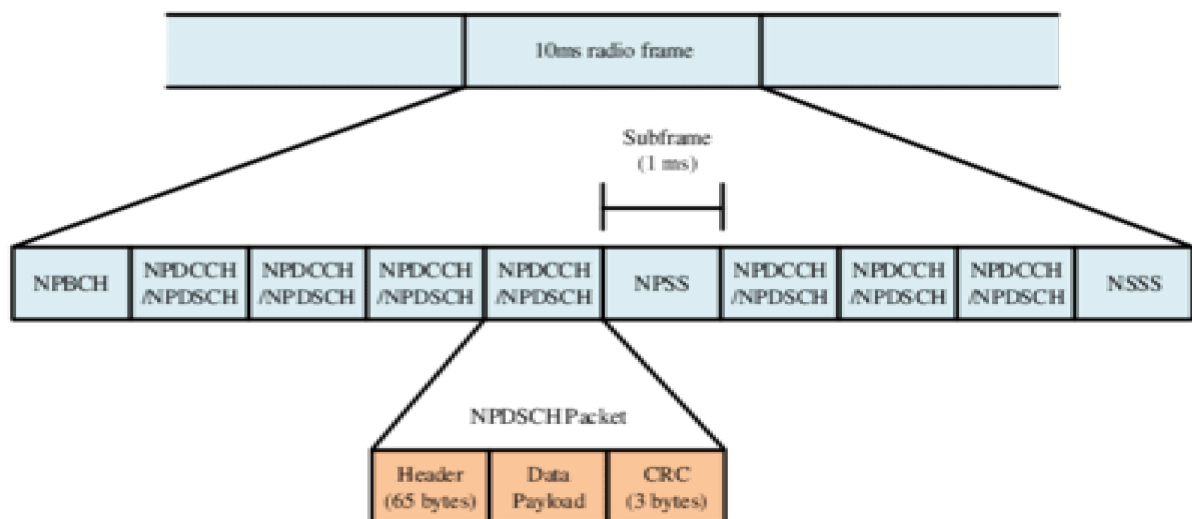


figura 16

Primary Resource Block

NB-IoT è parte integrante della rete LTE Advanced Pro, uno dei blocchi LTE a 180 kHz è dedicato per NB-IoT. Il PRB di NB-IoT da 180 kHz viene diviso in 12 sottoportanti, ogni sottoportante è divisa in due sottoframe ognuno dei quali ospita 7 simboli OFDMA ed utilizza la modulazione QPSK.

La struttura del frame per il canale downlink è simile a quella dell'LTE legacy ma con meno parametri per far fronte ai requisiti di snellezza di IoT. Ogni frame radio ha durata 10 ms e comprende 10 sottoframe da 1 ms ciascuno, come illustrato in figura 17.

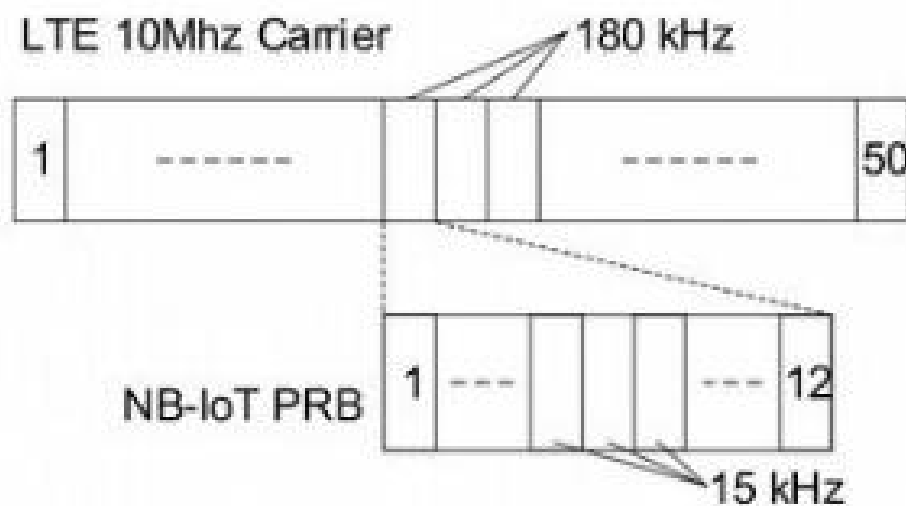


figura 17

Narrowband reference signal (NRS):

NRS viene utilizzato per fornire un riferimento di fase per la demodulazione del canale downlink. Esso viene trasmesso in tutti i sottoframe che possono essere utilizzati per la trasmissione broadcast o downlink utilizzando otto RE per porta.

Per il funzionamento in banda, anche i segnali di riferimento specifici delle cellule LTE (CRS: “*Cell Specific Reference*”) vengono trasmessi nella banda NB-IoT.

Narrowband Primary Synchronous Signal (NPSS)

NPSS fornisce la sincronizzazione sia in tempo che in frequenza nell’acquisizione preliminare del canale; a differenza dell’LTE legacy che contiene 3 segnali di sincronizzazione principali, NB-IoT ha un NPSS per tutte le celle, il che semplifica il processo di rilevamento. NPSS viene trasmesso nel sottoframe 5 di ogni frame radio downlink ogni 10 ms.

Narrowband Secondary Synchronous Signal (NSSS)

Nelle trasmissioni iniziali, l’end device non conosce la modalità di distribuzione e di utilizzo del sottoframe per il traffico LTE, NSSS è pensato quindi per evitare la collisione con i sottoframe LTE.

NSSS determina l’ ID della cella ed ulteriori informazioni sulla struttura del frame, ne esistono 504 e sono univoci.

A differenza dell’ SSS nell’ LTE legacy la cui periodicità è di 5 ms, quello NB-IoT viene trasmesso sul sottoframe 9, con una periodicità di 20 ms e con una periodicità di ripetizione della sequenza di 80 ms. Il segnale di sincronizzazione secondario a banda stretta comprende una sequenza di 132 numeri che utilizzano le m-sequences.

Narrowband physical broadcast channel (NPBCH):

NPBCH trasporta le caratteristiche fisiche del sistema, “*Master Information Block*” (MIB), informazioni importanti per gli end devices nella fase iniziale di connessione.

Ogni MIB viene inviato attraverso una combinazione di otto blocchi di durata 80 ms ed è trasmesso nel sottoframe 0 di ogni frame.

Per evitare interferenza con i canali di segnalazione LTE, NPBCH non viene trasmesso nei tre simboli iniziali. La sua periodicità è di 640 ms (durante i quali il blocco MIB resta invariato) e trasporta 34 bit di informazioni.

MIB trasporta le informazioni di sistema per ottenere il SIB1 (comprende le informazioni di sistema per raggiungere altri SIB), e da queste viene deciso il numero di ripetizioni NPDSCH.

Le dimensioni dei dati del livello fisico contenenti le informazioni del livello superiore corrispondente sono riferite al Transport Block Size (TBS).

La dimensione dei blocchi di trasporto è diversa tra uplink e downlink, in NB-IoT, l’end device sceglie la dimensione del blocco di trasporto compresa tra 2 byte e 125 byte in base all’overhead dello strato superiore; la dimensione massima consentita del blocco di trasporto per NPDSCH è 680 bit (85 byte).

Narrowband physical downlink shared channel(NPDSCH):

NPDSCH si occupa del rilevamento dati, è costituito da dati unicast dell'utente, alcune informazioni di controllo e il blocco di informazioni di sistema (SIB).

IL MIB in NPDSCH trasporta tutte le informazioni per acquisire NB-SIB1, mentre NB-SIB1 trasporta tutte le informazioni per acquisire altre SIB. Se SIB è presente nel frame, occupa sempre il sottoframe 4 in 16 frame continui.

System Information Base

Il SIB è trasmesso su NPDSCH e trasporta le informazioni sulla selezione delle celle, le informazioni sull'accesso alle celle e le informazioni sulla pianificazione.

La sua periodicità è di 2560 ms e contiene informazioni sull'accesso alle celle che contengono lo stato bloccato, il prefisso dell'area di tracciamento, l'ID della cella, le informazioni sulla selezione della cella e il livello minimo di qualità.

Narrowband physical downlink control channel(NPDCCH):

NPDCCH è considerato come l'elemento centrale dei canali di controllo downlink in quanto contiene informazioni di controllo come l'assegnazione UL/DL, la risposta del canale di accesso casuale (RACH: "*Random Access Channel*"), il tipo di modulazione utilizzato per la trasmissione, il controllo dell'alimentazione e così via.

Si occupa inoltre di controllare la trasmissione dei dati tra la stazione base (BS: "*Base Station*") e l'apparecchiatura utente (UE: "*User Equipment*").

La dimensione delle informazioni sul controllo è fissa a 23 bit e viene codificata su un sottoframe, l'estensione della copertura si ottiene attraverso l'uso della codifica di ripetizione, per un massimo di 2048 ripetizioni.

NPDCCH contiene un blocco logico sulle informazioni di controllo del downlink (DCI), esso trasporta informazioni critiche come: delay scheduling, assegnazione delle risorse e indicazione della sottoportante. DCI è classificato in 3 formati, di cui il formato N0 corrisponde allo scheduling in uplink e il formato N1 è principalmente destinato allo scheduling in downlink [11].

NPDCCH è codificato e occupa più elementi del canale di controllo a banda stretta (NCCE: “*Narrowband Control Channel Element*”), quest’ultimo è composto da sei frequenze in un sottoframe; nel formato 0, ci sarà un NCCE e due nel formato 1.

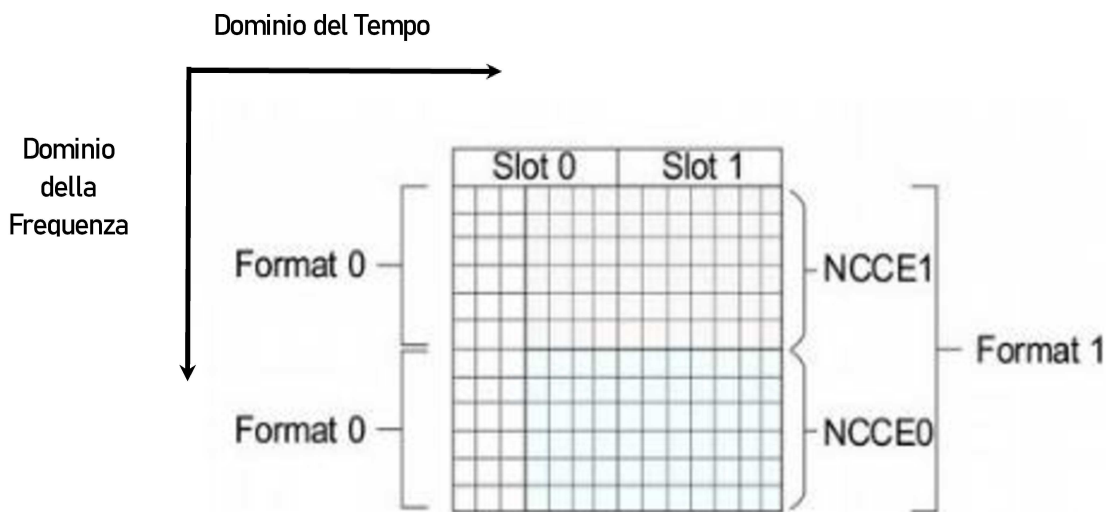


Figura 18

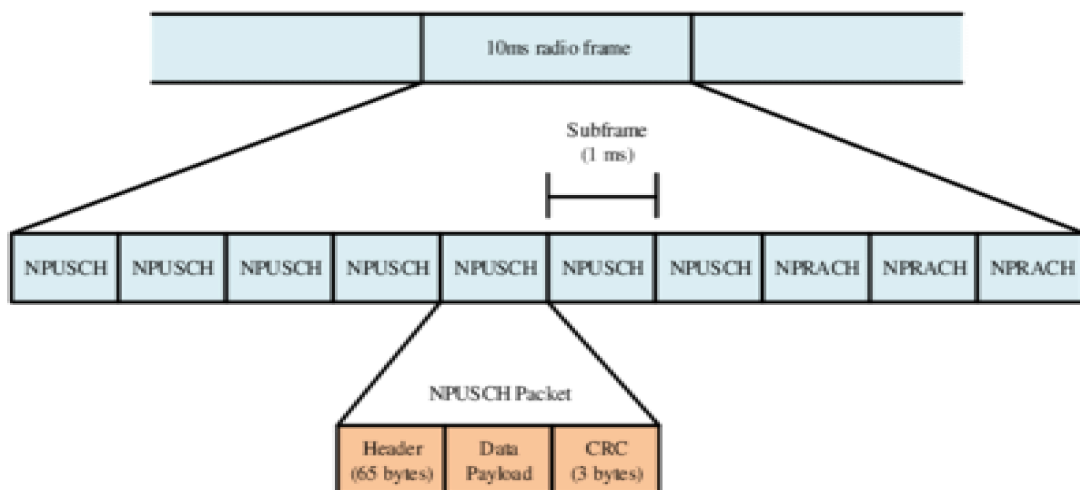
3.3.2 Uplink

Nell'uplink (figura 19), NB-IoT supporta sia trasmissioni single-tone che multi-tone. La trasmissione multi-tone utilizza lo stesso schema SC-FDMA con una spaziatura tra sottoportanti di 15 kHz e una larghezza di banda totale di 180 kHz, gli slot hanno una durata di 0,5 ms e i sottoframe di 1 ms come nell'LTE.

Tuttavia, la trasmissione single-tone supporta sia 15kHz che 3,75 kHz di spaziatura tra sottoportanti. La modalità 15 kHz è simile a quella di LTE mentre la spaziatura di 3,5 kHz è quattro volte più lunga rispetto a 15 kHz, il che si traduce in una lunghezza di slot di 2 ms, ognuno dei quali ha 7 simboli OFDM con 48 sottoportanti.

NB-IoT in uplink ha un segnale fisico e due canali fisici:

- De-modulated Reference Signal (DMRS)
- NB Physical Uplink Shared Channel (NPUSCH)
- NB Physical Random Access Channel (NPRACH).



(b) Uplink Frame Structure

Figura 19

Demodulation reference signal (DMRS):

In uplink, i DMRS sono multiplati con i dati in modo che vengano trasmessi solo in RU (*Resource Unit*) contenenti dati. A seconda della trasmissione uplink, il DMRS viene trasmesso in uno o tre simboli SC-OFDMA per slot.

Narrowband Uplink Shared Channel (NPUSCH):

Contrariamente a LTE, sia i dati che le informazioni di controllo vengono trasferiti sul canale condiviso uplink. La distinzione è fatta utilizzando due formati, il formato 1 viene utilizzato per l'uplink dei dati e utilizza il codice turbo per la correzione degli errori. Il formato 2 viene utilizzato per segnalare il riconoscimento HARQ (*Hybrid automatic repeat request*) per i dati downlink e utilizza il codice di ripetizione fino a 128.

NPUSCH è composto da canali da 3,75 kHz o 15 kHz, come in figura 20, i canali da 15 kHz, come visto prima, hanno 20 sottoframe con 12 portanti ed ogni slot (di durata 0,5 ms) trasporta sette simboli di informazione usando OFDMA, allo stesso modo i canali da 3,75 kHz hanno cinque sottoframe con 48 portanti.

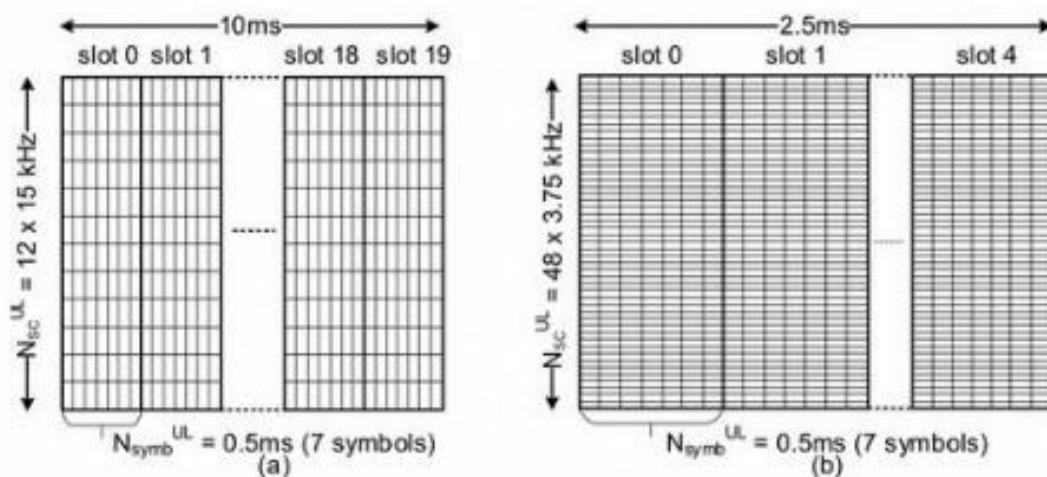


Figura 20

Narrowband physical random access channel (NPRACH):

NPRACH consente all'UE di connettersi ad una BS (Base Station) e permette a quest'ultima di stimare il tempo di arrivo (ToA) del segnale. Il ToA del segnale NPRACH ricevuto è il ritardo di propagazione tra la stazione radio base e l'end device. La BS utilizza il preambolo di accesso casuale inviato da un terminale utente per stimare la tempistica uplink e ciò è necessario per emettere un comando di temporizzazione in modo da mantenere l'ortogonalità uplink tra i diversi utenti.

Il preambolo di accesso casuale NB-IoT è mostrato nella Figura 21. È principalmente caratterizzato da quattro gruppi di simboli, ognuno dei quali comprende cinque simboli OFDM più il prefisso ciclico. In particolare, la rete configura il formato del preambolo (ovvero la durata del prefisso ciclico), a seconda della dimensione della cella. Un singolo preambolo NPRACH dura 5,6 ms o 6,4 ms, quando si applica rispettivamente il formato 0 o il formato 1.

La risorsa NPRACH occupa un insieme contiguo di 12, 24, 36 e 48 canali e per supportare l'estensione della copertura, il preambolo di quattro gruppi di simboli può essere ripetuto fino a 128 volte, con un algoritmo pseudo-casuale di salto di frequenza che offre come preamboli diversi il numero di portanti secondarie assegnate ad NPRACH. Di conseguenza, la selezione di diverse sottoportanti all'inizio della trasmissione garantisce schemi di salto che non si sovrappongono mai.

All'inizio, la stazione radio base alloca le risorse RACH in base al livello di copertura, esistono tre livelli classificati in base al valore MCL, però l'operatore di rete può impostare manualmente il numero di risorse a supporto del particolare livello di copertura. Generalmente, una risorsa NPRACH è chiamata "Random Access Opportunity" (RAO).

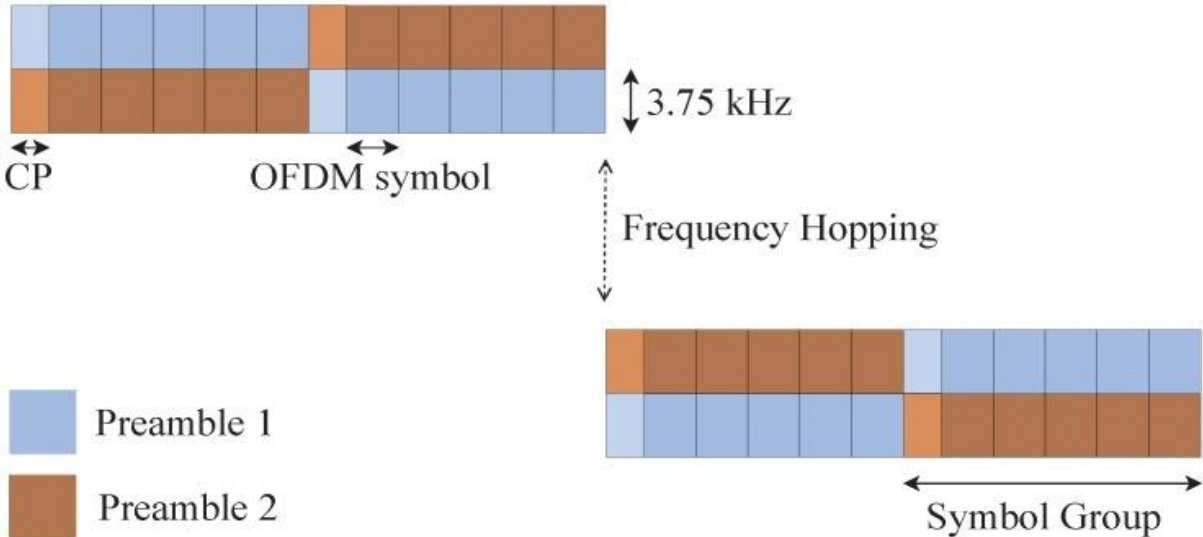


Figura 21

Random Access Procedure

Come illustrato in figura 22 la procedura inizia con la stazione radiobase che trasmette la sincronizzazione dei segnali agli UE; durante NPSS, l'UE viene sincronizzato con la stazione radiobase. Durante NSSS, l'UE rileva la struttura del frame e l'identità della cella e dopo essersi sincronizzato si mette in ascolto del canale di trasmissione (NPBCH) per le informazioni sullo scheduling in uplink. Al momento del recupero delle informazioni del MIB, l' UE conosce la modalità operativa, l' SFN, l' Hyper SFN e le informazioni di scheduling SIB1.

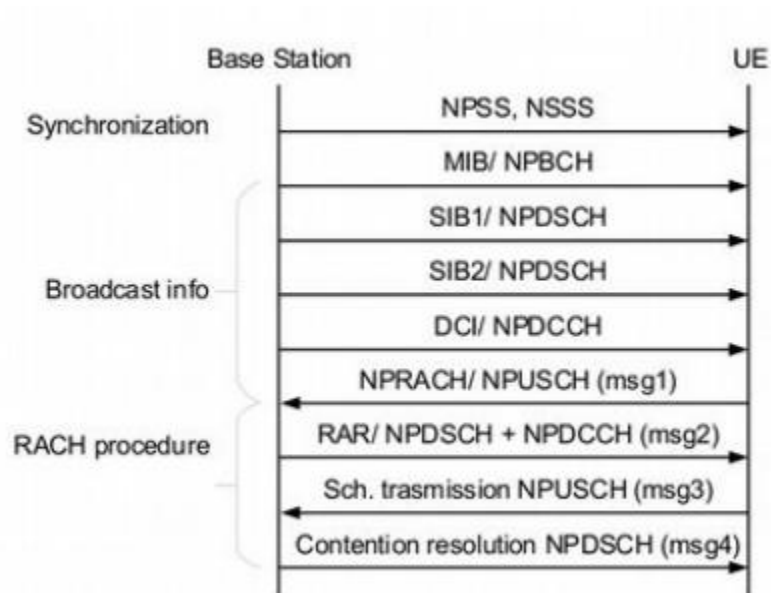


Figura 22

SIB2 contiene la configurazione delle risorse radio, la configurazione NPRACH e le informazioni sullo scheduling uplink/downlink.

Il downlink control information (DCI) trasmette i parametri di allocazione delle risorse per la procedura di accesso casuale oltre che il ritardo di scheduling. Dopo aver letto l'allocazione delle risorse di uplink per la procedura di accesso casuale, l'UE adegua i tempi di uplink ed inizia a trasmettere il preambolo sulla frequenza scelta casualmente.

Ripetizione del frame uplink (figura 23):

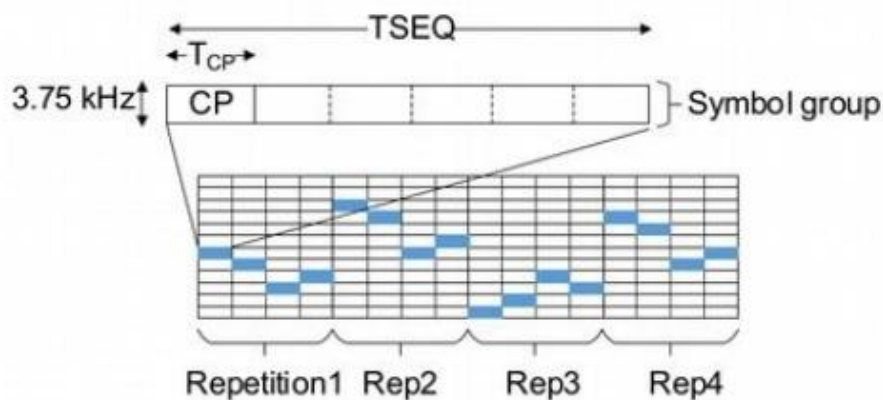


Figura 23

Meccanismo di Contesa

La procedura di accesso casuale NB-IoT è basata sulla contesa e comporta lo scambio di quattro messaggi, come illustrato nella figura 24 :

1. Allocazione del preambolo ad accesso casuale: la stazione radiobase trasmette informazioni sul preambolo ad accesso casuale (RA). L'UE trasmette il preambolo durante la risorsa di accesso casuale fornita. Potrebbe esserci una collisione con altri UE. La collisione viene rilevata dalla stazione radio base.

2. Allocazione RAR (Random Access Response): alla ricezione del preambolo, la stazione base trasmette un RAR che istruisce esplicitamente l'utente su quali risorse di uplink devono essere utilizzate per la trasmissione del messaggio successivo. Se questo messaggio non viene ricevuto, l'utente continua ad attendere fino alla scadenza del W .
3. Richiesta di risorse radio: sfruttando le risorse programmate, l'utente trasmette la sua identità e altre informazioni importanti; questo messaggio è anche noto come Msg3. Successivamente, il terminale avvia il timer di risoluzione contesa.
4. Risoluzione dei conflitti: la stazione base esegue la risoluzione della contesa e restituisce ai dispositivi l'identità degli utenti vincenti tramite il Messaggio di risoluzione della contesa. Se questo messaggio non viene ricevuto, gli utenti continuano ad attenderlo fino alla scadenza del timer di risoluzione delle controversie.

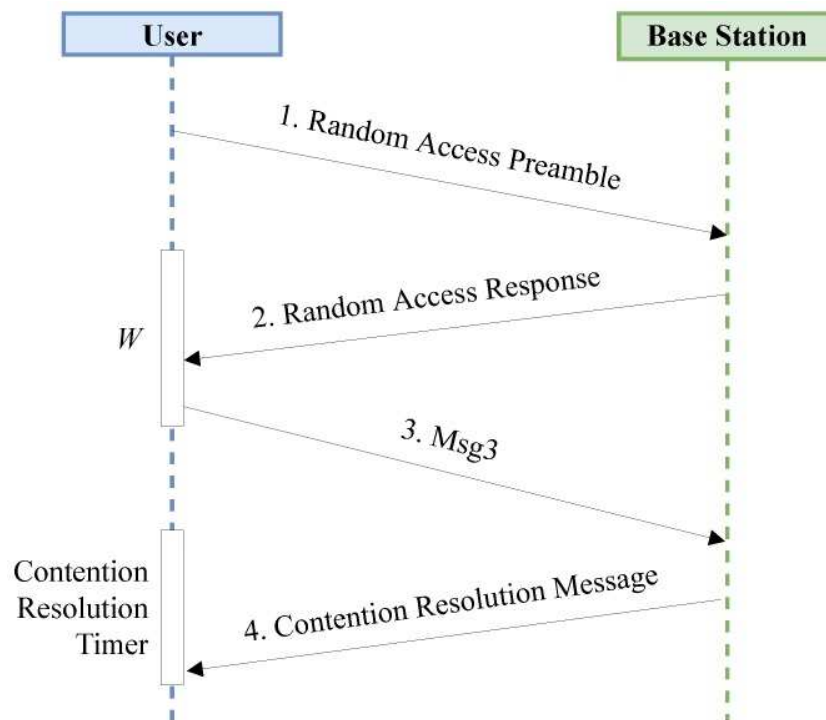


Figura 24

CAPITOLO 4

MQTT

L'MQTT (**Message Queue Telemetry Transport**), è un protocollo di comunicazione nato circa 20 anni fa [12], per l'esattezza nel 1999, con l'obiettivo di gestire le connessioni machine to machine, ovvero per abilitare la comunicazione tra le macchine in maniera estremamente efficiente e nel corso del tempo si è rivelato particolarmente adatto per la trasmissione dei dati prodotti dai dispositivi IoT.

Le caratteristiche principali di MQTT sono quelle di essere un protocollo semplice e leggero per lo scambio di messaggi, nonché di minimizzare il traffico sulle reti e richiedere poche risorse ai dispositivi per la sua gestione. Si tratta inoltre di un protocollo capace di eseguire in maniera efficiente la distribuzione di messaggi da uno a molti destinatari, il disaccoppiamento delle applicazioni e la scalabilità dei sistemi. Questo protocollo è mirato a ridurre al minimo i requisiti delle risorse dei dispositivi tentando al contempo di garantire affidabilità e un certo livello di garanzia di consegna.

L'MQTT è un protocollo di messaggistica asincrono di pubblicazione/sottoscrizione (figura 25) orientato alla connessione basato sullo stack TCP/IP ed è basato su un apposito message broker progettato per essere semplice, leggero e facile da implementare.

Tale protocollo è un'alternativa al modello client/server, tipico dell'http, dove il client comunica direttamente con un endpoint, appunto il server. Il publish/subscribe prevede invece che il client mittente, chiamato publisher, invece di inviare messaggi a un determinato set di destinatari, pubblica essi su un certo argomento, detto topic, sul message broker, un applicativo in grado di filtrare e gestire il flusso dati.

Ogni client destinatario, detto subscriber, si iscrive ai topic di suo interesse e, ogni volta che un nuovo messaggio viene pubblicato su uno specifico topic il message broker lo distribuisce a tutti i destinatari sottoscritti.

In tal modo è molto semplice configurare una messaggistica uno-a-molti ed inoltre tale infrastruttura non necessita di sincronizzazione, ovvero i client possono lavorare anche non

contemporaneamente e dispositivi e processi in esecuzione non devono interrompersi durante la ricezione o la trasmissione dei dati.

Dal punto implementativo occorre configurare il broker definendone l'indirizzo IP e la porta in uso (di default l'MQTT utilizza 1883), in caso previsti occorre anche effettuare una fase di autenticazione tramite la coppia username/password. Sono disponibili differenti broker, per citarne alcuni: Mosquitto, completamente open source e scritto in C, Mosca, basato su node.js, VerneMQ, HiveMQ, emqttd.

Un ulteriore punto di forza del protocollo risulta essere l'offerta di tre differenti livelli di qualità del servizio (QoS) per la consegna dei messaggi:

- **QoS 0:** Anche chiamato 'at most once' (al massimo una volta), prevede la distribuzione dei messaggi in base a quanto di meglio garantito dalla rete TCP/IP sottostante. Con tale livello i messaggi possono essere duplicati o non arrivare affatto. È tipicamente utilizzato in tutti quei contesti nei quali la perdita di un messaggio non è cruciale, ad esempio nelle reti di sensori ambientali/domestiche.
- QoS 1: Anche chiamato 'at least once' (almeno una volta) assicura l'arrivo dei messaggi, ma possono comunque verificarsi dei duplicati.
- QoS 2: Definito 'exactly once' (esattamente una volta), assicura che i messaggi vengano consegnati una e una sola volta. Da ciò risulta essere la soluzione più onerosa per i sistemi, per la rete e per i timing ma d'altro canto anche la più affidabile.

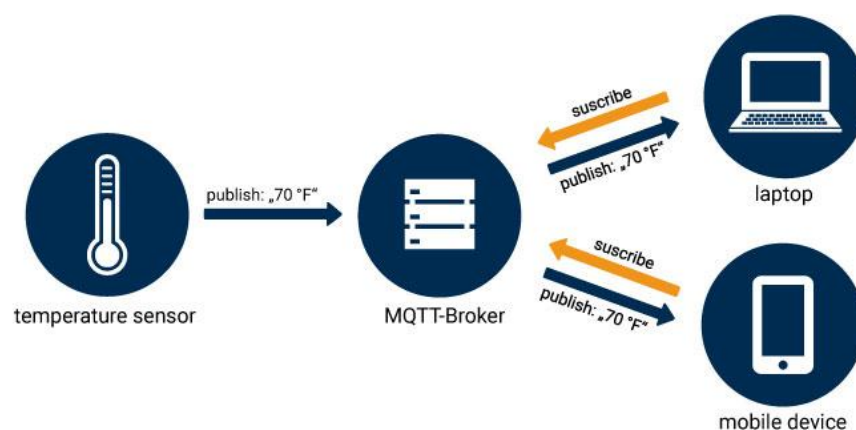


figura 25

CAPITOLO 5

Implementazione sketch Arduino

La scheda Sodaq Sara R412M è compatibile Arduino, pertanto è stato utilizzato il software Arduino IDE per la programmazione.

Tale scheda supporta l'uso degli AT commands (uno specifico insieme di comandi originalmente sviluppato per il modem Hayes Smartmodem da 300 baud).

Ogni sketch (programma eseguibile) Arduino si compone di due funzioni principali:

“void setup” che viene invocata solo una volta all’inizio del programma ed è utilizzata per configurare il setting iniziale;

“void loop” che viene invocata ripetutamente, la cui esecuzione si interrompe solo quando viene tolta l'alimentazione alla scheda oppure viene premuto il tasto di reset.

Di seguito è riportato lo sketch relativo all'attivazione degli AT commands:

```
#include <Arduino.h>
#include <Sodaq_R4X.h>
#include <Sodaq_LSM303AGR.h>

#define R4XX // Uncomment when you use the ublox R4XX module

#if defined(ARDUINO_SODAQ_SARA) /* SODAQ SARA */
#define DEBUG_STREAM SerialUSB
#define MODEM_STREAM Serial1
#define powerPin SARA_ENABLE
#define enablePin SARA_TX_ENABLE
#else
#error "Please select the SODAQ SARA as your board"
#endif

#if defined(R4XX)
unsigned long baud = 115200; //start at 115200 allow the USB port to change the Baudrate
#else
unsigned long baud = 9600; //start at 9600 allow the USB port to change the Baudrate
```

```

#endif
void setup()
{
#ifdef powerPin
  // Turn the nb-iot module on
  pinMode(powerPin, OUTPUT);
  digitalWrite(powerPin, HIGH);
#endif

#ifdef enablePin
  //Set state to active
  pinMode(enablePin, OUTPUT);
  digitalWrite(enablePin, HIGH);
#endif //enablePin

#ifdef defined(R4XX)
  pinMode(SARA_R4XX_TOGGLE, OUTPUT);
  digitalWrite(SARA_R4XX_TOGGLE, LOW);
  delay(2000);
  pinMode(SARA_R4XX_TOGGLE, INPUT);
#endif

  // Start communication
  DEBUG_STREAM.begin(baud);
  MODEM_STREAM.begin(baud);
}
// Forward every message to the other serial
void loop()
{
  while (DEBUG_STREAM.available())
  {
    MODEM_STREAM.write(DEBUG_STREAM.read());
  }
  while (MODEM_STREAM.available())
  {
    DEBUG_STREAM.write(MODEM_STREAM.read());
  }
  // check if the USB virtual serial wants a new baud rate
  // This will be used by the UEUpdater to flash new software
  if (DEBUG_STREAM.baud() != baud) {
    baud = DEBUG_STREAM.baud();
    MODEM_STREAM.begin(baud);
  }
}
}

```

Una volta inserito lo sketch possiamo inserire i comandi da monitor seriale.

Uno dei comandi più utili in questo progetto è “AT+CSQ” il quale serve per verificare la connessione: se il valore è 99,99 non c’è connessione, in tutti gli altri casi la scheda riesce a connettersi alla rete e a seconda del valore è possibile determinare la qualità della connessione

es.	9	-95 (RSSI)	Debole
	12	-89 (RSSI)	Ok
	17	-79 (RSSI)	Buono
	23	-67 (RSSI)	Eccellente

Una volta che la connessione è stata garantita procediamo all’inserimento dello sketch che prende i dati raccolti dai sensori e li invia al broker (Cloud MQTT) tramite protocollo MQTT.

La prima parte del codice vede l’inclusione di tutte le librerie necessarie e la definizione dei parametri per la connessione e per l’operazione di raccolta dati.

```
#include <Sodaq_R4X.h>
#include <Sodaq_wdt.h>
#include <Sodaq_LSM303AGR.h>
#include <Sodaq_UBlox_GPS.h>
#include <Wire.h>
#include "ThingSpeak.h"
#include <DHT.h>

/* Parametri per SODAQ SARA AFF R412M*/
#define DEBUG_STREAM SerialUSB
#define MODEM_STREAM Serial1
#define powerPin SARA_ENABLE
#define enablePin SARA_TX_ENABLE

#define DEBUG_STREAM SerialUSB
#define DEBUG_STREAM_BAUD 115200

#define STARTUP_DELAY 15000

//parametri TIM per connessione mqtt
#define CURRENT_APN "nbiot.tids.tim.it"
#define CURRENT_URAT "8"
```

```
//parametri per il server cloudmqtt
#define MQTT_SERVER_NAME "hairdresser.cloudmqtt.com"
#define MQTT_SERVER_PORT 15731
#define MQTT_SERVER_CLIENT "jylhjcod"
#define MQTT_SERVER_password "UyFRsdfOp10T"
```

I parametri per la connessione dipendono dal tipo di scheda sim utilizzata (nel caso attuale, una scheda Tim M2M) e dal paese nel quale ci troviamo.

Ad ogni sensore sono associate diverse librerie e comandi per la raccolta dei dati, di seguito sono riportati gli sketch per l'inizializzazione della scheda e dei sensori:

```
//DHT22

#define DHTPIN 2 //Pin a cui è connesso il sensore
#define DHTTYPE DHT22 //Tipo di sensore che stiamo utilizzando (DHT22)
DHT dht(DHTPIN, DHTTYPE); //Inizializza oggetto chiamato "dht", parametri: pin a cui è connesso il sensore, tipo di dht
//11/22
//Variabili
float hum; //Variabile in cui verrà inserita la % di umidità
float temp; //Variabile in cui verrà inserita la temperatura

//Z14A

//Request Gas concentration command
const long samplePeriod = 10000L;
const byte requestReading[] = {0xFF, 0x01, 0x86, 0x00, 0x00, 0x00, 0x00, 0x00, 0x79};
byte result[9];
long lastSampleTime = 0;

//SODAQ

static Sodaq_R4X r4x;
static Sodaq_SARA_R4XX_OnOff saraR4xxOnOff;
static bool isReady;
{
DEBUG_STREAM.println();
    DEBUG_STREAM.println("Sending message through MQTT");
```

```

String reading = "{\"RSSI\":{\"value\":"+ String(getsegnale())+"}"}";
uint8_t size = reading.length();
int lengthSent = r4x.mqttPublish("boh", (uint8_t*)reading.c_str(), size, 0, 0, 1);

DEBUG_STREAM.print("Length buffer vs sent:");
DEBUG_STREAM.print(size);
DEBUG_STREAM.print(",");
DEBUG_STREAM.println(lengthSent);
DEBUG_STREAM.println();
}

//SETUP//

void setup()
{
  sodaq_wdt_safe_delay(STARTUP_DELAY);

  Serial.begin(9600);
  SerialUSB.begin(9600);
  Serial1.begin( 9600, SERIAL_8N1 );

  //SETUP DHT22

  dht.begin();
  Wire.begin();

  //SETUP Z14A

  analogWriteResolution(10); //valgono anche per mq137
  analogReadResolution(12); //e servono per abilitare lettura da porta analogica

  pinMode(A0,INPUT);

  //SETUP MQ137

  pinMode(A1,INPUT);

```


Una volta completata l'operazione di inizializzazione possiamo procedere alla connessione con il broker MQTT e all'invio dei dati:

```
//INIZIO LA CONNESSIONE
DEBUG_STREAM.begin(DEBUG_STREAM_BAUD);
MODEM_STREAM.begin(r4x.getDefaultBaudrate());
DEBUG_STREAM.println("Inizializzazione e connessione... ");
r4x.setDiag(DEBUG_STREAM);
r4x.init(&saraR4xxOnOff, MODEM_STREAM)
isReady = r4x.connect(CURRENT_APN, CURRENT_URAT); //per connessione Nb-IoT

//creo la connessione al cloud server MQTT
if(isReady){
isReady=r4x.mqttSetServer(MQTT_SERVER_NAME,MQTT_SERVER_PORT)&&r4x.mqttSetAuth(MQTT_SERVER_C
IENT, MQTT_SERVER_password) && r4x.mqttLogin();
DEBUG_STREAM.println(isReady ? "MQTT connected" : "MQTT failed");
}
else DEBUG_STREAM.println("connection failed");

//dopo aver terminato il setup della scheda e la connessione invio i dati per la prima volta
sendrssiMQTT();
sendDHT22();
sendZ14A();
sendMQ137();
}

//SEND DHT22 MQTT

void sendDHT22(){
hum = dht.readHumidity();
temp= dht.readTemperature();
uint8_t size = String(hum).length(); //utilizzo il comando String per passare da dato di tipo float o int a stringa
uint8_t size2 = String(temp).length();

//invio dati mqtt
int lengthSent = r4x.mqttPublish("dht22/temperature", (uint8_t*)String(temp).c_str(), size, 0, 0, 1);
int lengthSent2 = r4x.mqttPublish("dht22/humidity", (uint8_t*)String(hum).c_str(), size2, 0, 0, 1);

//stampa a video
SerialUSB.print("Umidità: ");
SerialUSB.print(hum);
SerialUSB.print(" %, Temp: ");
SerialUSB.print(temp);
SerialUSB.println(" Celsius");
}
```

```
//SEND Z14A MQTT
```

```
void sendZ14A()
```

```
{
```

```
  analogReference(AR_DEFAULT);
```

```
  ADC->INPUTCTRL.bit.GAIN = ADC_INPUTCTRL_GAIN_DIV2_Val;
```

```
  ADC->REFCTRL.bit.REFSEL = ADC_REFCTRL_REFSEL_INTVCC1;
```

```
  delay (1000); //should give the ADC a little bit time to save and apply the settings
```

```
  long now = millis();
```

```
  if (now > lastSampleTime + samplePeriod) {
```

```
    lastSampleTime = now;
```

```
    int ppmV = readPPMV();
```

```
    //invio dati mqtt
```

```
    uint8_t size = String(ppmV).length();
```

```
    int lengthSent = r4x.mqttPublish("z14a/ppm", (uint8_t*)String(ppmV).c_str(), size, 0, 0, 1);
```

```
    //stampa a video
```

```
    SerialUSB.print(ppmV);
```

```
    SerialUSB.print(" ppm");
```

```
    SerialUSB.print("\n");
```

```
  }
```

```
}
```

```
int readPPMV() {
```

```
  float v = analogRead(A0); /* 5.0 / 1023.0;
```

```
  int ppm = int((v - 0.4) * 3125.0);
```

```
  return v;
```

```
}
```

```
//SEND MQ137 MQTT
```

```
void sendMQ137()
```

```
{
```

```
  analogReference(AR_DEFAULT);
```

```
  ADC->INPUTCTRL.bit.GAIN = ADC_INPUTCTRL_GAIN_DIV2_Val;
```

```
  ADC->REFCTRL.bit.REFSEL = ADC_REFCTRL_REFSEL_INTVCC1;
```

```
  delay (1000); //ritardo per dare tempo all ADC di salvare e applicare le impostazioni
```

```
  float sensor_volt; //Variabile per il voltaggio del sensore
```

```
  float RS_air; //Variabile per la resistenza del sensore
```

```
  float R0;
```

```

float sensorValue;    //Definisce la variabile per la lettura dei valori analogici
for(int x = 0 ; x < 500 ; x++) //Inizia il loop
{
    sensorValue = sensorValue + analogRead(A1);    //Somma i valori analogici del sensore 500 volte
}
sensorValue = sensorValue/500.0;    //Prende il valor medio dei dati raccolti
sensor_volt = sensorValue*(5.0/1023.0);    //converte dalla media ai volt
RS_air = ((5.0*10.0)/sensor_volt)-10.0;    //calcolo di RS nell'aria
R0 = RS_air/3.6; //Calculate R0

//invio dati mqtt

uint8_t size = String(R0).length();
int lengthSent = r4x.mqttPublish("mq137/R0", (uint8_t*)String(R0).c_str(), size, 0, 0, 1);

//stampa a video
SerialUSB.print("R0 = "); //Stampa "R0"
SerialUSB.println(R0);    //Stampa valore di R0
delay(1000);    //Aspetto 1 secondo
}

//SEND SDS018 MQTT
void sendSDS018() {

uint8_t mData = 0;
uint8_t mPkt[10] = {0};
uint8_t mCheck = 0;

if(Serial.available() > 0){

    for( int i=0; i<10; ++i ) {
        mPkt[i] = Serial.read();
        Serial.println( mPkt[i], HEX );
    }
}

uint8_t pm25Low  = mPkt[2];
uint8_t pm25High = mPkt[3];
uint8_t pm10Low  = mPkt[4];
uint8_t pm10High = mPkt[5];

float pm25 = ( ( pm25High * 256.0 ) + pm25Low ) / 10.0;
float pm10 = ( ( pm10High * 256.0 ) + pm10Low ) / 10.0;

SerialUSB.print( "PM2.5: " );
SerialUSB.print( pm25 );

```

```

SerialUSB.print( "\nPM10 :");
SerialUSB.print( pm10 );
SerialUSB.println();
Serial1.flush();

uint8_t size = String(pm25).length();
uint8_t size2 = String(pm10).length();

//invio dati mqtt

int lengthSent = r4x.mqttPublish("dht22/pm_2.5", (uint8_t*)String(pm25).c_str(), size, 0, 0, 1);
int lengthSent2 = r4x.mqttPublish("dht22/pm_10", (uint8_t*)String(pm10).c_str(), size2, 0, 0, 1);
delay( 10000 );
}

//terminato il setup passo al ciclo di ripetizione dei comandi da effettuare
void loop()
{
  sondaq_wdt_safe_delay(10000); // invio dati ogni 10s
  sendDHT22();
  sendZ14A();
  sendMQ137();
  sendSDS018();
}

```

Quando il codice è stato caricato possiamo controllare la ricezione dei dati da parte del broker (figura 26):

The screenshot shows two windows. The left window is a terminal titled 'COM3' displaying the following output:

```

18:30:33.090 -> MQTT connected
18:30:33.090 ->
18:30:33.090 -> Sending message through MQTT
18:30:33.090 -> Length buffer vs sent:22,1
18:30:33.090 ->
18:30:33.124 -> Umidità: 58.20 %, Temp: 25.30 Celsius
18:30:34.146 -> 1205 ppm
18:30:35.374 -> R0 = 20.04
18:30:46.412 -> Umidità: 58.00 %, Temp: 25.30 Celsius
18:30:47.432 -> 1196 ppm
18:30:48.655 -> R0 = 19.96
18:30:49.675 -> PM2.5: 21.70
18:30:49.675 -> PM10 :23.80

```

The right window is a web browser showing the MQTT broker interface. The 'Received messages' section contains the following data:

Topic	Message
boh	{\"RSSI\":{\"value\":-87}}
dht22/temperature	25.30
dht22/humidity	58.20
z14a/ppm	1205
mq137/R0	20.04

figura 26

CAPITOLO 6

Conclusioni

In seguito verranno riportati i dati raccolti in situazioni diverse, ricreate apposta per estremizzare situazioni di vita quotidiana.

Valori ottenuti misurando la qualità dell'aria in una sala vuota e arieggiata (figura 27):

```
16:25:31.879 -> RO = 23.09
16:25:32.877 -> PM2.5: 20.50
16:25:32.877 -> PM10 :22.30
16:25:52.926 -> Umidità: 57.70 %, Temp: 25.30 Celsius
16:25:53.949 -> 1209 ppm
16:25:55.177 -> RO = 23.02
16:25:56.167 -> PM2.5: 20.30
16:25:56.167 -> PM10 :22.10
16:26:16.264 -> Umidità: 57.70 %, Temp: 25.30 Celsius
16:26:17.250 -> 1219 ppm
16:26:18.518 -> RO = 23.03
16:26:19.509 -> PM2.5: 20.30
16:26:19.509 -> PM10 :22.10
16:26:39.568 -> Umidità: 57.70 %, Temp: 25.30 Celsius
16:26:40.562 -> 1204 ppm
16:26:41.797 -> RO = 22.99
16:26:42.822 -> PM2.5: 20.20
16:26:42.822 -> PM10 :22.00
16:27:02.890 -> Umidità: 57.60 %, Temp: 25.30 Celsius
16:27:03.911 -> 1206 ppm
16:27:05.144 -> RO = 23.04
16:27:06.142 -> PM2.5: 19.90
16:27:06.142 -> PM10 :21.70
16:27:26.210 -> Umidità: 57.60 %, Temp: 25.30 Celsius
16:27:27.207 -> 1203 ppm
16:27:28.440 -> RO = 23.11
16:27:29.432 -> PM2.5: 19.60
16:27:29.432 -> PM10 :21.40
16:27:49.513 -> Umidità: 57.60 %, Temp: 25.30 Celsius
16:27:50.536 -> 1203 ppm
16:27:51.758 -> RO = 22.94
16:27:52.776 -> PM2.5: 20.00
16:27:52.776 -> PM10 :21.80
16:28:12.845 -> Umidità: 57.60 %, Temp: 25.30 Celsius
16:28:13.836 -> 1191 ppm
16:28:15.093 -> RO = 23.10
```

figura 27

N.B. queste misure sono state effettuate di pomeriggio, quelle che seguiranno (ottenute nella stessa sala) sono state effettuate invece di sera; questo aspetto può influire su alcuni valori come ad esempio la temperatura e l'umidità.

Valori ottenuti misurando la qualità dell'aria in un salone con otto persone (durante la cena) e finestre chiuse (figura 28):

```

21:17:24.273 -> PM10 :22.50
21:17:44.327 -> Umidità: 62.40 %, Temp: 25.10 Celsius
21:18:45.537 -> 1602 ppm
21:18:46.795 -> R0 = 7.46
21:18:47.784 -> PM2.5: 16.50
21:18:47.784 -> PM10 :22.60
21:21:08.255 -> Umidità: 60.50 %, Temp: 25.20 Celsius
21:21:09.275 -> 1637 ppm
21:21:10.497 -> R0 = 7.12
21:21:11.515 -> PM2.5: 16.50
21:21:11.515 -> PM10 :22.20
21:21:31.581 -> Umidità: 61.50 %, Temp: 25.20 Celsius
21:21:32.570 -> 1641 ppm
21:21:33.828 -> R0 = 6.95
21:21:34.814 -> PM2.5: 16.50
21:21:34.814 -> PM10 :22.10
21:21:54.905 -> Umidità: 61.90 %, Temp: 25.30 Celsius
21:21:55.926 -> 1650 ppm
21:21:57.183 -> R0 = 6.86
21:21:58.173 -> PM2.5: 16.30
21:49:26.978 -> Umidità: 62.10 %, Temp: 25.60 Celsius
21:50:28.217 -> 1961 ppm
21:51:29.644 -> R0 = 3.11
21:53:41.124 -> Umidità: 62.20 %, Temp: 25.60 Celsius
21:54:42.354 -> 1973 ppm
21:55:43.800 -> R0 = 3.09
21:55:44.783 -> PM2.5: 24.50
21:55:44.783 -> PM10 :27.10
22:00:05.701 -> Umidità: 62.60 %, Temp: 25.70 Celsius
22:01:06.927 -> 1907 ppm
22:02:08.361 -> R0 = 8.23
22:02:09.381 -> PM2.5: 24.60
22:02:09.381 -> PM10 :27.40

```

figura 28

Nella seguente tabella sono analizzate le variazioni tra la sala vuota e arieggiata (Situazione 1) e la sala con finestre chiuse e otto persone (Situazione 2).

	Situazione 1	Situazione 2	Variazione (2)
Umidità	57.6	62.6	+8,5%
Temperatura	25.3	25.7	+1.6%
CO2	1206	1973	+63.6%
R0	23.04	7.12	-69.01%
PM2.5	19.9	16.3	-18.1%
PM10	22.1	22.1	--

N.B. nella tabella sopra (e in tutte le successive) sono stati raccolti i dati più significativi (evidenziati nelle immagini) per far notare maggiormente le variazioni dei diversi valori.

Dopo circa un'ora ho aperto le finestre per vedere il cambiamento dei valori raccolti.
 Valori ottenuti misurando la qualità dell'aria in un salone con otto persone (durante la cena) e finestre aperte (figura 29):

```

21:49:26.978 -> Umidità: 62.10 %, Temp: 25.60 Celsius
21:50:28.217 -> 1961 ppm
21:51:29.644 -> R0 = 3.11
21:53:41.124 -> Umidità: 62.20 %, Temp: 25.60 Celsius
21:54:42.354 -> 1973 ppm
21:55:43.800 -> R0 = 3.09
21:55:44.783 -> PM2.5: 24.50
21:55:44.783 -> PM10 :27.10
22:00:05.701 -> Umidità: 62.60 %, Temp: 25.70 Celsius
22:01:06.927 -> 1907 ppm
22:02:08.361 -> R0 = 8.23
22:02:09.381 -> PM2.5: 24.60
22:02:09.381 -> PM10 :27.40
22:06:30.331 -> Umidità: 62.90 %, Temp: 25.70 Celsius
22:07:31.574 -> 1385 ppm
22:08:33.017 -> R0 = 15.26
22:08:34.000 -> PM2.5: 24.60
22:08:34.000 -> PM10 :27.40
22:12:54.968 -> Umidità: 59.30 %, Temp: 25.30 Celsius
22:13:56.226 -> 1175 ppm
22:14:57.664 -> R0 = 21.67
22:14:58.685 -> PM2.5: 24.50
22:14:58.685 -> PM10 :27.30
22:19:19.637 -> Umidità: 60.00 %, Temp: 24.80 Celsius
22:20:20.868 -> 1140 ppm
22:21:22.329 -> R0 = 22.68
22:21:23.352 -> PM2.5: 24.50
22:21:23.352 -> PM10 :27.20
22:25:44.346 -> Umidità: 60.70 %, Temp: 24.50 Celsius
22:26:45.597 -> 1125 ppm
22:27:47.038 -> R0 = 25.26
22:27:48.056 -> PM2.5: 24.40
22:27:48.056 -> PM10 :27.10
22:32:09.051 -> Umidità: 61.30 %, Temp: 24.30 Celsius
22:33:10.299 -> 1140 ppm
22:34:11.757 -> R0 = 25.90
    
```

figura 29

Passiamo dunque ad analizzare le variazioni dei dati raccolti quando si passa dalla Situazione 1 (sala con finestre chiuse e otto persone) alla Situazione 2 (sala con otto persone arieggiata).

	Situazione 1	Situazione 2	Variazione (2)
Umidità	62.6	59.3	-5.27%
Temperatura	25.7	24.3	-5.44%
CO2	1973	1140	-42.2%
R0	7.12	22.68	+218.5%
PM2.5	16.3	24.4	+49.7%
PM10	22.1	27.1	+22.6%

Valori ottenuti misurando la qualità dell'aria in una camera con porta e finestre chiuse (per due giorni) e una persona al computer (figura 30):

```
11:00:10.978 -> Umidità: 56.60 %, Temp: 26.60 Celsius
11:00:11.997 -> 2366 ppm
11:00:13.226 -> R0 = 16.85
11:00:24.282 -> Umidità: 56.50 %, Temp: 26.60 Celsius
11:00:25.298 -> 2372 ppm
11:00:26.520 -> R0 = 16.89
11:00:27.506 -> PM2.5: 26.90
11:00:27.506 -> PM10 :29.20
11:00:47.590 -> Umidità: 56.50 %, Temp: 26.60 Celsius
11:00:48.609 -> 2381 ppm
11:00:49.829 -> R0 = 16.97
11:00:50.845 -> PM2.5: 27.00
11:00:50.845 -> PM10 :29.30
11:01:10.901 -> Umidità: 56.40 %, Temp: 26.60 Celsius
11:01:11.922 -> 2375 ppm
11:01:13.146 -> R0 = 16.80
11:01:14.141 -> PM2.5: 27.40
11:01:14.141 -> PM10 :29.70
11:07:50.446 -> PM10 :28.60
11:08:10.511 -> Umidità: 56.50 %, Temp: 26.70 Celsius
11:08:11.531 -> 2451 ppm
11:08:12.758 -> R0 = 16.42
11:08:13.747 -> PM2.5: 25.90
11:08:13.747 -> PM10 :28.30
11:08:33.815 -> Umidità: 56.50 %, Temp: 26.70 Celsius
11:08:34.834 -> 2446 ppm
11:08:36.087 -> R0 = 16.39
11:08:37.075 -> PM2.5: 25.60
11:08:37.075 -> PM10 :27.90
11:08:57.140 -> Umidità: 56.50 %, Temp: 26.70 Celsius
11:08:58.162 -> 2462 ppm
11:08:59.389 -> R0 = 16.38
11:09:00.407 -> PM2.5: 25.70
11:09:00.407 -> PM10 :28.00
11:09:20.474 -> Umidità: 56.50 %, Temp: 26.70 Celsius
11:09:21.459 -> 2474 ppm
11:09:22.721 -> R0 = 16.49
11:09:23.711 -> PM2.5: 25.80
11:09:23.711 -> PM10 :28.10
11:09:43.756 -> Umidità: 56.50 %, Temp: 26.70 Celsius
11:09:44.776 -> 2477 ppm
11:09:46.003 -> R0 = 16.34
11:09:47.018 -> PM2.5: 25.50
11:09:47.018 -> PM10 :27.80
11:10:07.083 -> Umidità: 56.50 %, Temp: 26.70 Celsius
11:10:08.101 -> 2467 ppm
11:10:09.321 -> R0 = 16.34
11:10:10.343 -> PM2.5: 25.60
```

figura 30

Dopo aver aperto la porta e le finestre (figura 31):

```

12:03:40.273 -> Umidità: 57.70 %, Temp: 26.50 Celsius
12:03:41.261 -> 2472 ppm
12:03:42.515 -> R0 = 14.10
12:03:43.501 -> PM2.5: 26.60
12:03:43.501 -> PM10 :28.70
12:04:03.591 -> Umidità: 57.70 %, Temp: 26.50 Celsius
12:04:04.601 -> 2484 ppm
12:04:05.849 -> R0 = 14.00
12:04:06.832 -> PM2.5: 26.30
12:04:06.832 -> PM10 :28.40
12:04:26.895 -> Umidità: 57.70 %, Temp: 26.50 Celsius
12:04:27.938 -> 2471 ppm
12:04:29.152 -> R0 = 14.31
12:04:30.161 -> PM2.5: 26.20
12:04:30.161 -> PM10 :28.20

12:23:23.464 -> Umidità: 51.50 %, Temp: 26.60 Celsius
12:23:24.504 -> 1197 ppm
12:23:25.727 -> R0 = 29.04
12:23:26.740 -> PM2.5: 32.00
12:23:26.740 -> PM10 :35.00
12:23:46.791 -> Umidità: 51.50 %, Temp: 26.60 Celsius
12:23:47.786 -> 1199 ppm
12:23:49.020 -> R0 = 29.23
12:23:50.043 -> PM2.5: 32.00
12:23:50.043 -> PM10 :35.00
12:24:10.090 -> Umidità: 51.50 %, Temp: 26.60 Celsius
12:24:11.108 -> 1191 ppm
12:24:12.357 -> R0 = 29.08
12:24:13.323 -> PM2.5: 32.40
12:24:13.323 -> PM10 :35.40
12:24:33.402 -> Umidità: 51.60 %, Temp: 26.60 Celsius
12:24:34.420 -> 1188 ppm
12:24:35.635 -> R0 = 29.37
12:24:36.631 -> PM2.5: 31.70
12:24:36.668 -> PM10 :34.60

```

figura 31

Valutiamo infine come cambiano i dati raccolti in quest'ultima stanza tra la camera con porta e finestre chiuse e una persona dentro (Situazione 1) e la stessa con porta e finestre aperte (Situazione 2):

	Situazione 1	Situazione 2	Variazione (2)
Umidità	56.5	51.5	-8.85%
Temperatura	26.7	26.6	--
CO2	2462	1191	-51.62%
R0	16.38	29.37	+79.3%
PM2.5	25.8	31.7	+22.86%
PM10	28.1	34.6	+23.13%

In conclusione è stato evidenziato come varia la concentrazione di diversi gas nell'aria a seconda della condizione nella quale vengono effettuate le misure (stanza arieggiata, con poche o tante persone).

Da questi dati potrebbe essere possibile valutare come variano i valori a seconda dell'aumentare del numero di persone e condurre un piccolo studio per capire (approssimativamente) il numero di persone presenti in una determinata stanza; ciò permette, in una situazione delicata come quella che stiamo vivendo adesso, di garantire il distanziamento sociale e dunque il rispetto delle normative imposte.

Questo è inoltre un buon punto di partenza per iniziare a lavorare con la sempre più diffusa tecnologia NB-IoT. Essa permette infatti di lavorare con diversi dispositivi anche per un lungo periodo in quanto implica un consumo della batteria molto ridotto e permette anche di contenere i costi totali del progetto.

Infine questo progetto ha evidenziato come affiancando la tecnologia NB IoT all'uso del protocollo di comunicazione MQTT si riescono ad ottenere ottime prestazioni nel campo della misurazione e dell'analisi dei dati.

Bibliografia

- [1] Effects of Indoor Air Pollution on Human Health (Commission of the European Communities Directorate General for Science, Research and Development Joint Research Centre – Environment Institute)
- [2] WHO guidelines for indoor air quality : dampness and mould
(World Health Organization 2009)
- [3] Principali inquinanti indoor e loro fonti (Ministero della Salute – salute.gov.it)
- [4] Sindrome dell'edificio malato - Sick building syndrome (Ministero della Salute – salute.gov.it)
- [5] Data sheet Sodaq Sara R412M (https://www.u-blox.com/sites/default/files/SARA-R4_DataSheet_%28UBX-16024152%29.pdf)
- [6] Data sheet MQ-137
(<https://datasheetspdf.com/pdf/904648/HANWEIELECTRONICS/MQ-137/1>)
- [7] Data sheet DHT22
(<https://html.alldatasheet.com/html-pdf/1132459/ETC2/DHT22/225/2/DHT22.html>)
- [8] Data sheet sds018
(<https://ecksteinimg.de/Datasheet/SDS018%20Laser%20PM20Product%20Spec%20V1.5.pdf>)
- [9] Data sheet MH-Z14
(https://www.openhacks.com/uploadsproductos/mh-z14_co2.pdf)
- [10] Narrowband IoT (NB-IoT) – gemalto.com
- [11] (<https://patentimages.storage.googleapis.com/17/e1/02/6d78804c2b0c9f/EP3182634A1.pdf>)
- [12] MQTT cos'è e come funziona - internet4things.it