



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea in INGEGNERIA ELETTRONICA

Studio del protocollo wireless LORA per applicazioni IoT

Study of the LORA wireless protocol for IoT applications

Relatore: Chiar.mo
Prof. Massimo Conti

Tesi di Laurea di
Stefano Di Labio

A.A. 2019/2020

Indice

1	Introduzione	3
2	Stato dell'arte	4
2.1	Principali ostacoli	4
2.2	Applicazioni dell'IoT	5
3	LoRaWAN	7
3.1	Modulazione LoRa	9
3.1.1	Layer fisici	9
3.2	Metriche di valutazione del banco di prova	10
3.3	Limiti di LoRaWAN e sviluppi futuri	11
3.4	SYNCHRONOUS LORA MESH	13
3.5	LoRa per il monitoraggio a basso costo dei sistemi ambientali	15
3.6	The Things Network	17
4	Hardware	19
4.1	STEVAL-STRKT01	19
4.1.1	Descrizione hardware steval	21
4.1.1.1	Modulo LoRa	21
4.1.1.2	Gestione energia	22
4.1.1.3	Memoria	24
4.1.1.4	Interfaccia	25
4.1.1.5	Pulsanti e LEDs	25
4.1.1.6	SWD e connettori di espansione	26
4.1.1.7	GNSS	26
4.1.1.8	Sensori	27
4.2	Gateway	29
5	Progetto	30
5.1	FP-ATR-LORA1 firmware	30
5.1.1	FP-ATR-LORA1: macchina a stati	30
5.1.2	Firmware utilizzato	32
5.2	Configurazione steval	37
5.3	Configurazione gateway	39
5.4	Analisi misure	40
6	Conclusioni	51

Riferimenti bibliografici

1 INTRODUZIONE

In questi tempi moderni è sempre più comune, e lo sarà sempre di più, sentire il termine “Internet of Things”. Costantemente tirato in ballo nelle più importanti ricerche che collegano la fisica degli oggetti al mondo digitale che hanno, e avranno, una considerevole influenza nel futuro dell’Internet. Un importante parte della moderna tecnologia che avanza con passi da gigante, o meglio dire in modo esponenziale, che sta portando a risultati estremamente notevoli in molti campi della scienza:

- | | | |
|-------------------|------------------------------------|---------------------------|
| -Agricoltura | -Videosorveglianza | -Robotica |
| -Smart grid | -Zootecnica | -Reti wireless di sensori |
| -Smart City | -Avionica | -Rilevamenti eventi |
| -Sistemi embedded | -Domotica | -Monitoraggio |
| -Telelettura | -Intelligent transportation system | |
| -Telemetria | -Industria automobilistica | |
| -Telematica | - Ingegneria biomedica | |

In generale un sistema IoT può essere definito come un insieme di dispositivi, architetture e servizi basati sull’utilizzo di oggetti “intelligenti”, così definiti perché hanno la capacità di connettersi ad internet sfruttando un mezzo di comunicazione che li permetta di inviare e ricevere informazioni. In questa tesi in particolare verranno studiati gli aspetti di un tipo di connettività a basso consumo chiamata LoRaWAN. LoRa sta per “ Long Range” cioè a lungo raggio dove si ha una buona copertura accompagnata ad una efficiente gestione dei consumi di energia elettrica. La struttura di questa tesi è composta in vari capitoli in cui nei primi si darà una panoramica esaustiva, storica ed funzionale di quello che risulta essere il concetto di LoRa e della connettività ad essa associata chiamata LoRaWAN; nei capitoli seguenti si mostreranno le caratteristiche dei dispositivi utilizzati per testare questo tipo di connettività e le prestazioni di quest’ultimi. Dopo di ciò si andranno a mostrare tutte le configurazioni dei suddetti dispositivi necessarie al progetto il quale rappresenta l’ultima parte, dove si spiegheranno e setteranno i dispositivi all’interno del sistema IoT che si andrà ad implementare. Come già detto, nell’ultima parte c’è tutto il lavoro relativo alle misure che si effettuano nella pratica per testare questo particolare sistema IoT.

2 STATO DELL'ARTE

Lo sviluppo delle reti di sensori WSN è di grande attualità ed è soggetto in diversi centri di ricerca. Le applicazioni che possono integrare queste reti WSN vanno dal militare al campo medico. Negli ultimi anni, c'è stato uno sviluppo costante dei protocolli di comunicazione WSN, supportato anche dallo sviluppo del concetto di Internet of Things. Anche se il concetto di Internet delle cose (IoT) è spesso menzionato nella letteratura professionale, c'è una certa ambiguità ad esso, a causa dei numerosi altri concetti che comporta, e anche per il suo nome [1]. Così, il termine è stato definito vari modi, per esempio l'IoT si riferisce ad una rete mondiale (WWW- World Wide Web) di oggetti interconnessi indirizzabili in modo univoco, basati su standard protocolli di comunicazione [2].

2.1 PRINCIPALI OSTACOLI

La letteratura professionale comprende una serie di documenti scientifici che presentano i diversi modelli concettuali e le applicazioni con cui i moduli WSN si comportano utilizzando l'infrastruttura IP di Internet come mezzo di comunicazione. Così, il concetto di IoT comprende anche le reti di sensori WSN. Il sito del concetto di IoT comprende anche le comunicazioni M2M (Machine-to-Machine). I principali ostacoli che intralciano lo sviluppo dell'internet of things sono:

Connettività: la sfida principale consiste nell'assicurare l'interconnettività tra i dispositivi costruiti da diversi produttori. L'uso di un unico protocollo di comunicazione non è una scelta praticabile, in luce dei vari canali di comunicazione e delle relative applicazioni. Pertanto, l'obiettivo principale è quello di garantire la interconnettività dei vari formati di dati su una scala mondiale. Altrettanto vitale è l'analisi degli standard utilizzati al giorno d'oggi nel concetto di IoT.

Gestione efficiente dell'energia: la maggior parte dei moduli WSN sono alimentati a batteria, e quindi c'è un costante bisogno di ridurre il consumo di energia. Dal momento che il costo di questi dispositivi è piuttosto bassa, la sostituzione della batteria non può essere considerato efficiente, in quanto questa operazione sarebbe più costosa che la sostituzione dell'intero modulo. Pertanto, il ciclo di vita della batteria dovrebbe essere di decenni. Ciò comporta lo sviluppo di sofisticate strategie di comunicazione focalizzate sull'efficienza energetica. Un'altra possibile soluzione sarebbe il miglioramento di tecniche di immagazzinamento di energia.

Sicurezza: la fornitura di elevati livelli di sicurezza dovrebbe essere preso in considerazione. Pertanto, è altamente necessario sviluppare nuovi protocolli di autenticazione e crittografia che sono in grado di utilizzare un hardware specifico di limitati nodi WSN.

Complessità: L'integrazione della comunicazione è anche un compito molto complesso. I ricercatori sono ora alla ricerca di nuovi modi per aumentare l'efficienza e facilitare la programmazione di questi moduli.

Sviluppo rapido: il concetto di IoT è un concetto di sviluppo straordinariamente veloce, ogni giorno centinaia di nuovi nodi vengono aggiunti. Molte delle difficoltà non sono ancora note, in quanto si verificano con l'aumento del numero di nodi dell'internet degli oggetti e con la maggiore complessità delle applicazioni utilizzate per il trasferimento di informazioni a distanza nel cloud. La visione dell'internet degli oggetti è in costante evoluzione e in via di sviluppo.

2.2 APPLICAZIONI DELL'IOT

L'internet degli oggetti comporta lo sviluppo di un'invisibile e intelligente rete che può essere controllata e programmata. I dispositivi utilizzati su questa rete utilizzano la tecnologia embedded, che permette loro di comunicare direttamente o indirettamente utilizzando l'infrastruttura Internet [6]. Finora, l'universo dei dispositivi IoT comprende circa 5 miliardi di moduli interconnessi. Le stime attuali mostrano che, entro il 2020, il numero di questi dispositivi supererà il 50 miliardi [6]. Questo numero di dispositivi interconnessi sarà raggiunto solo con l'aiuto del progresso tecnologico consistente nell'improvvisazione/semplificazione e sviluppo di nuovi protocolli di comunicazione. Il concetto di IoT è ora diventato un'estensione dell'Internet. Il suo impatto sul modo in cui l'umanità si appropria ai problemi correnti saranno estesi negli anni a seguire. Alcune delle applicazioni dell'internet degli oggetti includono:

Smart Power Grid: Negli ultimi anni, sono stati fatti molti tentativi nel passare alle fonti di energia rinnovabili. Tutte queste fonti sono imprevedibili e geograficamente disperse, il che significa che sarà necessario implementare un sistema remoto di monitoraggio e trasferimento di informazioni.

e-Learning: L'internet degli oggetti deve prima di tutto facilitare l'accesso all'istruzione. L'apprendimento a distanza deve essere accessibile a tutti, e questo è reso possibile dalle reti WSN interconnesse che consentirebbe il trasferimento di informazioni a lunga distanza.

Smart Grid: L'uso più efficiente delle risorse è uno dei vantaggi dell'internet degli oggetti. L'impatto delle tecnologie WSN sulle smart grid è stato analizzato in molti documenti.

HealthCare: Le tecnologie WSN conferiscono la capacità di monitorare e proteggere le persone nelle loro case, quindi contribuendo ad aumentare la qualità dei servizi medici.

Protezione dell'ambiente: La protezione dell'ambiente è un problema di attualità che non deve essere trascurato. Le reti di sensori WSN abilitano il monitoraggio dei livelli di inquinamento e dell'impatto dell'umanità sulla natura.

Lo sviluppo delle reti WSN nell'internet degli oggetti è fondamentale, in quanto permette di migliorare la qualità della vita, anche contribuire all'abbassamento dei livelli di inquinamento rendendo il consumo di risorse più efficiente. Le sfide e i limiti del WSN sono:

- Difficoltà nell'integrazione di un gran numero di sensori;
- Problemi legati alla coesistenza, che si verificano in una zona congestionata da ambiente RF;
- Lo stato congestionato di un ambiente RF (Radio Frequenza) porta spesso a una bassa efficienza. Quindi, nuove soluzioni dovrebbero essere trovate per abilitare la selezione automatica e la regolazione del canale di comunicazione a seconda del livello di interferenza.
- L'assenza di modelli matematici che consentirebbero la simulazione di un numero molto elevato di nodi WSN;
- I simulatori WSN esistenti sono limitati per quanto riguarda la scalabilità. Così, al fine di simulare applicazioni complesse, si dovrebbe integrare l'interoperabilità tra i diversi produttori. Pertanto, il livello di difficoltà dell'internet degli oggetti è un concetto piuttosto alto.

3 LORAWAN

Il protocollo LoRaWAN è relativamente nuovo ed è diventato il focus di diversi centri di ricerca in tutto il mondo. LoRa (Long Range) è una tecnica di modulazione che permette il trasferimento a lungo raggio di informazioni con una bassa velocità di trasferimento. La modulazione LoRa è stata brevettata da Semtech Corporation. LoRa è un tipo di modulazione SS-Spread Spectrum, e la novità di questa tecnica consiste nell'utilizzo di un segnale che varia costantemente con la frequenza. Il vantaggio di questo metodo è che l'offset nel tempo e nella frequenza nel mittente e ricevente sono gli stessi, riducendo così notevolmente la complessità del ricevitore [3]. Una rete LoRaWAN (Long Range Network Protocol) è del tipo LPWAN (Low Power Wide Area Network) e comprende dispositivi alimentati a batteria che garantiscono la bidirezionalità nella comunicazione. La specifica LoRaWAN garantisce la perfetta interoperabilità tra gli oggetti dell'internet of things, senza la necessità di complesse implementazioni locali [4]. La rete LoRa è implementata utilizzando la stella topologia della rete. La struttura di un'architettura LoRa può essere separata in due parti: back-end e front-end. Il back-end consiste nel server di rete che memorizza le informazioni ricevute dai sensori. Il front-end è costituito dai moduli gateway e i nodi del dispositivo finale. I moduli gateway fungono da ponte tra i nodi del dispositivo finale e i nodi server di rete. Le informazioni tra il server di rete e i moduli Gateway vengono inviate attraverso la connessione IP. I moduli del dispositivo finale non hanno capacità di routing, e i messaggi vengono inviati direttamente al concentratore Gateway attraverso un meccanismo one hop. La tecnologia LoRaWAN ha velocità di trasferimento che variano da 0,3 kbps a 50 kbps. Al fine di massimizzare la durata della batteria, i dispositivi gestiscono la loro potenza di comunicazione RF e la velocità di trasferimento attraverso un Meccanismo Adaptive Data Rate (ADR).

La tecnologia LoRa è una LPWAN (Low-Power Wide-Area Wide-Area a bassa potenza Reti) standard, che propone il compromesso di riduzione della velocità di trasmissione dati a scapito di una comunicazione più lunga gamma. LoRa ha diversi tipi di dispositivi finali, a seconda del meccanismo di comunicazione utilizzato.

- **Classe A:** i dispositivi di classe A sono bidirezionali a livello di comunicazione. La fascia oraria durante la quale il dispositivo sta trasmettendo è seguito da due brevi intervalli di tempo durante il quale il dispositivo può ricevere informazioni. Dispositivi inclusi in questa categoria possono ricevere dati solo dal server dopo che avevano inviato le informazioni.

Così, i dispositivi di classe A garantiscono la più alta efficienza energetica. Se si desidera inviare i dati dal server, si attende il prossimo uplink dello slot programmato [4].

- **Classe B:** i dispositivi di classe B hanno bidirezionale capacità di comunicazione e una fascia oraria aggiuntiva che permette loro di ricevere dati. A parte le fasce orarie casuali durante la quale è possibile ricevere i dati che il dispositivo di tipo A consente, il dispositivo di tipo B può anche utilizzare una serie di ricevitori slot attivati da un messaggio di tipo Beacon inviato dal Gateway [4].

- **Classe C:** I dispositivi di classe C sono bidirezionali capacità di comunicazione e fasce orarie durante le quali può ricevere informazioni illimitate. L'unico momento in cui una classe Il dispositivo C non può ricevere informazioni è quando lo invia [4].

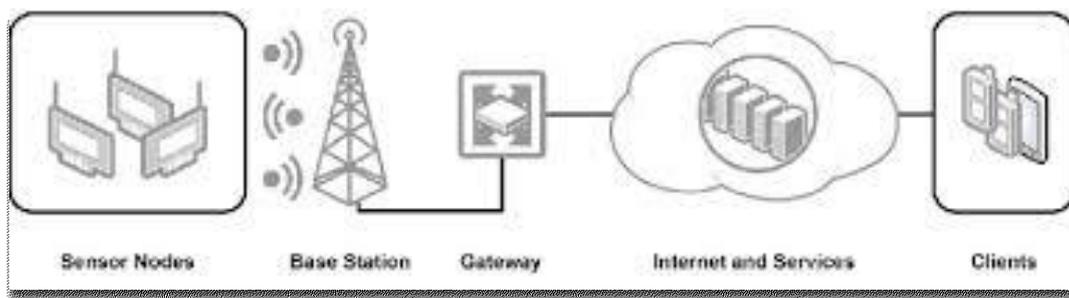


Figura 1: Architettura LoRaWAN [40]

Application				
LoRa® MAC				
MAC options				
Class A (Baseline)	Class B (Baseline)	Class C (Continuous)		
LoRa® Modulation				
Regional ISM band				
EU 868	EU 433	US 915	AS 430	—

Figura 2: LoRa section [40]

3.1 MODULAZIONE LORA

La tecnica di modulazione LoRa è di Chirp Spread Spectrum (CSS). Utilizza diversi fattori di diffusione della modulazione (SF) che vanno da SF7 a SF12. Questo meccanismo fornisce resistenza alle interferenze e al multipath fading. Così, è possibile regolare individualmente per ogni nodo il tasso di modulazione e la potenza di trasmissione. Un segnale chirp codifica un simbolo di informazione. Se il SF è aumentato il pacchetto sarà ridotto, con conseguente aumento della potenza sul e una maggiore distanza di comunicazione. La tabella 1 presenta i fattori di diffusione LoRa per 125 kHz di larghezza di banda. Come si può osservare se il fattore di diffusione è aumentato il bitrate è diminuito, il parametro time-on-air è aumentato, ma il limite SNR è notevolmente migliorato. Il numero di simboli codificati diminuisce mentre il fattore di diffusione è diminuito.

Tabella 1

Spreading Factor	Symbols /second	SNR limit	Time-on-air for 10 byte packet	Bitrate
7	976	-7.5	56	5469
8	488	-10	103	3125
9	244	-12.5	205	1758
10	122	-15	371	977
11	61	-17.5	741	537
12	30	-20	1483	293

LORA SPREADING FACTORS per 125 KHZ BANDWIDTH

3.1.1 Layer fisici

LoRa è un derivato della modulazione Chirp Spread Spectrum (CSS) con correzione degli errori in avanti (FEC) integrata [13]. LoRa utilizza bande ISM sub 1 GHz in Europa e Nord America e la sua natura a banda larga permette di compensare meglio un basso rapporto segnale/rumore (SNR)[14]. Questo permette a LoRa di demodulare i segnali anche quando sono 19,5 dB sotto il livello del rumore di fondo [15]. Il CSS permette di un campo di comunicazione più lungo rispetto al Frequency-Shift Keying (FSK) senza un aumento del consumo di energia [9].

La trasmissione a livelli di potenza più elevati aumenterà il raggio d'azione del nodo Lora. I nodi possono regolare la loro potenza di uscita per soddisfare requisiti normativi. In Europa +14 dBm è il valore massimo di potenza trasmessa tranne che nella banda G3 (+27 dBm) [9].

Le LoRaWAN distribuite in Europa hanno una larghezza di banda di canale di 125 kHz o 250 kHz e anche una singola trasmissione FSK è disponibile il quale fornisce una maggiore velocità di trasmissione dati [10]. I data rates sono dipendenti dalla regione (restrizioni normative) e dal fattore di diffusione (SF). Aumento del fattore di diffusione migliora il SNR ma comporta tempi di trasmissione più lunghi [11]. L'utilizzo di una maggiore larghezza di banda riduce i tempi di trasmissione ma riduce la massima sensibilità del ricevitore [15]. Calcoli effettuati in [16] hanno rivelato che quando un singolo gateway deve servire molti dispositivi, la maggior parte dei quali deve essere vicino al gateway ($SF = 7$) come solo pochi nodi con il massimo SF possono essere supportati data il loro lungo tempo di trasmissione. LoRa utilizza il FEC per consentire il recupero dalla trasmissione errori dovuti a scoppi di interferenze, ma l'uso di FEC aggiunge un po' di codifica in alto [17]. I tassi di codifica di LoRa sono $4/(CR + 4)$ con $CR \in \{1, 2, 3, 4\}$. L'intestazione di un pacchetto LoRa e il suo controllo di ridondanza ciclica (CRC) sarà sempre trasmessa utilizzando un CR di 4/8 e il carico utile con il suo opzionale CRC alla velocità di codifica scelta [17]. Quando LoRa trasmette con un BW di 125 kHz e un SF di 11 o 12 può essere abilitata un'ottimizzazione della velocità di trasmissione dati bassa. Ciò riduce l'impatto sulla trasmissione a causa della deriva nella frequenza di riferimento dell'oscillatore, ma aggiunge ulteriori dati in alto [17]. LoRa può rilevare l'attività del canale utilizzando Carrier Activity Rilevamento (CAD) [7]. Questo è più veloce del Received Signal Strength Indicator identification (RSSI) e può differenziare tra il rumore o un segnale LoRa desiderato [14].

3.2 METRICHE DI VALUTAZIONE DEL BANCO DI PROVA

Nel decidere se una LoRaWAN sarà la più adatta soluzione le capacità di questa tecnologia devono essere comprese. Queste capacità possono essere trovate attraverso la simulazione o attraverso i banchi di prova. Prima che questo possa iniziare una chiara definizione di metriche di valutazione è necessario per confrontare accuratamente i risultati. Metriche di valutazione suggerite per i banchi di prova LoRa e LoRaWAN può essere visualizzato nella tabella 2. Inoltre, i seguenti dovrebbero da tenere presente: l'SNR in un nodo avrà un impatto di il Bit Error Rate (BER) del nodo che a sua volta influisce il Packet Error Rate (PER) [20]. Un SF scelto avrà un impatto sulla velocità di trasmissione massima, sulla dimensione massima del carico utile e la quantità di tempo in cui un nodo deve essere spento prima che trasmettere di nuovo [18].

Tabella 2

Metric	Explanation
Numero dei gateways	Un LoRaWAN con un solo livello funzionerà in modo diverso da uno con più livelli
Numero dei nodes	L'aumento del numero di nodi richiede maggiori prestazioni della LoRaWAN.
RSSI	L'RSSI misurato da un nodo fornisce un'indicazione del campo di copertura del gateway e di quanto sarà affidabile la comunicazione.
SNR	L'SNR sperimentato da un nodo fornisce anche un'indicazione di quanto sarà affidabile la comunicazione.
Distanza dal gateway	Le misure di distanza possono essere combinate con le misure di connettività (RSSI, SNR, PER) per determinare il campo effettivo di un gateway.
Dimensioni Payload	Idealmente le prestazioni della rete sono state valutate per diverse dimensioni del carico utile in quanto questi impatti hanno un impatto sui tempi di trasmissione.
Potenza trasmessa	La trasmissione a livelli di potenza più elevati (dBm) può raggiungere una portata maggiore, ma si applicano restrizioni normative.
Affidabilità Network	L'affidabilità della rete può essere definita in diversi modi: numero di pacchetti ricevuti con successo, il numero di pacchetti validi ricevuti o l'esame del PER.
Consumo potenza	I valori di consumo di energia per i motes possono essere utilizzati per stimare la durata di vita della rete.

3.3 LIMITI DI LORAWAN E SVILUPPI FUTURI

Esaminando LoRaWAN dal punto di vista di modello di interconnessione dei sistemi aperti (OSI) si scopre che LoRaWAN è principalmente un protocollo di livello MAC con un'architettura di rete impostata e il resto dello stack è lasciato allo sviluppatore [9], [25]. Avanzate caratteristiche come un meccanismo di auto-frammentazione dei carichi utili che è più grande della dimensione massima che dovrà essere sviluppata. I gateway possono utilizzare l'ADR per controllare la velocità dei dati degli end-device ma non si conosce ancora il modo ottimale per eseguire l'ADR [8]. I dispositivi di classe A stabiliscono due finestre di

ricezione solo dopo aver trasmesso un messaggio. Per impostazione predefinita la finestra due è fissa di utilizzare 869,525 MHz e un SF di 12 in Europa [26]. Un potenziale scenario potrebbe presentarsi quando un gateway occupato è costretto ad utilizzare la seconda finestra per rispondere ai dispositivi, con conseguente allungamento dei tempi di trasmissione durante i quali un canale è occupato [8]. Inoltre, in Europa molti dei canali sono limitati a un ciclo di lavoro massimo dell'1% che limita anche il tempo di permanenza in aria per i dispositivi. C'è una finestra di ritardo tra quando un gateway ha ricevuto un messaggio e quando la prima finestra di ricezione del nodo del mittente si apre. Durante questo tempo, il gateway potrebbe continuare ad ascoltare il canale e iniziare a ricevere un messaggio da un altro nodo [8]. LoRa utilizza diverse bande sub 1 GHz a seconda della vostra regione. Ciò comporta problemi di configurazione specifici della regione come dispositivi finali che devono essere riconfigurati o rilevare la loro regione. La specifica LoRaWAN suggerisce metodi per i dispositivi finali per determinare la loro regione. I suggerimenti includono l'aggiunta di GPS hardware per terminare i dispositivi, equipaggiando i segnalatori con GPS e trasmettendo le loro coordinate o istruendoli a trasmettere una lista di frequenze di collegamento. La programmazione dei radiofari per la trasmissione di un elenco di frequenze di connessione evita il costo dell'hardware del GPS, ma richiede la presenza di nodi di segnalazione [19]. La redditività a lungo termine di LoRaWAN rimane in discussione. Attualmente esiste un solo produttore di LoRa (Semtech) e mentre le bande non autorizzate si congestionano sempre più, la rete può non fornire più le prestazioni che aveva in origine. Le reti LoRa pubbliche e private interferiranno tra loro, come soprattutto negli ambienti urbani in cui tutti i dispositivi finali sono nelle immediate vicinanze. Questo è un problema diffuso per tutte le soluzioni wireless che utilizzano le bande senza licenza. L'uso delle bande protette con licenza avranno sempre un vantaggio. Questo problema è in parte alleviato dalla configurabilità dell'SF di LoRa, BW e potenza di trasmissione e CAD [11]. Il CAD di LoRa soffre tuttavia di falsi positivi quando sono presenti più combinazioni SF/BW (più reti in un'area) [11]. Il supporto per gli aggiornamenti del firmware over the air sembra essere limitato, se non del tutto impossibile. Gli aggiornamenti del firmware sono limitati dalla restrizione del ciclo di lavoro dell'1% e dalla classe A i nodi sono solo messaggi unicast. Le specifiche LoRa per i dispositivi di classe B lasciano la configurazione dei gruppi multicast al livello di applicazione o come parte della personalizzazione del nodo [19]. Il formato del sito del messaggio multicast non richiede alcuna conferma della richiesta per i messaggi multicast. Allo stesso modo il supporto per gli aggiornamenti dei over the air sono carenti, questo è uno dei principali motivi di preoccupazione per applicazioni sensibili alla sicurezza, come le

infrastrutture critiche. Semtech ha annunciato in un comunicato stampa del giugno 2016 che le applicazioni native sulla geolocalizzazione sono state aggiunte a LoRa. Informazioni limitate su questa funzione è disponibile ma il comunicato stampa riporta che la versione 2 dei gateway LoRaWAN sono obbligatori. LoRa supporta la geolocalizzazione utilizzando timestamp e algoritmi di tempo differenziale di arrivo [5]. I nodi mobili che tentano la geolocalizzazione riceveranno risultati imprecisi in quanto, soprattutto in Europa, il ciclo di lavoro dell'1% limiterà la frequenza dei tentativi di geolocalizzazione. Le funzioni di geolocalizzazione aumenteranno il numero di tentativi di casi d'uso, ma sono necessari algoritmi di localizzazione sicuri per impedire che nodi fraudolenti o maligni disturbino la rete. Quando i dispositivi si ricongiungono ad una rete, ritornano alla rete impostazioni MAC predefinite [26]. Il server di rete quindi devono riaggiustare le modifiche a queste impostazioni su ogni procedura di giunzione. Di conseguenza, un importante picco di traffico per la rete server si verifica se una grande quantità di dispositivi hanno improvvisamente bisogno di tornare a far parte di una rete. Questo potrebbe accadere quando un gateway ha perso potenza ed è servito come unico gateway per tutti i nodi della sua area di copertura. Il protocollo LoRaWAN consente ai dispositivi finali (Classe A) di decidere quando inviare gli ACK per i messaggi ricevuti e quando ricevere i messaggi in attesa [26]. Questo permette di terminare dispositivi per garantire che rimangano entro i limiti del ciclo di lavoro. Tuttavia, i gateway non hanno un limite superiore su quando i messaggi possano essere riconosciuti o quando i dispositivi riceveranno eventuali pendenti messaggi(i). Le applicazioni con requisiti di tempo critici dovranno progettare con attenzione il loro firmware del nodo finale, scegliere una classe LoRaWAN diversa o optare per un LPWAN diverso.

3.4 SYNCHRONOUS LORA MESH

La recente tecnologia LPWAN può essere ulteriormente ingegnerizzata per adattarsi al particolare scopo del monitoraggio distribuito, nelle reti di drenaggio urbano e oltre. A quest'ultimo sono legate molte limitazioni ed è per questo che è stato creato un metodo di acquisizione con LoRaWAN chiamato "synchronous LoRa mesh", un'architettura e un concetto nuovo per controllo dell'accesso medio. Le caratteristiche principali visibili sono i nodi ripetitori intermedi (RN) che permettono la formazione di singole sottoreti, cioè cluster di nodi di sensori (SN). Le RN hanno una connettività LoRaWAN fuori terra verso un gateway e fungono da radice per i nodi figli (sensori), che sono posizionati a distanza oltre la

copertura LoRaWAN. A tal fine, estendiamo la architettura originale LPWAN da un semplice tipo a stella verso una topologia a maglie gerarchiche, che consente una trasmissione multi-hop e quindi un instradamento più flessibile ed esteso copertura. A differenza dei gateway LoRaWAN, gli RN funzionano a batterie e non necessitano di accesso a Internet. L'hardware e il firmware degli RN e degli SN sono identici. Entrambi funzionano con lo stesso tipo di batterie, ma i nodi soddisfano diversi ruoli operativi. La comunicazione tra una RN e SNs corrispondenti è sincronizzato con la temporizzazione ad alta precisione, consentendo il cambio coordinato tra la modalità sleep e svegliarsi entro l'intervallo di trasmissione. Lo scambio di dati tra LoRaWAN e il synchronous LoRa meshBsi basa sulla stessa tecnologia dello strato fisico, ma vengono utilizzati parametri di trasmissione leggermente diversi (Tabella 3). Per semplicità, la LoRa mesh utilizza un fattore di diffusione fisso e una potenza di trasmissione fissa. Entrambi possono essere configurati prima dell'implementazione degli RN e degli SN. In contrasto con questo semplice approccio, i dispositivi con rete LoRaWAN possono informare il server di rete se dovessero controllare la loro potenza di trasmissione e il fattore di diffusione o no. Nel caso in cui questa funzione sia abilitata, il server di rete cerca di abbassare il fattore di diffusione di un dispositivo quando la qualità del segnale ricevuto è sufficiente. Questo meccanismo si chiama controllo adattativo della velocità di trasmissione dati (ADR) [33].

Tabella 3 [34]

Parameter		LoRaWAN (EU863-870)	Synchronous LoRa mesh
Spreading Factor	SF	7 – 12 (variable)	9
Bandwidth	BW	125 kHz	125 kHz
Preamble length	nPreambleSymbols	8	8
Transmission Power	P _{TX}	Variable	+14 dBm
Coding Rate	CR	4/5	4/5
CRC checking		Header & Payload	Header & Payload

LoRaWAN e LoRa mesh utilizzano entrambi come formati dei pacchetti lo stesso formato LoRa utilizzato dalla SemTech[32]. LoRaWAN comprende un campo MAC Header (MHDR) a un byte in prima posizione contenente informazioni sul tipo di messaggio e la versione di protocollo utilizzato. Il campo Frame Header field (FHDR) contiene l'indirizzo del dispositivo

ricevente, e altri controlli e informazioni di riconoscimento, mentre il campo porta (FPort) definisce quale delle 223 porte disponibili viene utilizzata per il pacchetto consegna. Il carico utile effettivo dell'applicazione è incluso nel Campo MAC Frame Payload Encryption (FRMPayload). L'integrità dei dati in LoRaWAN è controllata da un messaggio di controllo integrità utilizzando il campo MIC. Al contrario, il formato sincrono del carico utile della mesh LoRa è più semplice. Un'intestazione Frame a cinque byte contiene informazioni su destinazione, indirizzo di origine e il tipo di pacchetto corrente (ad es. dati o sincronizzazione). A seconda di questo tipo di pacchetto, il carico utile contiene una quantità variabile o fissa di byte di dati. Nel caso di un pacchetto del tipo di dati, le informazioni sulla lunghezza sono consegnate all'interno del campo del carico utile.

3.5 LORA PER IL MONITORAGGIO A BASSO COSTO DEI SISTEMI AMBIENTALI

LoRa viene utilizzato per creare un collegamento di comunicazione a lungo raggio. Esso si basa sulla modulazione dello spettro diffuso del chirp per mantenere basse caratteristiche di potenza, e aumenta significativamente il raggio di comunicazione coprendo centinaia di chilometri quadrati [34]. Uno dei maggiori vantaggi della rete LoRa è che, copre una distanza di diversi chilometri e non è necessario avere accesso a Internet presso i nodi dei sensori. Inoltre, questa rete può essere distribuita in ambienti urbani o in aree non collegate senza costi di comunicazione e fornisce comode funzionalità per le applicazioni IoT [34]. Per queste ragioni LoRa è usata per implementare il monitoraggio in una rete ad ampio raggio indipendente a bassa potenza. Nell'ambito di queste misure viene applicato un metodo definito come metodo di riduzione dei dati descritto qui di seguito:

Il processo inizia con la lettura delle misure dai sensori, le fasi intermedie sono focalizzate nella riduzione dei dati e il tempo di acquisizione usando metodi semplici, e termina con l'invio dei dati ad un nodo gateway. Da questo nodo i dati vengono inviati a un servizio di cloud storage. Queste fasi intermedie vengono eseguite al nodo del sensore in un microprocessore a bassa potenza prima di inviare i dati al gateway e sono spiegate in dettaglio qui di seguito.

- 1) Filtro 1-D Kalman: Il segnale di uscita del sensore PM ha una elevata componente di rumore che può essere caratterizzata come rumore gaussiano. Pertanto, è necessario

implementare un 1-D Filtro Kalman. L'equazione del filtro è mostrata in (1), dove \hat{X}_k è il valore stimato, Z_k è il valore misurato e K_k è il Guadagno Kalman

$$\hat{X}_k = K_k Z_k + (1 - K_k) \hat{X}_{k-1}. \quad (1)$$

Per calcolare il K_k è necessario conoscere o stimare la covarianza di errore R del processo P . Per il sensore PM la covarianza di errore dell'osservazione può essere ottenuta con la funzione di densità di probabilità dei dati, mentre il sensore PM è statico. Se il valore di R è noto, viene utilizzato il processo mostrato in (2), dove m è un vettore di precedenti campioni grezzi acquisiti utilizzati per ottenere la natura del rumore.

$$\begin{aligned} K_k &= \frac{P_k}{P_k + R} \\ \hat{X}_k &= \hat{X}_{k-1} + K_k (Z_k - \hat{X}_{k-1}) \\ P_k &= (1 - K_k) P_{k-1}. \end{aligned} \quad (2)$$

2) Downsampling: Il filtro Kalman applica una levigatura al segnale PM2.5. Il downsampling viene eseguito in un tale modo in cui un campione è registrato solo quando ci sono significative variazioni del segnale PM del campione precedente, prelevando in considerazione di una soglia stabilita. Pertanto, l'importo di dati inviati per la memorizzazione nel cloud diminuisce.

3) Fusione dei dati e scalatura dei dati: La misurazione di dati ridondanti utilizzando diversi sensori di alcuni sensori ambientali variabili permette di ottenere una misura correlata utilizzando la fusione dei dati. Per il nodo implementato, alcune variabili sono misurate utilizzando due sensori di marche diverse. Le misurazioni effettuate con entrambi i sensori sono mediate per generare un unico valore. Un'altra tecnica usata in questo processo è la scalatura dei dati, che consiste nel ridurre i dati ad un intervallo previsto e rimuovere i decimali irrilevanti. Questo processo riduce i dati senza perdere informazioni rilevanti.

3.6 THE THINGS NETWORK

The Things Network è l'infrastruttura nata con lo scopo di gestire centralmente i vari dispositivi ed i dati inviati all'interno di una rete IoT basata su protocollo LoRaWAN. Tipicamente in una rete IoT il gateway svolge la funzione di ponte tra specifici protocolli radio e internet, dove i pacchetti IP provenienti dai dispositivi vengono inoltrati se essi lo supportano. Nei dispositivi LoRaWAN invece è necessario una specie di elaborazione o routing prima che i messaggi possano essere consegnati ad un'applicazione; qui entra in gioco The Things Network il quale ha il compito di gestire le elaborazioni necessarie ed eseguire tutte le funzioni di routing in modo decentralizzato e distribuito, in questo modo ogni implementazione è in grado di creare la propria rete e la propria parte del back-end, inoltre può partecipare alla rete della comunità globale.

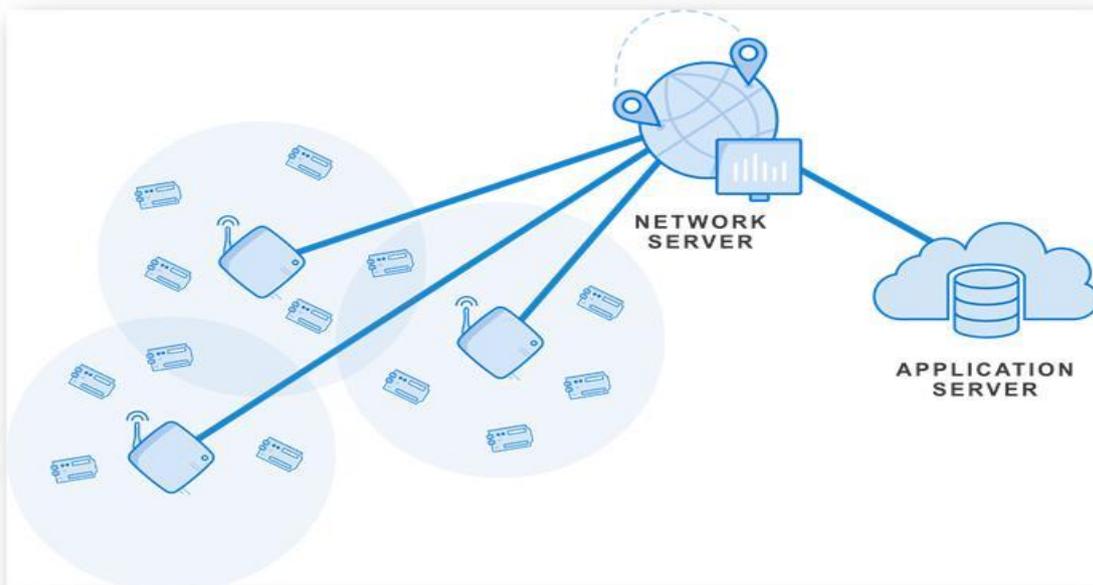


Figura 3: Architettura di rete LoRa - The Things Network[31]

All'interno della rete i dispositivi trasmettono messaggi tramite il protocollo radio LoRa dove il gateway riceve i messaggi e li inoltra al router. Ogni router è connesso ad uno o più broker che rappresentano la parte centrale de TTN; il Broker è un microservizio che identifica il dispositivo, deduplica il traffico e invia il pacchetto all'handler sul quale l'applicazione è registrata; in pratica ha il compito di associare un dispositivo ad un'applicazione, inoltrare i messaggi di Uplink all'applicazione corretta, inoltrare i messaggi di Downlink al router corretto che a sua volta li inoltra a un gateway. Anche l'Handler è un microservizio

responsabile della gestione dei dati di applicazioni ed ha funzione di crittografia e decrittografia dei dati.

La scelta del numero di gateway da utilizzare è relativa al traffico di rete in quanto è necessario garantire scalabilità e ridondanza, infatti con più gateway è possibile ridurre la distanza rispetto ai dispositivi e quindi aumentare la velocità di trasferimento dati. Tutto ciò aumenta la capacità della rete supportato anche dall'utilizzo dell'algoritmo Adaptive Data Rate (ADR) che esegue uno scaling automatico.

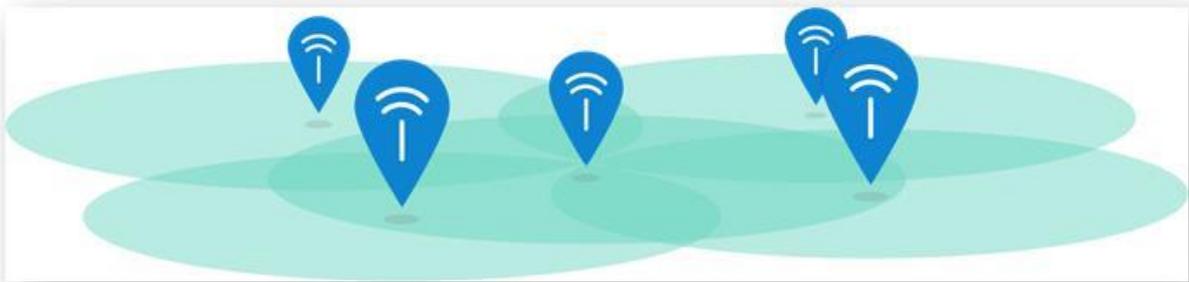


Figura 4: Distribuzione gateway [30]

La Things Network è una rete pubblica altamente sicura che supporta la vera crittografia end-to-end, le attenuazioni contro i vari attacchi man-in-the-middle e il supporto di diverse chiavi di crittografia a 128 bit per ogni singolo dispositivo finale. LoRaWAN applica il controllo dell'integrità dei messaggi AES a 128 bit e la crittografia del carico utile. I payload sono completamente criptati tra il Nodo e il componente Handler del backend. Ciò significa che si può scegliere di gestire il proprio Handler privato e disporre di una vera e propria crittografia end-to-end. I componenti Router e Broker instradano i dati sulla base di metadati pubblici e non possono decifrare il carico utile effettivo.

4 HARDWARE

4.1 STEVAL-STRKT01



Figura 5: STEVAL-STRKT01[29]

La STEVAL-STRKT01 LoRa® IoT tracker è progettata e ottimizzata per implementare le più recenti tecnologie di tracking delle applicazioni IoT come la tracciabilità di beni, persone e animali e la gestione della flotta. La scheda di valutazione semplifica la prototipazione, la valutazione e lo sviluppo di soluzioni innovative per il tracker. Viene fornito con software completo, librerie di firmware, strumenti, batteria, cavi e custodia in plastica.

Grazie all'STM32L072CZ integrato nel modulo LoRa® CMWX1ZZABZ-091 (di Murata), lo STEVAL-STRKT01 permette l'acquisizione della posizione, gestione geofence e registrazione dei dati dal modulo GNSS Teseo-LIV3F e monitoraggio del movimento (LIS2DW12) e sensori ambientali (HTS221 e LPS22HB).

La scheda inoltre trasmette e riceve dati, configurazioni ed eventi da e verso il cloud su una rete LoRaWAN™, oppure memorizza i dati a livello locale nell'EEPROM M95M02-DR.

La STEVAL-STRKT01 è una soluzione a batteria LiPo e implementa strategie a bassa potenza grazie ad una maggiore potenza/ progettazione della gestione delle batterie, basata sul caricabatterie STBC02 e sul convertitore step-down ST1PS01, per garantire una lunga durata autonomia della batteria. Lo STUSB1600A indirizza la gestione delle porte USB di tipo C a 5 V e offre pin di protezione ad alta tensione.

Il pacchetto STEVAL-STRKT01 include:

- una scheda di valutazione;
- una custodia in plastica;
- un cavo USB di tipo C;
- un adattatore USB da Tipo A a Tipo C;
- un cavo di programmazione;
- una batteria;
- un'antenna;
- una banda di plastica e un supporto in plastica.



Figura 6: Pacchetto componenti steval [29]

4.1.1 Descrizione hardware steval

La scheda di valutazione STEVAL-STRKT01 comprende i seguenti dispositivi:

- STM32L072CZ- ARM Cortex-M0+ MCU a bassissima potenza con 192 Kbyte Flash, 32 MHz CPU, USB
- Teseo-LIV3F - piccolo modulo GNSS
- STBC02 - Caricabatterie lineare agli ioni di litio con LDO, interruttori di carico e generatore di reset
- STUSB1600A - Controllore USB di tipo C (con protezione da corto a VBUS)
- M95M02-DR - bus SPI seriale da 2 Mbit EEPROM

I componenti principali sono descritti nelle sezioni seguenti.

4.1.1.1 Modulo LoRa

Lo STEVAL-STRKT01 incorpora il modulo CMWX1ZZABZ-091 LoRa®/Sigfox™ (by Murata) che permette di sviluppare applicazioni con l'STM32L072CZ e la connettività RF LoRa®/Sigfox™ in un unico modulo. Il CMWX1ZZABZ-091 ha la serie completa di funzioni disponibili nella serie STM32L0 e offre una potenza ultra-bassa e le caratteristiche LoRa® RF:

- MCU integrate a bassissimo consumo della serie STM32L072CZ, basate sul core ARM® Cortex®-M0+, con 192 Kbyte di memoria Flash, 20 Kbyte di RAM, 20 Kbyte di EEPROM
- Gamma di frequenza RF: 860 - 930 MHz
- USB 2.0 FS
- 4 canali, ADC a 12 bit, 2xDAC
- Timer a 6 bit, LP-UART, I2C e SPI
- Ricetrasmittitore SX1276 integrato
- Modulazioni LoRa®, FSK, GFSK, MSK, GMSK e OOK
- +14 o +20 dBm di potenza di uscita selezionabile
- 157 dB di budget massimo per i link

- Bit rate programmabile fino a 300 kbit/s
- Alta sensibilità: fino a -137 dBm
- Frontale a prova di proiettile: IIP3 = -12,5 dBm
- Immunità di blocco di 89 dB
- Bassa corrente RX di 10 mA, 200 nA di ritenzione del registro
- Sintetizzatore completamente integrato con una risoluzione di 61 Hz
- Sincronizzatore di bit integrato per il recupero dell'orologio
- Riconoscimento delle parole in sincronia
- Rilevamento del preambolo
- 127 dB+ campo dinamico RSSI

Il modulo Murata LoRa® incorpora il proprio TCXO che funziona a 32 MHz quando è abilitato. Il TCXO è controllato dal pin STM32 PA12 (R106 montato - R107 non montato) o per default sempre abilitato (R106 non montato - R107 montato). Quando l'STM32 necessita di un preciso orologio esterno ad alta velocità, il perno dell'orologio TCXO_OUT può alimentare il modulo PH0_OSC_IN pin mediante il montaggio di R105. Lo STEVAL-STRKT01 può essere dotato, se necessario, di un oscillatore a cristalli esterni: un oscillatore a 24 MHz con 20 I condensatori pF possono essere aggiunti alla scheda (Y100, C108 e C109 non sono montati di default).

4.1.1.2 Gestione energia

Lo STEVAL-STRKT01 mira a un consumo energetico molto basso e ad un'elevata efficienza energetica. Un'analisi accurata e la definizione del funzionamento del sistema e delle strategie di gestione dell'energia non può essere ottenuta semplicemente selezionando componenti ad alta efficienza energetica.

Il sottosistema energia è stato integrato dai seguenti dispositivi principali:

- STBC02
- ST1PS01EJR
- batteria LiPo ricaricabile da 480 mAh

L'STBC02 è un dispositivo di gestione dell'alimentazione altamente integrato, che incorpora un caricabatteria lineare, un 150 mA LDO, 2 interruttori di carico SPDT, un blocco intelligente di reset/watchdog e un modulo di circuito di protezione (PCM) per evitare che la batteria di essere danneggiato in condizioni difettose. L'STBC02 implementa gli algoritmi CC/CV per caricare la batteria, nonché le modalità di carica rapida e di precarica le cui correnti possono essere programmate in modo indipendente utilizzando resistenze dedicate. La corrente di terminazione è impostata di default, essendo il 5% della corrente di carica rapida programmata, ma può essere anche fissa a valori diversi. Allo stesso modo, il valore della tensione di galleggiamento della batteria è programmabile e può essere impostato su un valore fino a 4.45 V. Inoltre, implementa controlli della batteria sotto/sopra la temperatura. L'STBC02 si spegne automaticamente dalla batteria collegata quando il pin IN non è collegato ad un valido fonte di alimentazione (modalità batteria) e dispone anche di un caricabatterie abilita l'ingresso per interrompere il processo di carica in qualsiasi momento.

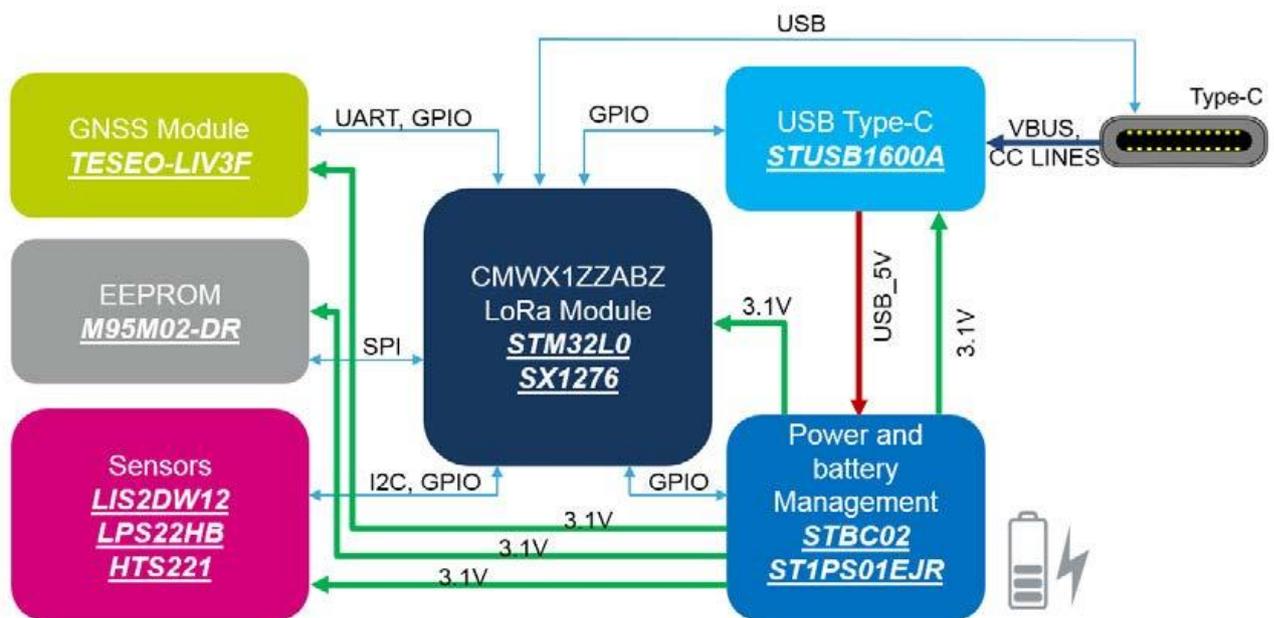


Figura 7: Diagramma a blocchi STEVAL-STRKT01 [29]

I sottocircuiti che possono essere attivati in base ai compiti programmati dall'applicazione sono:

- Sotto circuito GNSS TESEO-LIV3F
- Sottocircuito EEPROM
- Sottocircuito del controllore USB di tipo C

- Sottocircuito del sensore

4.1.1.3 Memoria

I dispositivi M95M02 sono memorie programmabili cancellabili elettricamente (EEPROM) organizzate come 262144 x 8 bit, a cui si accede attraverso l'autobus SPI. L'M95M02 può funzionare con un campo di alimentazione da 1,8 a 5,5 V. Questi dispositivi sono garantiti attraverso un campo di temperatura di -40/+85 °C. L'M95M02-DR offre una pagina aggiuntiva, denominata Identification Page (256 byte), che può essere utilizzata per memorizzare parametri sensibili dell'applicazione da bloccare (in seguito) in modo permanente in sola lettura. Il modulo M95M02-DR è alimentato dal dominio MEM_VDD attraverso uno degli interruttori incorporati nel STBC02 (SW1_OB). Si interfaccia con il modulo LoRa tramite SPI (pin del modulo LoRa PB12:PB15, cioè SPI2). I dati del sensore possono essere memorizzati nell'EEPROM; questo viene gestito da un modulo firmware, chiamato "log manager" che gestisce la formattazione, il salvataggio e il recupero dei dati. Il gestore del registro si occupa di tre tipi di dati: normali, di sistema e critici. Nell'applicazione sviluppata viene implementato solo il normale salvataggio dei dati. Per ulteriori informazioni, vedere il manuale d'uso di FP-ATR-LORA1.

La tabella seguente mostra il formato dei dati del log manager.

Size	Data	Units
32b	Timestamp	s from board start
16b	Temperature	°C * 100
16b	Pressure	hPa/10
16b	Humidity	percentage * 10
32b	Latitude	Sexagesimal degree converted to decimal.
32b	Longitude	Sexagesimal degree converted to decimal.
32b	Altitude	m

4.1.1.4 Interfaccia

Le interfacce disponibili sono:

- CON501: connettore SWD
- CON502, CON503: connettori ad espansione
- J501 (non montato): connettore SWD esteso
- SW500: pulsante utente
- SW400: pulsante di accensione e di risveglio
- D500: LED utente
- CN500: Connettore USB TYPE-C

4.1.1.5 Pulsanti e LEDs

Lo STEVAL-STRKT01 ha due pulsanti:

1) SW500 (pulsante utente) posto sul lato superiore della scheda. Il firmware di default (FP-ATR-LORA1) implementa:

- una breve pressione per attivare il modulo LoRa
- premere a lungo per spegnere

2) SW400 per accendere la scheda quando è alimentata solo dalla batteria (tenere premuto il pulsante per 1.250 s). Dopo l'accensione, SW400 diventa un pulsante utente generico. FP-ATR-LORA1 lo usa per resettare il sistema.

Il LED D500 è il LED bianco dell'utente utilizzato da FP-ATR-LORA1 per fornire l'indicazione dello stato di trasmissione LoRa:

- Lampeggio lento (2 secondi) = non collegato
- Lampeggio veloce (0,5 secondi) = unito
- Lampeggio molto veloce (200 ms) della durata di circa 5/10 secondi = trasmissione in corso

Il LED D401 è il LED rosso collegato al caricabatteria. Quando l'USB è collegato ad un alimentatore valido, esso lampeggia a 4,1 - 6,2 - 8,2 - 8,2 - 10,2 - 12,8 - 14,2 - 16,2 Hz. Per il diverso tasso di lampeggio fare riferimento ai messaggi di debug.

4.1.1.6 SWD e connettori di espansione

CON501 enables SWD connection to program and debug STM32L072CZ.



Figura 8: Connettore STEVAL-STRK01 [29]

4.1.1.7 GNSS

Il minuscolo modulo GNSS Teseo-LIV3F rappresenta un sistema di navigazione satellitare globale, economico e facile da usare, che incorpora un circuito integrato di posizionamento a singolo stampo TeseoIII per il ricevitore di posizionamento standalone. Si tratta di un modulo compatto (9,7x10,1 mm) che fornisce una precisione superiore grazie ai 26 MHz di oscillatore a cristallo compensato in temperatura (TCXO) e un ridotto time-to-first fix (TTFF) con il suo oscillatore del clock a 32 KHz in tempo reale (RTC).

Caratteristiche principali di Teseo-LIV3F:

- Multiconstellazione simultanea
- 163 dBm di sensibilità di navigazione
- Posizionamento di precisione CEP di 1,5 m
- Flash integrata a 16 Mbit per la registrazione dei dati e l'aggiornamento del firmware
- Campo di tensione di alimentazione da 2,1 a 4,3 V
- Piccolo pacchetto LCC a 18 pin (9,7x10,1)
- Temperatura di esercizio (-40°, 85°C)

- Configurazione gratuita del firmware
- 17 μ W di corrente di standby e 75 mW di consumo energetico di inseguimento

Il modulo Teseo-LIV3F viene fornito con un firmware pre-programmato per eseguire tutte le operazioni GNSS, tra cui acquisizione, tracciamento, navigazione, geofence e uscita dati senza supporto di memoria esterna.

4.1.1.8 Sensori

I sensori ambientali e di movimento integrati sono dotati di capacità di potenza estremamente bassa e avanzate prestazioni in termini di precisione e caratteristiche digitali integrate. I seguenti sensori sono montati sulla scheda di valutazione STEVAL-STRKT01:

- U301: LIS2DW12 - Accelerometro a 3 assi
- U303: HTS221 - sensore di umidità e temperatura
- U304: LPS22HB - sensore di pressione

LIS2DW12

Il LIS2DW12 è un accelerometro lineare a tre assi a bassissima potenza e ad alte prestazioni appartenente alla famiglia “femto” che fa leva sui robusti e maturi processi di produzione già utilizzati per la produzione di accelerometri microlavorati. Ha un fondo scala selezionabile dall'utente di $\pm 2g/\pm 4g/\pm 8g/\pm 16g$ ed è in grado di misurare le accelerazioni (con dati di uscita velocità da 1,6 a 1600 Hz) e un buffer integrato a 32 livelli first-in, first-out (FIFO) che consente all'utente di memorizzare i dati per limitare l'intervento del processore ospite. La capacità di autotest integrata permette anche di controllare il funzionamento del sensore nell'applicazione finale. Il LIS2DW12 ha un motore interno dedicato per elaborare il rilevamento del movimento e dell'accelerazione, compresa la caduta libera, sveglia, riconoscimento a singolo o doppio tocco altamente configurabile, attività/inattività, rilevamento stazionario/momento, ritratto/rilevamento del paesaggio e orientamento 6D/4D. L'accelerometro è disponibile in un piccolo e sottile pacchetto di plastica land grid array (LGA) ed è garantito per operare in un campo di temperatura esteso da -40 a +85 °C.

HTS221

L'HTS221 è un sensore ultracompatto per l'umidità relativa e la temperatura. Esso comprende un elemento sensibile e un ASIC a segnale misto per fornire le informazioni di misura

attraverso interfacce seriali digitali. L'elemento sensibile è costituito da una struttura polimerica dielettrica planare a condensatore dielettrico in grado di rilevare le relative variazioni di umidità e viene prodotto utilizzando un processo ST dedicato. L'HTS221 è disponibile in un piccolo pacchetto top-holed cap land grid array (HLGA) garantito per operare su un campo di temperatura da -40 a +120 °C. Il sensore incorpora una periferica I²C, con indirizzo slave impostato su 0xBE. L'HTS221 è alimentato dal dominio di potenza SENS_VDD, collegato al dominio VDD_BUCK (uscita 3,3 V da ST1PS01) attraverso uno degli interruttori incorporati nell'STBC02 (STBC02_SW2_OB). Questo permette strategie di bassa alimentazione implementate nel firmware, rimuovendo l'alimentazione dei sensori quando non vengono utilizzati. I pin del bus I²C (SENS_I2C_SCK_SWITCH e SENS_I2C_SDA_SWITCH) sono collegati ai pin del modulo LoRa PB8 e i perni PB9.

LPS22HB

LPS22HB è un sensore di pressione assoluta piezoresistivo ultracompatto che funziona come uscita digitale barometro. Il dispositivo comprende un elemento di rilevamento e un'interfaccia IC che comunica attraverso I²C o SPI dall'elemento sensibile all'applicazione. L'elemento sensibile, che rileva la pressione assoluta, è costituito da una membrana sospesa realizzata con un processo dedicato sviluppato dalla ST. Il LPS22HB è disponibile in un pacchetto LGA forato (HLGA). È garantito il funzionamento su un campo di temperatura che va da -40 a +85 °C. La confezione è forata per consentire alla pressione esterna di raggiungere l'elemento sensibile. Anche per quest'ultimo sensore descritto, le uscite dati sono accessibili attraverso l'interfaccia I²C/SPI. Nell'applicazione descritta, l'I²C è la periferica preferita, quindi il pin CS è legato al Vdd. L'indirizzo slave associato per questo apparecchio è impostato su 0xB8 (SA0 impostato su GND). LPS22HB è alimentato dal dominio di potenza SENS_VDD, collegato al dominio VDD_BUCK (uscita 3,3 V da ST1PS01) attraverso uno degli interruttori incorporati nell'STBC02 (STBC02_SW2_OB). Questo permette strategie di bassa alimentazione implementate nel firmware, tagliando l'alimentazione dei sensori quando non vengono utilizzati. Il bus I²C (SENS_I2C_SCK_SWITCH e SENS_I2C_SDA_SWITCH), condiviso con HTS221, è collegato a Pini del modulo LoRa PB8 e PB9.

4.2 GATEWAY



Figura 9: Sentrius™ RG1xx LoRa-Enabled Gateway di Laird[27]

Il Sentrius™ RG1xx LoRa-Enabled Gateway di Laird è il massimo in termini di sicurezza, scalabilità e robustezza per soluzioni LoRa per il controllo end-to-end della propria rete privata LoRaWAN. Sfruttando la rete del modulo Laird certificato "Wireless Bridge" della Serie 50, collaudato e affidabile, offre anche enterprise dual-band Wi-Fi, BT v4.0 (BLE e Classic) ed Ethernet cablata per una progettazione in completa libertà. Basato sul design dei chipset Semtech SX1301/SX1257, offre una gamma LoRa fino al 10 miglia e il software LoRa Packet Forwarder pre-caricato, perfetto per un IoT altamente scalabile ed una flessibile rete. Il gateway Sentrius™ RG1xx Gateway funziona con Sentrius™ RM1xx di moduli Laird certificati LoRa+BLE della serie LoRa+BLE per una semplice integrazione "out-of-the-box" ed è compatibile con Partner Cloud e LoRa di terze parti, così come qualsiasi dispositivo client certificato LoRaWAN.

5 PROGETTO

Per eseguire il progetto le 3 principali azioni di configurazione che si dovrebbero svolgere sono:

- Creazione e configurazione account TTN
- Configurazione firmware da caricare poi sulla steval
- Configurazione steval
- Configurazione gateway

In questo caso però, verrà utilizzato l'account con nome utente "mElabUnivPM" creato precedentemente da altri studenti i quali hanno già registrato sia la steval che il gateway. Riguardo la configurazione del firmware si andrà ad effettuare solo una piccola modifica di un parametro del firmware utilizzato dal precedente studente. In pratica, in ordine, si modifica il firmware, si carica sulla board e per il gateway si cambia solamente la sua posizione dalla piattaforma TTN.

5.1 FP-ATR-LORA1 firmware

La funzione principale del firmware applicativo FP-ATR-LORA1 è quella di raccogliere dati provenienti da sensori, geo-posizioni da GNSS e inviarli tramite la connettività LoRaWAN. Include i driver per la radio LoRa, il modulo Teseo-LIV3F GNSS, i sensori di movimento e ambientali e la gestione dell'alimentazione. Per garantire la piena funzionalità del sistema e una lunga autonomia della batteria, FP-ATR-LORA1 implementa il funzionamento completo e profili a bassa potenza. Nel firmware dell'applicazione è implementata una macchina a stati per gestire le transizioni tra i vari stati.

5.1.1 FP-ATR-LORA: macchina a stati

Il firmware dell'applicazione FP-ATR-LORA1 implementa una macchina a stati che può essere utilizzata per sviluppare, ad esempio, applicazioni di asset tracking, di gestione della flotta e di pet/child tracking. La macchina a stato è composta da tre stati principali: Run, Low Power e Ultra Low Power. Tuttavia, altri stati temporanei (Read, Send, Check Acknowledgement, Retrieve Data, Prepare Low Power, Prepare Ultra Low Power) possono essere impostati per eseguire azioni personalizzate, come la lettura di sensori o l'invio di dati attraverso la rete LoRaWAN. Al reset del sistema, il dispositivo rimane in funzione per

raccogliere i dati dai sensori e dal GNSS e monitorare i valori dei sensori per rilevare se alcune soglie vengono superate o se il GNSS segnala un allarme geofence. Quando l'algoritmo interno di rilevazione del movimento dell'accelerometro segnala che il dispositivo è a riposo, il dispositivo viene messo in modalità low power (inattività), disabilitando alcuni sensori e/o domini di potenza; il dispositivo può essere risvegliato dall'accelerometro stesso quando viene spostato (movimento). Se il dispositivo rimane in modalità a bassa potenza per un lungo periodo di tempo, un timer lo attiverà per entrare in modalità a bassissima potenza; in questo stato, il consumo di energia è ridotto al minimo. Dopo essere entrato nella modalità a bassissima potenza, il sistema si sveglia periodicamente solo per leggere i sensori e inviare i dati. È possibile implementare un controllo sui dati del sensore (accelerazione, dati ambientali, geofence, ecc.) per uscire dalla modalità a bassissima potenza ed entrare in modalità di funzionamento.

Run: è lo stato di default al reset del sistema. Tutti i dispositivi sono inizializzati e l'applicazione esegue il polling dei sensori ambientali, l'accelerometro, il sensore GNSS e il convertitore ADC della batteria per dati aggiornati. I sensori e il ricevitore GNSS sono continuamente monitorati, per rilevare se le soglie del sensore sono superate o il GNSS invia avvisi di geofence. Questi eventi fanno sì che il sistema vada in modalità Read And Send. Anche l'accelerometro è monitorato per controllare:

- inattività: nessun movimento viene rilevato per un certo periodo di tempo
- risveglio: viene rilevato un movimento (l'evento può essere attivato scuotendo brevemente la scheda).

Tre timer sono impostati su:

- passare allo stato di lettura ad un intervallo di tempo configurabile dall'utente
- passare a Invia stato
- passare alla modalità Ultra Low Power dopo un certo intervallo

Tutti i sottosistemi, sensori, GNSS e MEM_VDD sono alimentati e i dispositivi sono in funzione. L'MCU è in modalità run. STUSB1600_VDD si attiva a seconda del collegamento del cavo USB.

Low Power: l'MCU entra in modalità sleep per ridurre il consumo di energia e può essere svegliato con l'accelerometro interrompere gli eventi; i sensori ambientali sono disabilitati. Se

l'evento di risveglio dell'accelerometro o l'evento del timer sono rilevato, il sistema passa allo stato di lettura e, successivamente, allo stato di invio. Se l'evento di inattività dell'accelerometro è rilevato, il sistema passa allo stato di Ultra Low Power. Prima di entrare in questo stato, uno stato di preparazione a bassa potenza è eseguito per disabilitare i percorsi di alimentazione inutilizzati (nessun GNSS e nessun sensore ambientale) e impostare un timer di risveglio.

Ultra Low Power: l'MCU entra in modalità Stop per ridurre al minimo il consumo di energia. I sensori ambientali e l'accelerometro sono disabilitati e l'MCU può essere svegliata solo dall'interruzione del timer evento. Quando questo evento viene attivato, il sistema passa allo stato Read e, successivamente, allo stato Send. Prima di entrare nello stato di Ultra Low Power, uno stato di Preparare Ultra Low Power è impostato per disabilitare i percorsi di alimentazione non utilizzati: i sensori di sottosistema, SENS_VDD, il sottosistema GNSS, LIV3_VDD e MEM_VDD, non sono alimentati. L'accelerometro viene messo in modalità di spegnimento tramite comando software. L'MCU è in modalità di arresto. STUSB1600_VDD è attivato a seconda del collegamento del cavo USB.

Read: l'applicazione raccoglie dati ambientali, accelerometro e GNSS. Dopo lo stato Read, i dati sono pronti per essere salvati nella EEPROM interna e/o inviati attraverso la rete LoRaWAN. I sensori vengono letti e valutata anche in modalità di esecuzione. In ogni caso, questo "stato temporaneo" dedicato è implementato per avere un pacchetto coerente dei dati acquisiti dai sensori ad una specifica data e ora.

Send: l'applicazione invia i dati alla rete LoRaWAN confezionati in un unico messaggio. Dopo che il messaggio è inviato, l'applicazione esegue la verifica della conferma di ricezione e poi la macchina a stati ritorna allo stato precedente.

5.1.2 Firmware utilizzato

Il function pack appena descritto è messo a disposizione da ST Microelectronics per gli end-node delle reti LoRa. Il software viene caricato sull' IDE apposito reso disponibile da ST (ST Cube IDE) dove può essere modificato e successivamente buildato e caricato nella STEVAL attraverso un debugger. Il file che si andrà a modificare per avere le caratteristiche necessarie al nostro progetto sarà il main.c all'interno della cartella "STM32CubeFunctionPack_LORA1_V2.2.0". In particolare. l'obiettivo è di campionare 18 volte al secondo per 18 secondi, inviare le misure effettuate e, successivamente, inviare le misure di GPS, temperatura, umidità, pressione e livello di carica batteria, dopodichè ripetere

il ciclo. È prevista una modalità di risparmio energetico che entra in funzione quando viene rilevata inattività da parte dell'accelerometro (circa 20 misure consecutive di inattività); in questa modalità la board invia pacchetti di GPS, temperatura, pressione, umidità e percentuale di carica della batteria ogni due minuti. Quando l'accelerometro rileva un movimento la STEVAL esce dalla modalità di risparmio energetico e riprende il suo normale ciclo di funzionamento. In seguito verranno mostrate le modifiche più influenti effettuate sul codice:

- Lo **stato ReadMems** è rimasto di default. Se `txflag == 1` (ovvero se durante la Low Power mode il tx timer attiva il flag) oppure se `counter == 0` (contatore utilizzato per il normale ciclo di funzionamento) allora si vanno a leggere i sensori ambientali e il GPS (che viene letto nello stato `ReadGNSS`, impostato come il successivo); altrimenti si effettua una lettura dell'accelerometro e si passa allo stato `SaveData` per il salvataggio della misura nella EEPROM.

```

1166 }
1167 case State_ReadMems:
1168 {
1169     if( ( PlatformStatus.b.SHORTEN_JOIN_REQ_INTV==0 ) || ( CHECK_LORA_JOIN () || ( PlatformStatus.s.nErrorNoAck != 0 ) ) )
1170     {
1171 #if defined(SENSOR_ENABLED)
1172     HAL_Delay(SENSOR_POLL_INTERVAL);
1173     PRINTF_MAIN1("Enable and read sensors\r\n");
1174     PlatformStatus.s.xCurrentState = State_ReadMems;
1175     LED_On( LED_WHITE );
1176     BSP_sensor_Enable();
1177     BSP_accelero_Enable();
1178     if (txflag == 1 || counter == 0)
1179     {
1180         BSP_sensor_Read( &sensor_data );
1181         PRINTF_MAIN1("Got sensor data: T: %.2f H: %.2f P: %.2f\r\n", sensor_data.temperature, sensor_data.humidity, sensor_data.pressure);
1182         PlatformStatus.s.xNextState = State_ReadGnss;
1183         package_type = 0;
1184     }
1185     else
1186     {
1187
1188
1189         BSP_sensor_Read( &sensor_data );
1190         PRINTF_MAIN1("Accelerometer: X: %d Y: %d Z: %d\r\n", sensor_data.acceleration_x, sensor_data.acceleration_y, sensor_data.acceleration_z);
1191         PlatformStatus.s.xNextState = State_SaveData;
1192         package_type = 1;
1193

```

Figura 10: stato `ReadMems`

- **Stato SaveData**: a seconda del tipo di pacchetto che viene salvato (GPS/sensori ambientali o dati accelerometrici); è fondamentale che in ogni caso i due tipi di pacchetti abbiano la stessa lunghezza (per questo alcune variabili vengono settate a 0) perchè durante la fase di

lettura il puntatore alle celle di memoria sia perfettamente allineato. Alla fine dello stato SaveData viene effettuato il controllo del contatore il quale se è inferiore a 18 la macchina ritorna allo stato Read, altrimenti va allo stato Send per la trasmissione dei dati.

```

if (PlatformStatus.s.lastReadIndex > PlatformStatus.s.lastSavedIndex)
{
    PlatformStatus.s.lastReadIndex = PlatformStatus.s.lastSavedIndex    ;
}
if( ( PlatformStatus.b.GNSS_FIX_CHECK && gnss_fix_validity != INVALID ) || PlatformStatus.b.GNSS_FIX_CHECK == 0)
{
    if( ( PlatformStatus.b.SHORTEN_JOIN_REQ_INTV==0 ) || ( CHECK_LORA_JOIN () || ( PlatformStatus.s.nErrorNoAck != 0 ) ) )
    {
        PlatformStatus.s.xCurrentState = State_SaveData;
        if(package_type == 0)
        {
            data_logged.temperature = ( int16_t )( sensor_data.temperature * 10 );      /* in °C * 10 */
            data_logged.pressure     = ( uint16_t )( sensor_data.pressure * 10 );       /* in hPa / 10 */
            data_logged.humidity     = ( uint16_t )( sensor_data.humidity * 2 );       /* in %*2 */

            data_logged.latitude     = ( int32_t )( sensor_data.latitude * 10000 );
            data_logged.longitude    = ( int32_t )( sensor_data.longitude * 10000 );
            data_logged.altitudeGps  = ( int32_t )( sensor_data.altitudeGps * 100 );

            data_logged.batteryLevel = ( uint32_t ) (BSP_GetBatteryLevel16()/10);
            data_logged.epoch_value  = App_GetTimeStamp();
            data_logged.sentFlag     = 0;
            data_logged.AccEnable    = accActInact + 2*(gnss_fix_validity != INVALID);
            data_logged.accelero_x   = 0;
            data_logged.accelero_y   = 0;
            data_logged.accelero_z   = 0;
            PlatformStatus.s.xNextState = State_SendData;
            break;
        }

    }
    else
    {
        data_logged.pressure     = 0;          /* in hPa / 10 */
        data_logged.humidity     = 0;          /* in %*2 */
        data_logged.latitude     = 0;
        data_logged.longitude    = 0;
        data_logged.altitudeGps  = 0;
        data_logged.batteryLevel = 0;
        data_logged.epoch_value  = App_GetTimeStamp();
        data_logged.sentFlag     = 0;
        data_logged.AccEnable    = accActInact + 2*(gnss_fix_validity != INVALID);
        data_logged.temperature  = 0;
        data_logged.accelero_x   = ( int32_t )( sensor_data.acceleration_x * 10);
        data_logged.accelero_y   = ( int32_t )( sensor_data.acceleration_y * 10);
        data_logged.accelero_z   = ( int32_t )( sensor_data.acceleration_z * 10);
    }
}

```

Figura 11: Stato SaveData

```

if(counter< COUNTER_VALUE)
{
    PlatformStatus.s.xNextState = State_Read;
    break;
}
else
{
    PlatformStatus.s.xNextState = State_Send;
    break;
}
}

```

Figura 12: Condizione contatore

• **State Send:** qui si ha la trasmissione dei dati attraverso la funzione SendData(): se la trasmissione va a buon fine questa restituisce 0, altrimenti 1; ; in ogni caso dopo ogni trasmissione la STEVAL entra in stato di idle per qualche secondo, e successivamente allo stato CheckAck.

```
case State_SendData:
{
    PlatformStatus.s.xCurrentState = State_SendData;
    LED_On( LED_WHITE );
    PRINTF_MAIN1("Send data...\r\n");
    lastSendTick = HAL_GetTick();
    bLoraAck = FALSE;
    if((RetVal = SendData( )))
    {
        //send error
        PlatformStatus.s.xNextState = State_IdleAfterSend;
    }
    else
    {
        if(PlatformStatus.s.lastSavedIndex > PlatformStatus.s.lastReadIndex)
        {
            PRINTF_MAIN1("I'M SENDING ALL DATA.....\r\n");
            PlatformStatus.s.xNextState = State_IdleAfterSend;
        }
        PlatformStatus.s.xCurrentState = State_IdleAfterSend;
    }
}
break;
```

Figura 13: Stato Send

• **Stato CheckAck:** in questo stato la macchina ha il compito di decidere se un pacchetto sarà ritrasmesso, nel caso in cui ci sono stati errori in trasmissione, e viene fatto tramite il valore di ritorno della funzione SendData RetVal, nel caso in cui si ha necessità di tornare allo stato Read o allo stato di Low Power; queste decisioni vengono prese sulla base dello stato di txflag, package_type, e lastReadIndex. Conseguentemente, a seconda dei casi, vengono anche resettate le relative variabili, contatori, flags ecc. Infine si effettua la verifica sulla ricezione dell'LoRaAck da parte del gateway.

```

case State_CheckAck:
{
    if (package_type == 1)
    {
        if(PlatformStatus.s.lastSavedIndex == PlatformStatus.s.lastReadIndex && RetVal != TRUE)
        {
            counter = 0;
            PlatformStatus.s.xNextState = State_Read;
        }
        else if (PlatformStatus.s.lastSavedIndex != PlatformStatus.s.lastReadIndex)
        {
            if(RetVal != TRUE)
            {

                PlatformStatus.s.xNextState = State_Send;
                nPacketsSent = nPacketsSent + k;
            }
            else
            {
                PlatformStatus.s.lastReadIndex = PlatformStatus.s.lastReadIndex - k ; // !!!
                PlatformStatus.s.xNextState = State_Send;
            }
        }
    }
    else
    {
        if (txflag == 0)
        {
            if(RetVal != TRUE)
            {
                PlatformStatus.s.xNextState = State_Read;
                nPacketsSent++;
                counter++;
                package_type = 1;
            }
            else
            {
                PlatformStatus.s.xNextState = State_PrepareLowPower;
                nPacketsSent++;
                package_type = 1;
                txflag = 0;
                TimerStop(&TxTimer);
                TimerSetValue(&TxTimer, TX_TIMER_INTERVAL_DEF); //PlatformStatus.s.TX_TIMER_INTERVAL
                TimerStart(&TxTimer);
            }
            else
            {
                PlatformStatus.s.xNextState = State_SendData;
            }
        }
    }
}

```

```

if(PlatformStatus.b.LOG_MANAGER)
{
  if( ( bLoraAck == 1 ) || ( GET_CONFIRM_MSG_STATE() == LORAWAN_UNCONFIRMED_MSG && bwaitAck != 0 ) )
  {
    /* LoRa Ack received, set flag in the saved item */
    PRINTF_MAIN1("LoRa ack received\r\n");
    PushLogFlag(PlatformStatus.s.lastReadIndex);
    PlatformStatus.s.nErrorNoAck=0;
  }
  if(PlatformStatus.s.lastSavedIndex > PlatformStatus.s.lastReadIndex)
  {
    //PRINTF_MAIN1("Set tx timer to flush data\r\n");
    // TimerStop(&TxTimer);
    //TimerSetValue(&TxTimer, PlatformStatus.s.TX_TIMER_INTERVAL);
    //TimerStart(&TxTimer);
  }
}
}
else
{
  PRINTF_MAIN1("No LoRa ack, item will be resent\r\n");
  //PRINTF_MAIN2("Set tx timer to flush data\r\n");
  //TimerStop(&TxTimer);
  //TimerSetValue(&TxTimer, PlatformStatus.s.TX_TIMER_INTERVAL);
  //TimerStart(&TxTimer);
  if(PlatformStatus.s.nErrorNoAck < MAX_ERROR_NO_ACK)
  {
    PlatformStatus.s.nErrorNoAck++;
  }
  else
  {
    {
      if(CHECK_LORA_JOIN())
      {
        PlatformStatus.b.APP_LORA_JOINED = FALSE;
        PRINTF_MAIN2("Network is down\r\n");
      }
    }
  }
}

if(nTxWithoutLowPower)nTxWithoutLowPower--;
}

ReloadSleepTimer();
break;
}

```

Figura 14: Stato CheckAck

5.2 CONFIGURAZIONE STEVAL

Come già precedentemente accennato per la configurazione della steval l'unico compito è quello di caricare su esso il firmware descritto sopra, cioè effettuare un debug:

Passo 1. Collegare il cavo piatto a 5 pin (lato maschio) al connettore STEVAL-STRKT01 CN501 come da Tabella 4

Passo 2. Collegare il cavo piatto a 5 pin (lato femmina) ai pin 1-5 di ST-LINK/V2 come da Tabella 4

Passo 3.

Passo 3a. Collegare la batteria e premere il pulsante SW400 per accendere la scheda (per almeno 1.250 s)

oppure

Passo 3b. Alimentare lo STEVAL-STRKT01 tramite un cavo USB di tipo C (e un adattatore da tipo C a tipo A se necessario)

Passo 4. Collegare il ST-LINK/V2 ad un PC tramite un cavo di tipo A/mini B

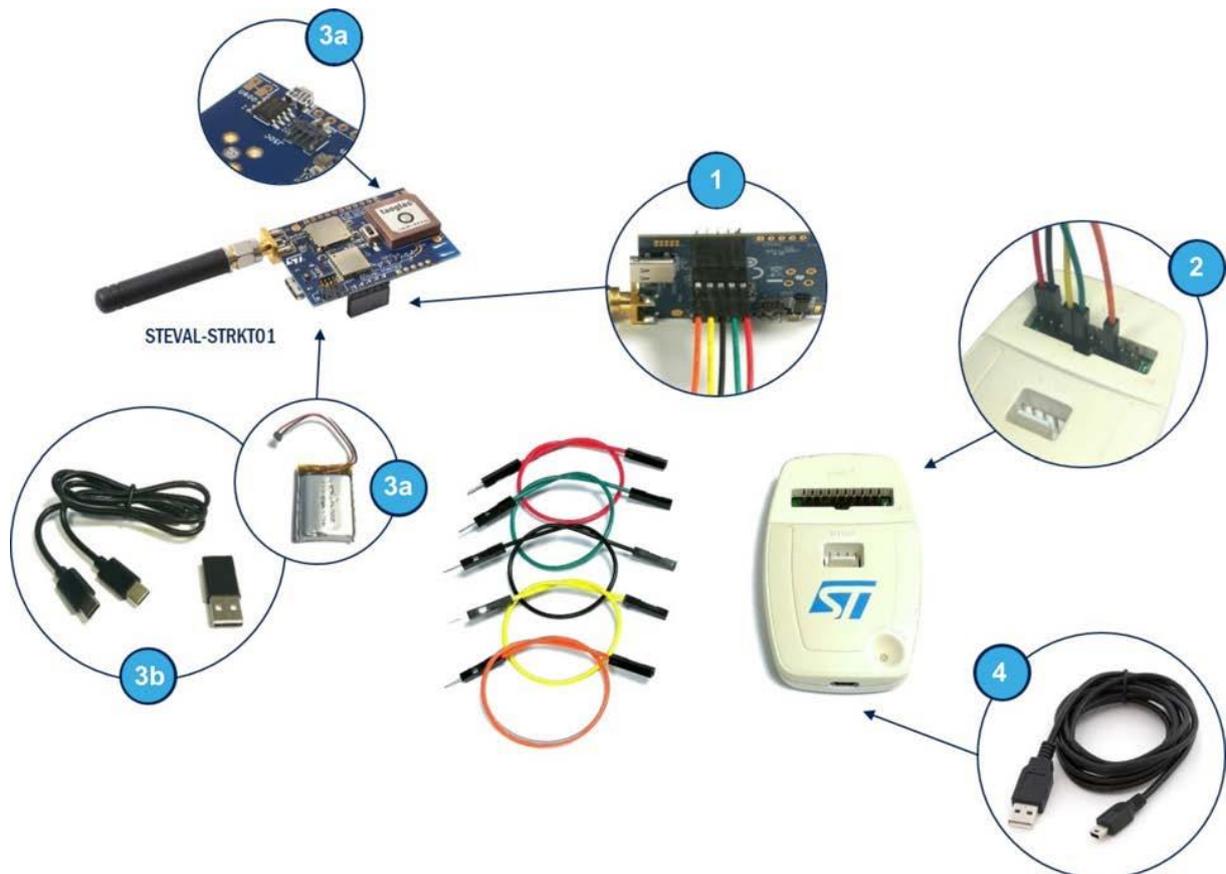


Figura 15: Configurazione hardware per l'aggiornamento del firmware STEVAL-STRKT01 utilizzando un ST-LINK/V2 debugger [29]

Tabella 4 [29]

ST-LINK/V2 connector (pin and label)	ST-LINK/V2 connector (pin and label)	STEVAl-STRKT01 CN501 (pin and label)
	2, VAPP	5, D_VDD
	4, GND	3, GND
	7, TMS_SWDIO	2, SWD_SWDIO
	9, TCK_SWCLK	4, SWD_SWCLK
	15, NRST	1, nRESET

Dopo la configurazione hardware si va sull'IDE e si esegue prima l'istruzione "build" e successivamente l'istruzione "debug" fino a suo completamento; A questo punto la board è pronta per l'utilizzo.

5.3 CONFIGURAZIONE GATEWAY

Per la configurazione del gateway, come detto in precedenza, si dovrà solamente cambiare la sua posizione da TTN, perciò si va su TTN si accede al profilo dove è registrato il nostro gateway, si va alla sezione gateway si clicca sul nostro dispositivo e si clicca su "location". A questo punto basta cliccare la posizione desiderata sulla mappa visualizzata e fare "save location".

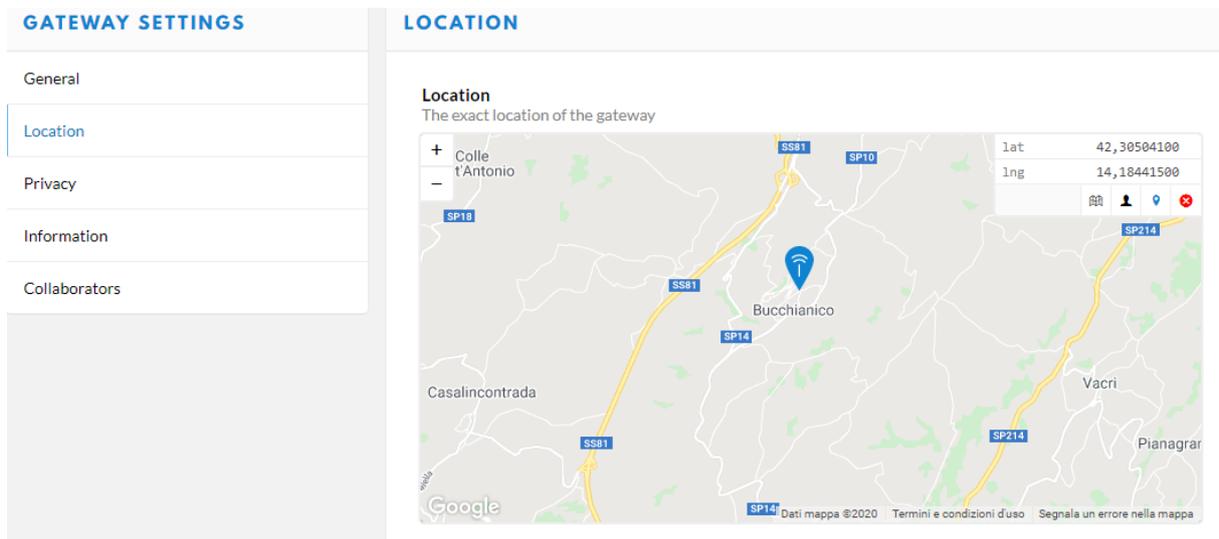


Figura 16: Sezione location sulla piattaforma TTN

5.4 ANALISI MISURE

Per le misure effettuate andremo a suddividere quest'ultime in 3 modalità di misura diverse:

-misure indoor

-misure outdoor con ostacoli

-misure outdoor con meno ostacoli possibili

La steval è stata settata in modo da mandare 2 pacchetti di dati ogni 18 secondi circa suddivisi come:

- 1) Pacchetto con 18 misure consecutive dell'accelerometro con frequenza di campionamento di 10Hz
- 2) Pacchetto con dati: GPS, temperatura, umidità, livello di carica.

Di conseguenza ogni 18 secondi verranno trasmessi alternatamente questi 2 pacchetti di dati.

Quest'ultimi saranno visualizzati sull'integration "data storage" disponibile su TTN.

Misure indoor

Riguardo queste misure esse sono state effettuate all'interno di una abitazione dove era presente il gateway. In questo caso si è limitati semplicemente a spostarsi all'interno dell'abitazione. Di seguito la mappa dell'appartamento in questione dove il tracciato blu indica le linee di spostamento della board mentre acquisiva i pacchetti di dati;

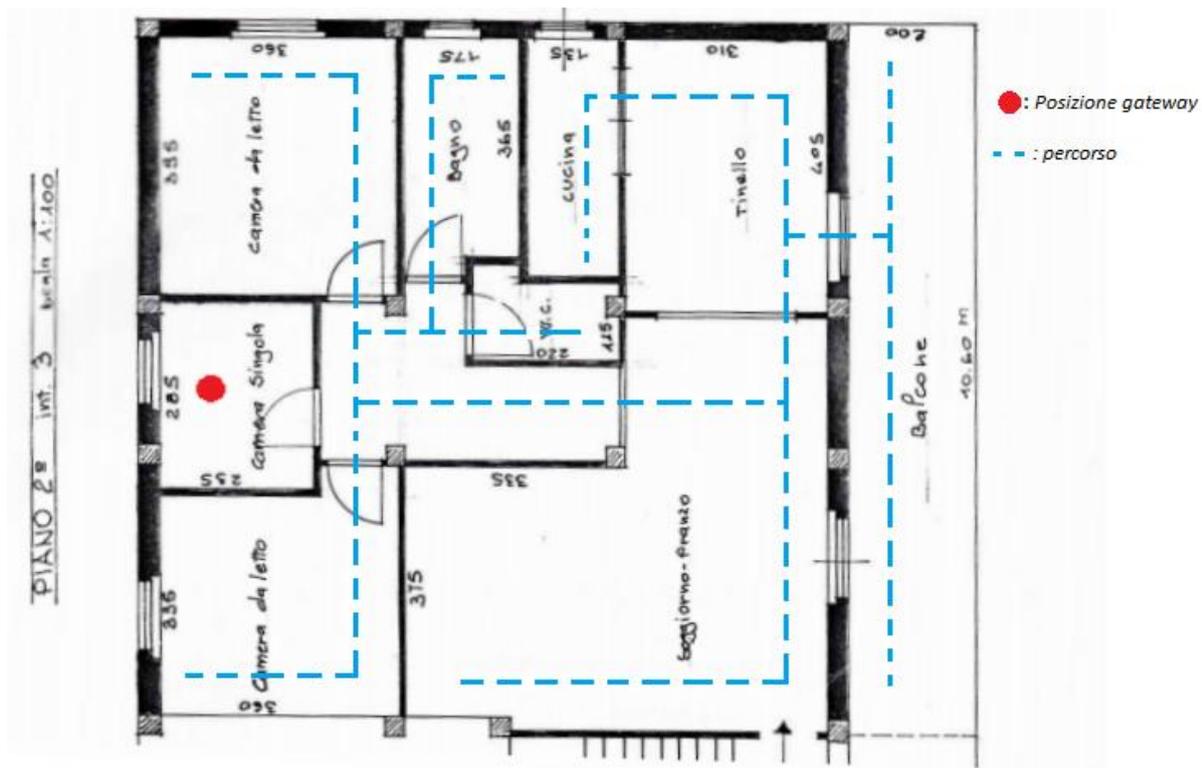


Figura 17: mappa appartamento tracciata

Qui sotto ci sono i parametri relativi al segnale durante il percorso racchiusi nella tabella dei valori osservabile;

Numero	SNR [dBm]	RSSI [dBm]	Temperatura [C°]
1	11.80	-58.00	20.7
2	11.20	-57.00	20.8
3	11.80	-65.00	21
4	12.50	-63.00	21.2
5	9.00	-70.00	21.4
6	12.00	-54.00	21.3
7	11.00	-49.00	21.4
8	12.00	-66.00	21.5
9	8.00	-51.00	21.7
10	11.50	-78.00	22
11	12.50	-60.00	22
12	11.50	-51.00	22

13	12.00	-69.00	21.8
14	11.80	-73.00	22.1
15	11.50	-69.00	22.1
16	11.00	-69.00	22.2
17	8.80	-67.00	22.4
18	13.00	-69.00	22.4
19	9.50	-66.00	22.3
20	11.20	-65.00	22.2

Per prelevare questi valori si è ricorso all'utilizzo delle integrations messe a disposizione dalla piattaforma TTN in cui, una volta messe tutte le autorizzazioni con le relative chiavi di accesso, è possibile andare a visualizzare i pacchetti di dati registrati durante le misure.

Qui di sotto sono riportate alcune immagini dell'interfaccia dell'integration "Data storage";

The Things Network Data Storage

Stores data and makes it available using a REST API

Created by The Things Industries B.V.
See more at <https://www.thethingsindustries.com>
[Contact the developer](#)

devices

Show/Hide | List Operations | Expand Operations

GET /api/v2/devices Query the devices for which data has been stored

Implementation Notes
The devices endpoint returns the list of devices for which data has been stored

Response Class (Status 200)
An array of device IDs

Model | Example Value

```
[
  "string"
]
```

Response Content Type:

Parameters

Parameter	Value	Description	Parameter Type	Data Type
last	<input type="text" value="3d"/>	Duration on which we want to get the data (default 1h). Pass 30s for the last 30 seconds, 1h for the last hour, 2d for the last 48 hours, etc	query	string

[Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: key ttn-account-v2.2vHw4IvZhZ7zL3yqe8nYUZJdteTIIIZ1CkCOFS'
```

Request URL

```
https://lora_network_application.data.thethingsnetwork.org/api/v2/query?last=3d
```

Response Body

```
{
  "accelerometer_0": null,
  "accelerometer_1": null,
  "accelerometer_10": null,
  "accelerometer_11": null,
  "accelerometer_12": null,
  "accelerometer_13": null,
  "accelerometer_14": null,
}
```

```
Request URL
https://loranetworkapplication.data.thethingsnetwork.org/api/v2/query?last=3d

Response Body
{"device_id": "steval_1ot_tracker_3",
 "gps_3": "map[longitude:13.6288 altitude:273.26 latitude:43.2237]",
 "raw": "AH!mGAFnAN4CaGHdIAaYbQIUYABqvgQCGiw=",
 "relative_humidity_2": 49.5,
 "temperature_1": 22.2,
 "time": "2020-10-16T15:52:29.343591112Z"}
{
  "accelerometer_0": "map[x:-0.17 y:-0.04 z:-9.89]",
  "accelerometer_1": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_10": "map[x:-0.16 y:-0.03 z:-9.89]",
  "accelerometer_11": "map[x:-0.16 y:-0.04 z:-9.9]",
  "accelerometer_12": "map[x:-0.17 y:-0.04 z:-9.89]",
  "accelerometer_13": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_14": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_15": "map[x:-0.16 y:-0.03 z:-9.89]",
  "accelerometer_16": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_17": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_2": "map[x:-0.16 y:-0.04 z:-9.88]",
  "accelerometer_3": "map[x:-0.15 y:-0.03 z:-9.89]",
  "accelerometer_4": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_5": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_6": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_7": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_8": "map[x:-0.16 y:-0.04 z:-9.89]",
  "accelerometer_9": "map[x:-0.16 y:-0.04 z:-9.89]"
}
```

Figura 18: Interfaccia “Data Storage”

Da notare la presenza dei valori di accelerazione i quali sono 18 per ogni pacchetto di misura dove si alternano con i pacchetti contenenti dati GPS, umidità, temperatura, data ed ecc..

Misure outdoor (con ostacoli)

In questa modalità di misure si indossa la board e si va a passeggio su strade che, a livello d’aria, tra end-device e gateway si trovano 1-5 palazzine come ostacolo. A differenza di prima qui si avrà un mapping delle posizioni così da mettere in relazione la distanza e i parametri del segnale. Di seguito la mappa delle posizioni prese dall’integration “TTN mapper” fornita sempre dalla piattaforma TTN;

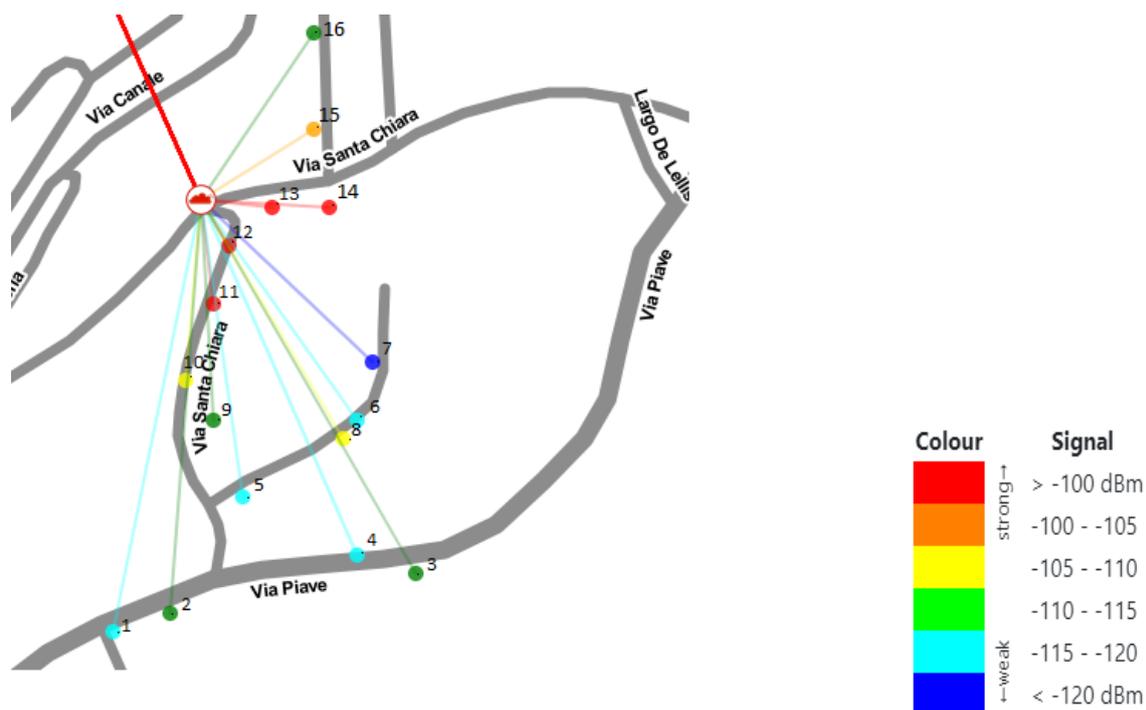


Figura 19: mapping misure outdoor con ostacoli

Di seguito la tabella dei valori dei parametri di ogni punto;

Numero	Distanza [m]	Altezza [m]	Potenza [dBm]	SNR [dBm]	RSSI [dBm]
1	254	301.8	-117	-2.00	-115.00
2	239	294.2	-115	1.00	-115.00
3	248	285.5	-115	-6.00	-109.00
4	223	285.7	-118.5	-4.50	-114.00
5	173	295.9	-119	-4.00	-115.00
6	155	298.0	-116.8	-1.80	-115.00
7	135	296.1	-123	-10.00	-113.00
8	160	295.0	-107	0.80	-107.00
9	127	297.1	-113	1.80	-113.00
10	105	300.6	-107	7.00	-107.00
11	61	306.9	-97	10.50	-97.00
12	31	309.1	-87	9.20	-87.00
13	40	309.2	-87	7.80	-87.00
14	73	301.6	-96	11.80	-96.00
15	76	302.2	-101	8.50	-101.00
16	115	304.4	-114	1.00	-114.00

Da come si può notare nella mappa, osservando la legenda di colori, i punti con più attenuazioni sono quelli segnati in blu e in celeste; questo perché in quei punti lo spazio d'aria tra board e gateway è ostacolato da 4-5 palazzi, infatti man mano che ci si riavvicina verso il gateway evitando i palazzi aumenta la qualità della ricezione. In generale nei punti segnati con un colore al di sotto dell'arancio si va ad indicare la presenza di ostacoli nello spazio d'aria. Nel mapping e nella tabella risultano 16 misure ma in realtà ne sono state effettuate molte di più, ma non risultano nel mapping perché la steval non riusciva ad inviare dati a causa della presenza di troppi ostacoli.

Misure outdoor (con meno ostacoli possibili)

In questa modalità le misure vengono effettuate in zone in cui non ci sono edifici di ostacolo tra end-device e gateway o al massimo solo la presenza di alberi con dimensioni contenute. Anche in questo caso si ha il mapping delle posizioni dove, a differenza di prima, si noterà un notevole allontanamento dal gateway di alcuni punti in quanto, a causa delle caratteristiche del territorio collinare, si è dovuti passare da una collina ad un'altra per non avere ostacoli tra gateway e end-device. Di seguito il mapping delle misure effettuate utilizzando, come sopra, TTN mapper;



Figura 20: mapping generale delle misure outdoor senza ostacoli

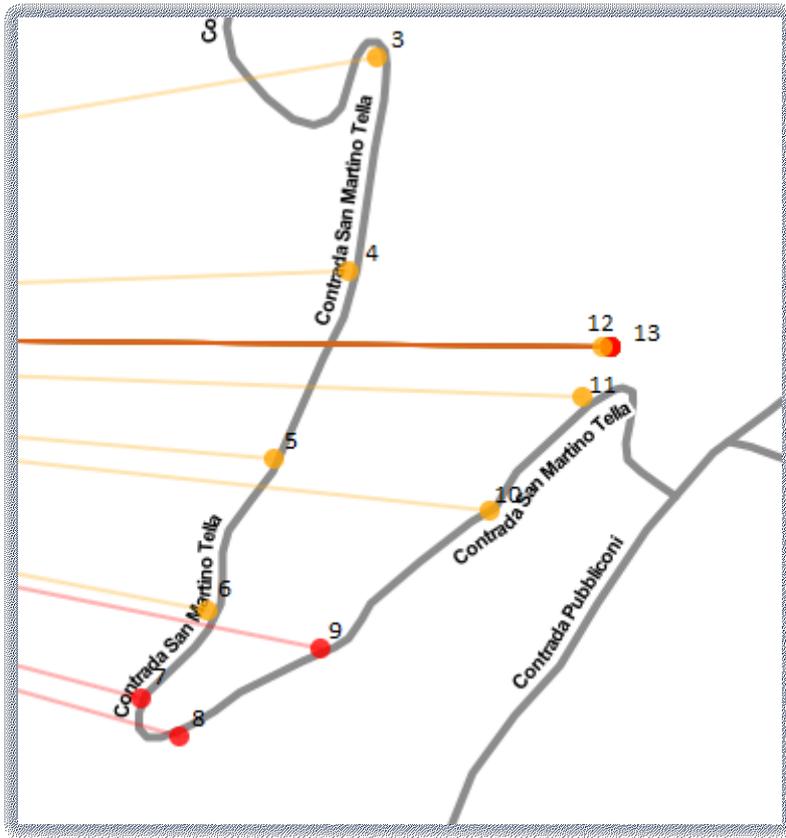


Figura 21: zoom 1 sul mapping generale

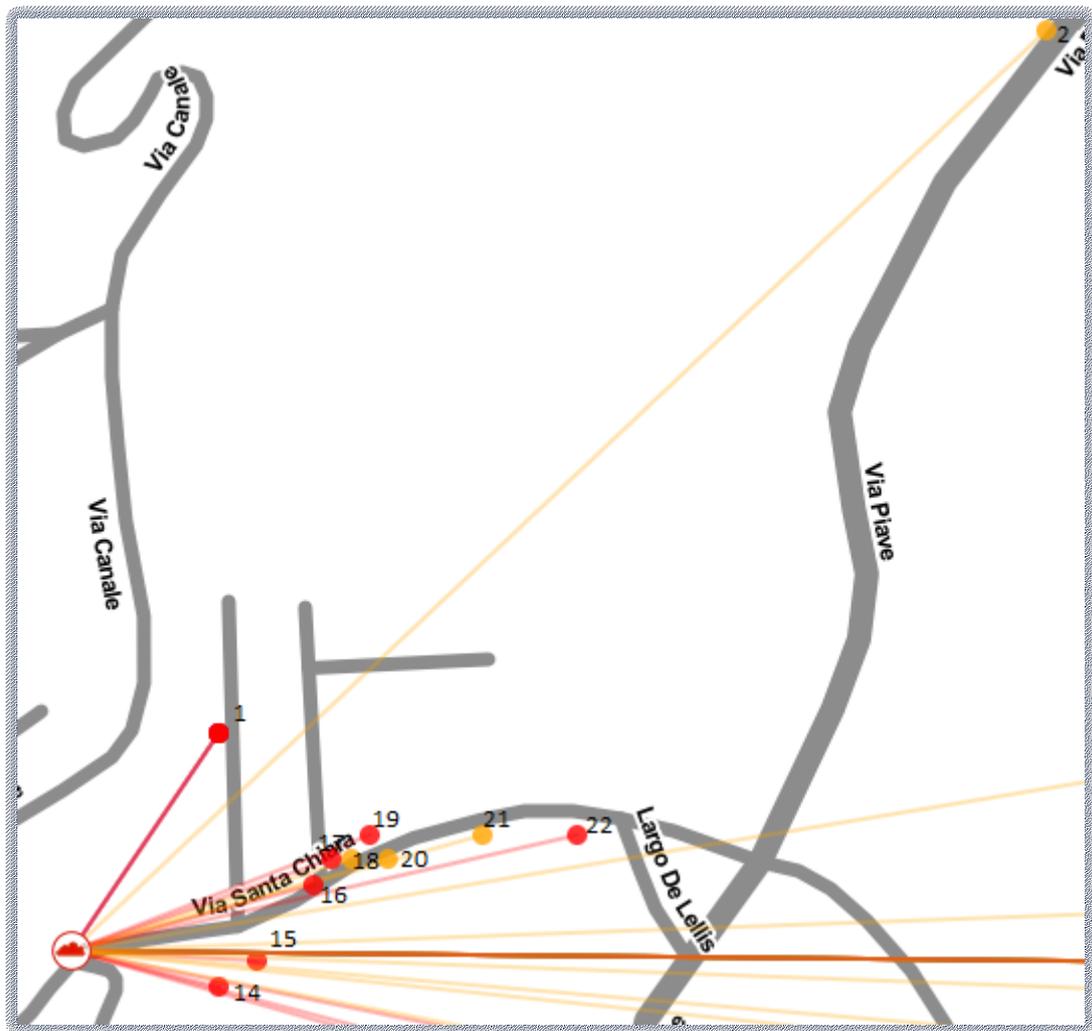


Figura 22: zoom 2 del mapping generale

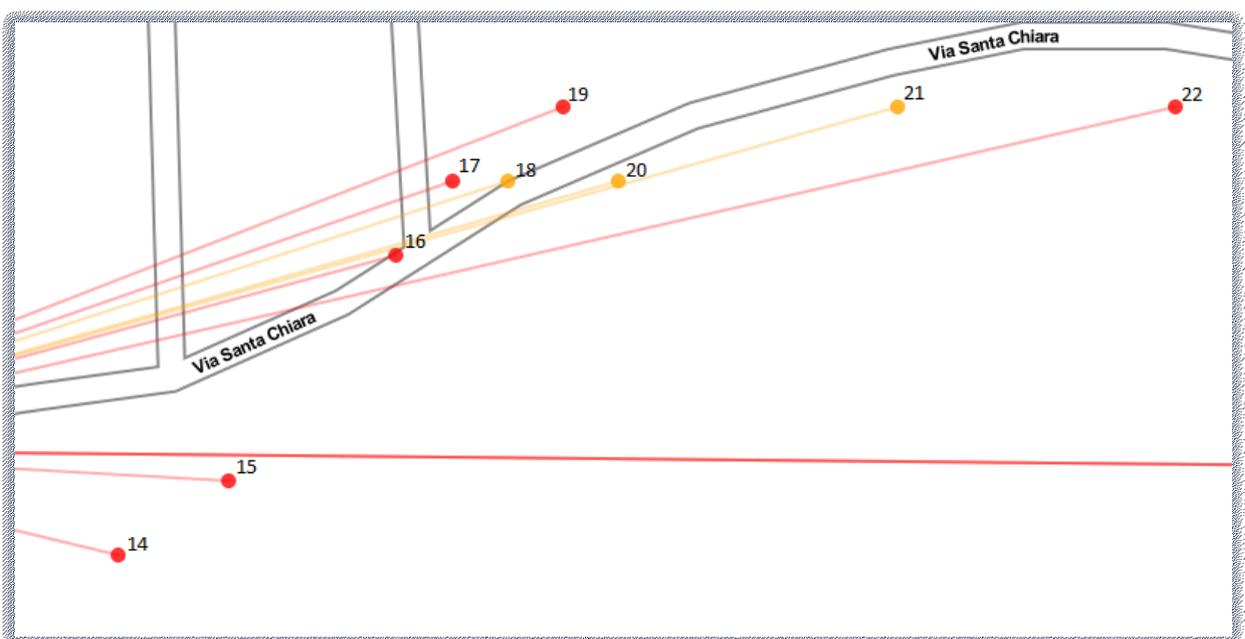


Figura 23: zoom 3 del mapping generale

Qui di seguito la tabella dei valori dei parametri di ogni punto;

Numero	Distanza [m]	Altezza [m]	Potenza [dBm]	SNR [dBm]	RSSI [dBm]
1	115	348.1	-32	11.50	-32.00
2	589	264.8	-104	0.50	-104.00
3	1450	169.4	-105	2.00	-105.00
4	1406	187.9	-102	5.50	-102.00
5	1344	200.9	-102	5.00	-102.00
6	1306	209.0	-101	7.50	-101.00
7	1267	214.7	-96	9.00	-96.00
8	1308	221.9	-97	10.80	-97.00
9	1409	284.3	-97	4.00	-97.00
10	1537	324.2	-105	10.50	-105.00
11	1612	314.6	-103	10.20	-103.00
12	1627	291.2	-102	8.20	-102.00
13	1635	287.8	-99	11.50	-99.00
14	66	299.8	-91	9.00	-91.00
15	81	301.0	-84	11.50	-84.00
16	110	298.3	-85	11.20	-85.00
17	121	295.7	-93	11.50	-93.00
18	128	294.8	-103	8.20	-103.00
19	140	290.5	-87	12.20	-87.00
20	144	308.6	-101	11.50	-101.00
21	187	313.3	-105	3.00	-105.00
22	227	303.1	-97	10.20	-97.00

Come si può notare dalle mappe e dai dati la qualità del segnale rimane buona anche allontanandosi di molto dal gateway a patto che non ci siano ostacoli infatti nel punto 13, cioè il punto più lontano rilevato, sia ha il tracciato di colore rosso perciò il segnale è forte. Anche in questo caso il numero di misure è ridotto rispetto a quelle effettuate in conseguenza del fatto che a causa della conformazione territoriale è stato molto frequente trovarsi in zone in

cui il segnale non riusciva ad arrivare al gateway. In generale con distanze abbastanza elevate il segnale si attenua molto poco esattamente come mostrato dalla figura 21 dove il colore del tracciato non va più basso dell'arancione sempre però che si abbia uno spazio libero privo di ostacoli, o per lo meno ostacoli molto piccoli.

5 CONCLUSIONI

In conclusione si può asserire, osservando le varie modalità di misura effettuate, che il dispositivo steval e il protocollo LoRaWAN risultano essere molto efficaci in ambienti in cui è favorito uno spazio d'aria privo di grandi ostacoli come aree verdi e zone simili. Non è da sottovalutare comunque l'efficacia di entrambi anche in ambienti urbani perché, come mostrato sopra, anche con la presenza di grandi ostacoli i dati riescono ad essere trasmessi con modesta efficienza. Molto importante è anche la semplicità nel cambio dei comandi di rilevazione della scheda in quanto, è necessario solamente andare ad apportare modifiche del function pack citato in capitoli precedenti, cioè il firmware che viene caricato sulla steval il quale risulta essere molto funzionale e di facile utilizzo.

RIFERIMENTI BIBLIOGRAFICI

- [1] Wu, X., Xiong, Y., Li, M., Huang, W., Distributed spatial-temporal compressive data gathering for large-scale WSNs, Computing, Communications and IT Applications Conference (ComComAp), pp. 105-110, 2013.
- [2] INFOS D.4 Networked Enterprise RFID INFOS G.2 Micro Nanosystems in Co-operation with the Working Group RFID of the ETP EPOSS, 2008.
- [3] Semtech, AN1200.22 LoRa™ Modulation Basics, Application Note, <http://www.semtech.com/images/datasheet/an1200.22.pdf>
- [4] LoRa Technology, <https://www.lora-alliance.org/What-IsLoRa/Technology>
- [5] ABI Research, “Best Fit Use Cases for LPWANs,” pp. 1–16, August 2016.
- [6] The Evolution of the Internet of Things, On-line <http://www.ti.com/lit/ml/swrb028/swrb028.pdf>.
- [7] The Things Network Documentation, <https://www.thethingsnetwork.org/docs/>
- [8] D. Bankov, E. Khorov, and A. Lyakhov, “On the Limits of LoRaWAN Channel Access,” in 2016 International Conference on Engineering and Telecommunication, Moscow, Russia, November 2016, pp. 10–14.

- [9] LoRa Alliance Technical Marketing Workgroup, "A technical overview of LoRa and LoRaWAN," November, 2015.
- [10] A. Wixted, P. Kinnaird, A. Tait, A. Ahmadiania, and N. Strachan, "Evaluation of LoRa and LoRaWAN for Wireless Sensor Networks," in 2016 IEEE SENSORS, October 2016, pp. 1–3.
- [11] M. Bor, J. Vidler, and U. Roedig, "LoRa for the Internet of Things," in Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks, Graz, Austria, February 2016, pp. 361–366.
- [13] B. Reynders and S. Pollin, "Chirp Spread Spectrum as a Modulation Technique for Long Range Communication," in 2016 Symposium on Communications and Vehicular Technologies (SCVT), Mons, Belgium, November 2016, pp. 1–5.
- [14] Semtech, "LoRa Modulation Basics," pp. 1–26, May 2015.
- [15] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa LowPower Wide-Area Networks Scale?" in Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Malta, Malta, November 2016, pp. 59–67.
- [16] K. Mikhaylov, J. Petajärvi, and T. Hämäläinen, "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology," European Wireless 2016, pp. 119–124, 2016.
- [17] Semtech, "SX1272/3/6/7/8: LoRa Modem Designer's Guide," pp. 1–9, July 2013, AN1200.13.
- [18] P. Neumann, J. Montavont, and T. Noel, "Indoor Deployment of Low-Power Wide Area Networks (LPWAN): a LoRaWAN case study," in 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), New York, NY, USA, October 2016, pp. 1–8.
- [19] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRaWAN TM Specification," pp. 1–70, July 2016, V1.0.2.
- [20] M. Aref and A. Sikora, "Free space range measurements with Semtech LoRaTM technology," in 2014 2nd International Symposium on Wireless Systems within the

Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS 2014, Offenburg, Germany, September 2014, pp. 19–23.

[25] A. Augustin, J. Yi, T. Clausen, and W. Townsley, “A Study of LoRa: Long Range & Low Power Networks for the Internet of Things,” *Sensors*, vol. 16, no. 9, 2016.

[26] LoRa Alliance Technical committee, “LoRaWAN Regional Parameters,” July 2016, V1.0.

[27] <https://www.lairdconnect.com/documentation/quick-start-guide-sentrius-rglxx-v30>

[28] https://www.st.com/resource/en/application_note/dm00658170-stevalstrkt01-power-management-architecture-description-and-configuration-for-optimized-battery-life-stmicroelectronics.pdf

[29] https://www.st.com/resource/en/user_manual/dm00595064-getting-started-with-the-stevalstrkt01-lora-iot-tracker-stmicroelectronics.pdf

[30] <https://www.thethingsnetwork.org/article/the-things-network-architecture-1>

[31] <https://www.thethingsnetwork.org/docs/network/architecture.html>

[32] Semtech Corporation. LoRa Modulation Basics. Accessed: Nov. 3, 2018. [Online]. Available: <https://www.semtech.com/uploads/documents/an1200.22.pdf>

[33] EU. Short Range Devices (SRD) Operating in the Frequency Range 25 MHz to 1000 MHz | Part 1: Technical Characteristics and Methods of Measurement. 75p. Accessed: Aug. 31, 2018. Available: https://www.etsi.org/deliver/etsi_en/300200_300299/30022001/03.01.01_30/en_30022001v030101v.pdf

[34] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A study of LoRa: Long range & low power networks for the Internet of Things,” *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[35] <https://ieeexplore.ieee.org/document/8648485>

[36] <https://ieeexplore.ieee.org/document/8515030>

[37] <https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/348400/3/08703036.pdf>

[38] <https://ieeexplore.ieee.org/document/8034915>

[39] <https://ieeexplore.ieee.org/document/8095703>

[40] https://tutcris.tut.fi/portal/files/15822869/cts_05_4807.pdf

[41] <https://ieeexplore.ieee.org/document/8553767>