

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA
DIPARTIMENTO GUGLIELMO MARCONI - DEI
INGEGNERIA ELETTRONICA

TESI DI LAUREA MAGISTRALE

in

ARCHITETTURA DEI CALCOLATORI ELETTRONICI M

Progetto di un nodo multi-sensore
LoRaWAN per la stima del
fabbisogno idrico in agricoltura

Relatore:

Prof. Tullio SALMON CINOTTI

Autore:

Simone SINDACO

Correlatori:

Dott. Cristiano AGUZZI

Prof.ssa Eleonora FRANCHI SCARSELLI

Dott. Luca PERILLI

Anno Accademico 2018/19

Sessione I

Indice

Introduzione	7
Obiettivi	8
1 Studio del protocollo LoRaWAN	9
1.1 La tecnologia LoRa	9
1.1.1 Direct Sequence Spread Spectrum (DSSS)	10
1.1.2 Chirp Spread Spectrum (CSS)	12
1.1.3 LoRa Spread Spectrum	12
1.2 Il protocollo LoRaWAN	14
1.2.1 Architettura	14
1.2.2 Classi di funzionamento	15
1.2.3 Indirizzamento e Sicurezza	17
1.3 The Things Network	18
2 Studio dei sensori e del relativo sistema di comunicazione	21
2.1 Teoria e funzionamento del sensore	22
2.1.1 Temperatura	22
2.1.2 Conducibilità Elettrica	22
2.1.3 Volumetric Water Content (VWC) e Permittività	23
2.2 Caratteristiche tecniche ed elettriche del sensore	23
2.2.1 Protocollo DDI	24
2.2.2 Protocollo SDI-12	26
2.2.3 Esempio di comunicazione in modalità DDI	27
3 Progettazione del nodo multi-sensore	29
3.1 B-L072Z-LRWAN1	29
3.1.1 Modalità Low-power	30

3.2	Scelte architettureali	33
3.2.1	Gestione e alimentazione dei sensori	33
3.2.2	Gestione della comunicazione	34
3.2.3	Insertion Detection	35
3.3	Alimentazione	36
3.4	Rilevazione della percentuale di carica delle batterie	40
3.5	PCB	41
3.5.1	Schematico	41
3.5.2	Layout	45
4	Firmware	49
4.1	Start-up	51
4.1.1	Inizializzazione	51
4.1.2	Insertion Detection	53
4.1.3	Riconfigurazione	54
4.1.4	Associazione alla rete LoRaWAN	54
4.2	Funzionamento a regime	55
4.2.1	Attivazione della Run mode e accensione dei sensori	55
4.2.2	Letture e controllo del dato ricevuto	56
4.2.3	Misura della percentuale di carica e invio del frame LoRaWAN	59
4.2.4	Riconfigurazione del timer e attivazione della Stop mode	61
4.3	Test mode	61
5	Stima dei consumi e dei tempi di vita del nodo multi-sensore	63
5.1	Misura dei consumi	63
5.2	Stima della lifetime	68
6	Esempio di funzionamento, ricezione e lettura dei dati tramite The Things Network	69
6.1	Configurazione della rete LoRaWAN	69
6.2	Decoder	70
6.3	Ricezione di un pacchetto LoRaWAN	71
7	Conclusioni	73
7.1	Sviluppi futuri	74

Bibliografia	75
Elenco delle figure	77
Elenco delle tabelle	79

Introduzione

L'agricoltura è il settore in cui vengono impiegate la maggior parte delle risorse idriche mondiali. Secondo l'Organizzazione per la Cooperazione e lo Sviluppo Economico (OCSE), la percentuale di risorse utilizzate nel mondo dal settore primario si aggirerebbe attorno al 70% [1]. Ben il 40% è sfruttato dai soli paesi membri dell'OCSE.

Di tutta l'acqua utilizzata si calcola inoltre che circa il 40% [2] venga perso durante il tragitto dalla fonte all'utilizzatore: dato che, secondo il Consorzio di Bonifica dell'Emilia Centrale, salirebbe addirittura fino al 50% per quanto riguarda il territorio locale [3].

Se si tiene conto dell'incremento della popolazione e, conseguentemente, della crescita della richiesta nel settore agro-alimentare, dei cambiamenti climatici e della salinizzazione dell'acqua dolce, è facile intuire come le attuali risorse idriche mondiali siano destinate ad una rapida diminuzione. Assodato questo trend, si stima che entro il 2050 avremo bisogno complessivamente dell'equivalente delle risorse idriche di oltre due pianeti.

Per ovviare a questo problema è necessario attuare politiche che vertano su una gestione più sostenibile delle risorse. In una visione a breve e medio periodo, una gestione sostenibile può essere ottenuta, per quanto riguarda l'agricoltura, attraverso il risparmio di acqua, l'aumento dell'efficienza con la quale questa viene utilizzata, e, dove possibile, il suo riutilizzo.

Sui primi due punti si focalizza il progetto SWAMP (Smart Water Management Platform) [4], inserito nell'ambito del programma di cooperazione tra Unione Europea e Brasile, il cui obiettivo principale è quello di realizzare un sistema efficiente di distribuzione delle acque irrigue, puntando sulle più recenti tecnologie in campo IoT, Big Data e su modelli fisico-chimici del suolo agricolo [5].

Il sistema descritto in questa trattazione è stato realizzato nell'ambito di tal

progetto, ed è orientato verso l'aumento dell'efficienza di utilizzo delle risorse idriche in ambito agricolo.

Obiettivi

Ciò che di seguito verrà illustrato è la progettazione di un nodo multi-sensore in grado di monitorare il suolo agricolo ed estrarre dati relativi alla temperatura, alla concentrazione di soluti nel terreno e al tasso di umidità. Attraverso tali dati sarà possibile gestire con più precisione ed efficienza la distribuzione di acque irrigue su tutto il suolo in maniera tale da minimizzare gli sprechi.

Il nodo in questione deve interfacciarsi ad una rete di nodi già esistente e che comunica attraverso il protocollo LoRaWAN e pertanto deve essere compatibile con tale standard di comunicazione. Inoltre si è cercato di minimizzare la potenza dissipata dal nodo, in modo tale da allungarne il tempo di vita.

Per i motivi sopra elencati, è stata scelta come unità centrale del dispositivo una board in grado di fornire l'efficienza energetica ricercata e la compatibilità con la tecnologia Long Range (LoRa).

Una volta descritte le caratteristiche che contraddistinguono il protocollo LoRaWAN e i sensori con cui il nodo sarà equipaggiato, si passerà alla progettazione hardware del nodo, in cui verranno esplicate le scelte progettuali. Verranno poi illustrate le dinamiche di funzionamento del nodo e i tratti salienti del firmware installato su di esso. Nella trattazione è stata inoltre inclusa la caratterizzazione dei consumi del nodo, attraverso la quale è stato possibile stimare il tempo di vita del sistema.

Capitolo 1

Studio del protocollo LoRaWAN

LoRaWAN è un protocollo per la comunicazione basato sulla tecnologia LoRa [6]. Mentre quest'ultima quindi si occupa di definire lo standard per la creazione e la gestione del collegamento fisico, LoRaWAN si occupa di definire le regole per la comunicazione e l'architettura del network a livello due e tre del modello ISO-OSI [7].

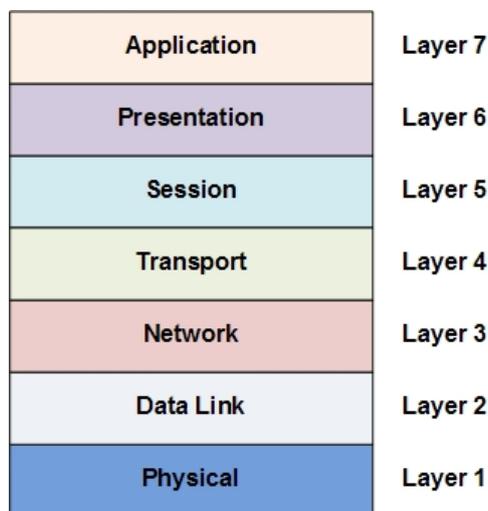


Figura 1.1: Modello ISO-OSI

1.1 La tecnologia LoRa

LoRa, abbreviazione di Long Range, è una tecnica di modulazione a spettro espanso [8], sviluppata da Cycleo e acquistata nel 2012 da Semtech. È

un derivato del Chirp Spread Spectrum (CSS), che a sua volta si basa sulla tradizionale Direct Sequence Spread Spectrum (DSSS). Essendo una modulazione proprietaria non si conoscono ufficialmente i meccanismi, tuttavia ci si può basare sulle sopracitate CSS e DSSS per delinearne i principi di funzionamento, che la rendono la modulazione adatta per le comunicazioni a basso consumo di potenza e a lunga distanza, e che pertanto sono soggette ad un rapporto segnale/rumore (SNR) generalmente basso.

1.1.1 Direct Sequence Spread Spectrum (DSSS)

A partire dal teorema di Shannon–Hartley, si intuisce come aumentando la larghezza di banda, può essere compensata la degradazione del segnale dovuta ad un rapporto segnale-rumore estremamente basso [8].

Il teorema stabilisce la massima velocità con cui l'informazione può essere trasmessa attraverso un collegamento, al variare della larghezza di banda e del rapporto segnale/rumore.

In formule:

$$C = B * \log_2\left(1 + \frac{S}{N}\right)$$

Dove:

- C rappresenta la capacità del canale (in bit/s);
- B rappresenta la Banda;
- $\frac{S}{N}$ rappresenta il rapporto segnale/rumore.

Manipolando algebricamente tale formula, sviluppando in serie di MacLaurin e tenendo conto che $\frac{S}{N}$, per i sistemi considerati, è molto minore di 1, otteniamo:

$$\frac{N}{S} \simeq \frac{B}{C}$$

Se lo SNR diminuisce, e quindi $\frac{N}{S}$ aumenta, allora, a parità di capacità del canale, dovrà aumentare la banda del segnale.

Nella modulazione DSSS quindi si cerca di raggiungere tale risultato, moltiplicando l'informazione per una sequenza prestabilita detta "Chip sequence". Il segnale ottenuto quindi avrà una banda superiore all'originale, dipendentemente dalla lunghezza di un chip. Il rapporto tra la banda del segnale trasmesso (ottenuto tramite il processo denominato "di spreading"

come mostrato in [Figura 1.2](#)) e quella del segnale originale è detto Processing Gain (G_p).

$$G_p = 10 * \log_{10}\left(\frac{R_c}{R_b}\right)[dB]$$

Dove:

- R_c Rappresenta la chip rate [chips/s]
- R_b Rappresenta la bit rate [bits/s]

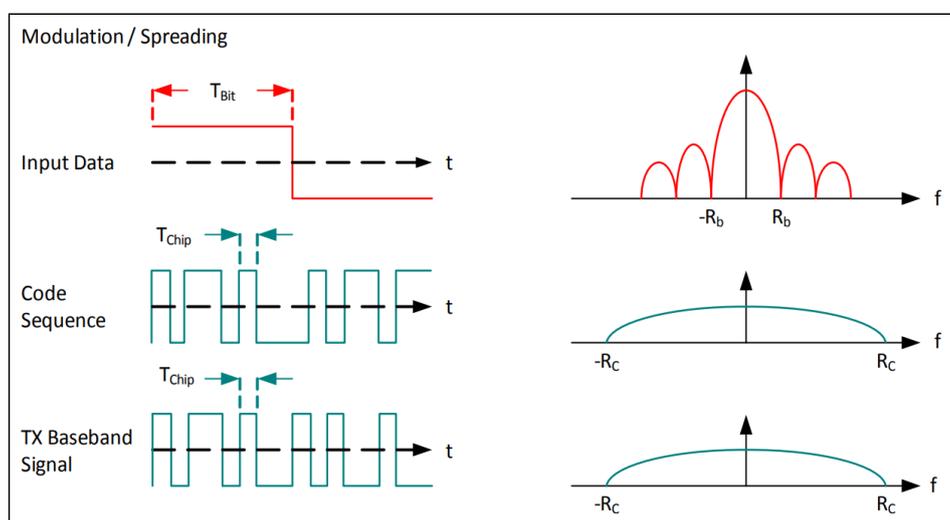


Figura 1.2: Spreading Process [8]

In fase di ricezione spetterà poi al ricevitore ri-moltiplicare il segnale ricevuto per la stessa Chip Sequence con cui era stato moltiplicato il segnale originale dal trasmettitore, durante un processo chiamato De-spreading Process ([Figura 1.3](#)).

Ovviamente, più sarà lunga la Chip Sequence e maggiori saranno i tempi di de-modulazione del segnale da parte del ricevitore, in quanto quest'ultimo dovrà effettuare una correlazione lungo l'intera lunghezza della sequenza di codice. Inoltre l'alto dispendio di energie in fase di sincronizzazione e il bisogno di una sorgente di clock estremamente accurata e quindi costosa, rendono la modulazione DSSS poco adatta in ambiti low power e a basso costo.

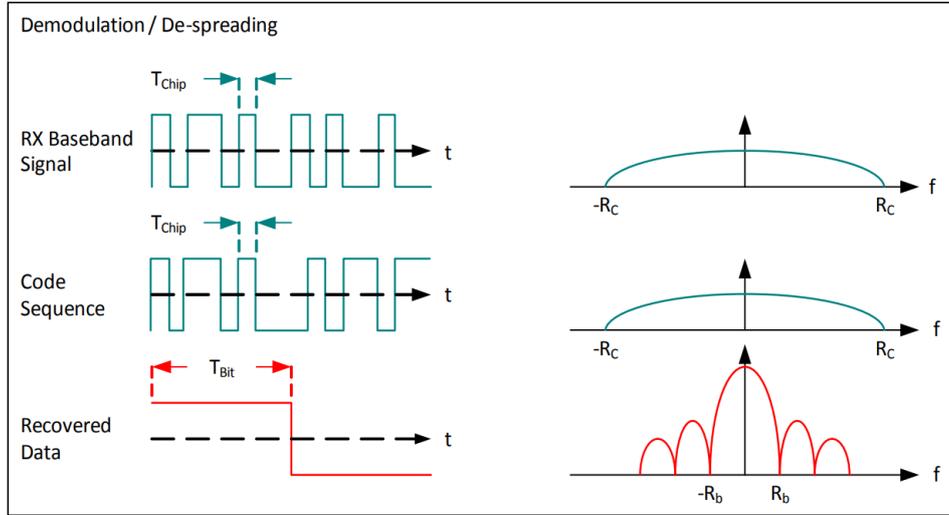


Figura 1.3: De-spreading Process [8]

1.1.2 Chirp Spread Spectrum (CSS)

Il CSS si differenzia dal DSSS perché il segnale trasmesso viene modulato attraverso dei Chirp, ossia dei segnali a frequenza incrementale (up-chirp) o decrementale (down-chirp), ottenuti facendo variare linearmente la portante in un intervallo di frequenze prestabilito.

1.1.3 LoRa Spread Spectrum

A partire dalle caratteristiche del DSSS e del CSS, attraverso la tecnologia LoRa si cerca di rendere fruibili i medesimi risultati in termini di efficienza di trasmissione e ricezione, in sistemi low-power e a basso costo.

Uno dei parametri più importanti della modulazione LoRa è lo Spreading Factor (SF) che può andare da 7 a 12. Il messaggio trasmesso è così diviso in parole (o simboli) da 2^{SF} chip ed ogni parola contiene SF bit dell'informazione originale, come mostrato in [Figura 1.4](#).

Con questo metodo si ha il vantaggio di avere offset di tempo e frequenza, tra trasmettitore e ricevitore, equivalenti, e quindi difficoltà ridotte nella realizzazione del ricevitore.

La relazione tra la bit rate, symbol rate e chip rate è la seguente:

$$R_b = SF * \frac{1}{2^{\text{SF}}} \frac{1}{BW} [bits/s]$$

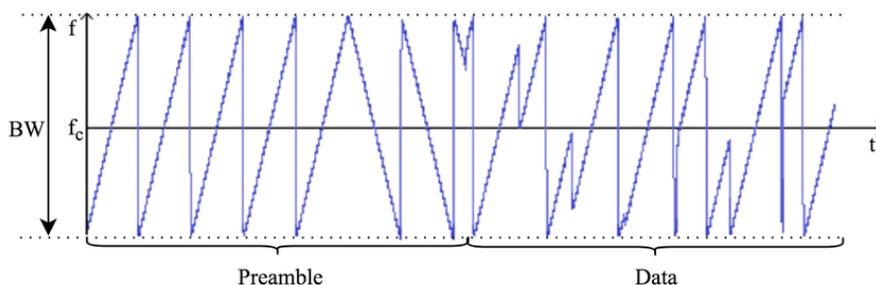


Figura 1.4: Segnale modulato con modulazione LoRa [9]

Dove:

- R_b Rappresenta la bit rate [bits/secondi]
- BW è la larghezza di banda del segnale trasmesso [Hz]
- SF è lo Spreading Factor (da 7 a 12)

Si può notare nella formula precedente la definizione di symbol rate, equivalente a:

$$R_s = \frac{1}{T_s}$$

Dove:

$$T_s = \frac{2^{SF}}{BW}$$

Allora si definisce la Chip Rate come:

$$R_C = R_s * 2^{SF} = \frac{BW}{2^{SF}} * 2^{SF} [chip/sec]$$

In altre parole viene trasmesso un chip per secondo per Hertz della banda della modulazione BW .

Nella modulazione LoRa è incluso inoltre uno schema per la correzione dell'errore atto ad aumentare la robustezza del segnale trasmesso, incrementando la ridondanza d'informazione. La bit rate nominale del segnale originale sarà ottenuta in questo caso come:

$$R_b = SF * \frac{4+CR}{\frac{2^{SF}}{BW}} [bits/sec]$$

Dove:

- CR rappresenta la Code Rate

1.2 Il protocollo LoRaWAN

Mentre quanto visto finora valeva per il livello fisico del modello ISO-OSI, ci si concentri adesso sui livelli due e tre del modello, descritti come precedentemente detto dal LoRaWAN. Il protocollo LoRaWAN è stato standardizzato dalla LoRa Alliance [10], un'associazione no-profit che conta al suo interno l'adesione di oltre 500 membri, tra cui Semtech, proprietaria della modulazione LoRa.

1.2.1 Architettura

L'architettura del network è basata su una topologia a "star of stars", in cui ogni nodo della rete non è connesso ad un particolare gateway, ma è in grado di trasmettere pacchetti a più gateways sfruttando la modulazione LoRa, come illustrato in Figura 1.5. Ogni gateway ha quindi il compito di inoltrare il pacchetto ricevuto sul Network Server, generalmente servendosi di altri standard di comunicazione, come ad esempio il 3G/4G o il Wi-Fi.

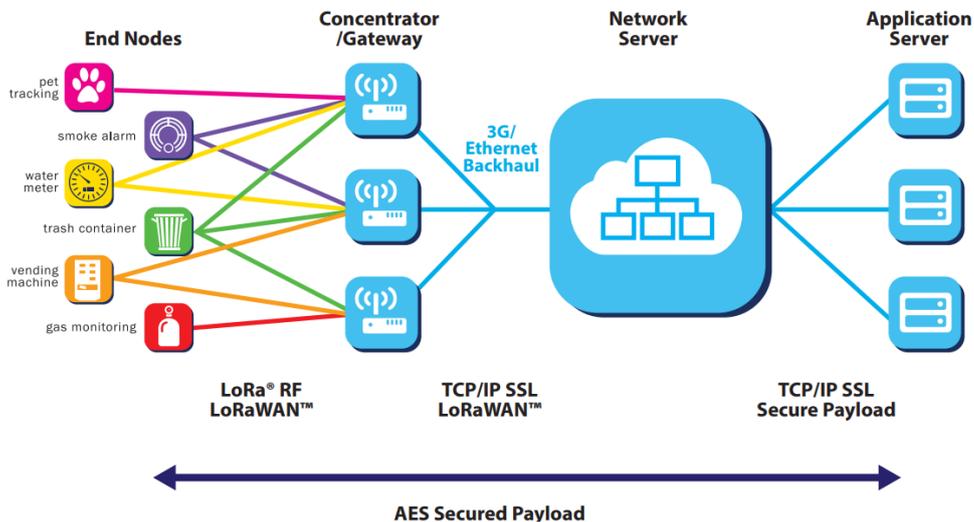


Figura 1.5: Topologia di una rete LoRaWAN [11]

In questo modo viene preservata la lifetime di ogni nodo e aumentato il raggio di copertura dell'intera rete.

Ogni nodo della rete è in grado di inviare un pacchetto ogni qual volta si

abbia bisogno di farlo, generalmente allo scadere di un countdown o quando uno dei parametri monitorati da un sensore cambia. Per questo motivo a livello MAC lo standard LoRaWAN si basa sul protocollo ALOHA.

Nel protocollo ALOHA i nodi sono asincroni e agiscono indipendentemente l'uno dagli altri. Quando più nodi trasmettono contemporaneamente un pacchetto allo stesso gateway avviene una "collisione" ed entrambi i pacchetti vengono scartati dai gateways in ascolto. Ognuno dei nodi trasmettenti cercherà di re-inviare lo stesso pacchetto dopo un intervallo di tempo di lunghezza casuale: casualità che garantisce una bassa probabilità di avere un'altra collisione durante il re-invio del pacchetto.

A partire dall'ALOHA, nello standard LoRaWAN è implementato un meccanismo di Adaptive Data Rate che permette di adattare la velocità di trasmissione, in modo da raggiungere il giusto compromesso tra efficienza energetica e robustezza del collegamento.

1.2.2 Classi di funzionamento

I nodi che costituiscono la rete LoRaWAN possono appartenere a tre classi di funzionamento diverse, in modo da poter soddisfare tutti i possibili casi di utilizzo:

- Classe A: è la classe standard di funzionamento, che implementa la comunicazione bi-direzionale tra nodi e gateways. La comunicazione può essere intrapresa solo dai nodi, in qualsiasi istante temporale. Dopo aver iniziato la comunicazione e trasmesso il pacchetto di dati al gateway (Uplink), il nodo apre due finestre di ricezione, ossia due brevi intervalli temporali, a distanza di un secondo l'uno dall'altro, in cui si mette in ascolto ed è pronto a ricevere i messaggi in arrivo dal server (Downlink). Il processo di trasmissione è illustrato in [Figura 1.6](#). Qualora il server non inviasse alcun messaggio, allora dovrà aspettare la successiva trasmissione da parte del nodo e la conseguente apertura di una nuova finestra di ricezione. Se il server decide di inviare un messaggio al nodo durante la prima finestra di ricezione, non potrà inviarne un altro durante la seconda finestra, e viceversa.

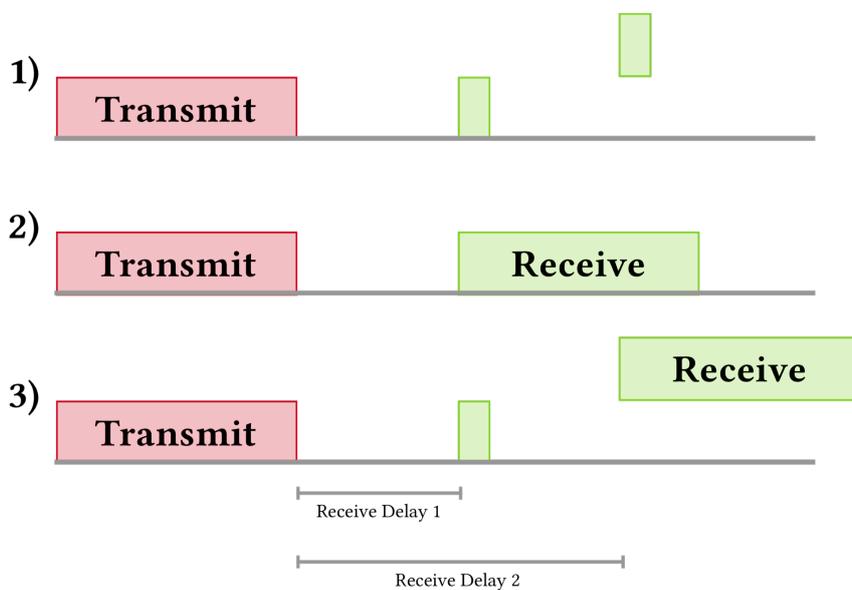


Figura 1.6: Funzionamento di un nodo in Classe A [11]

- Classe B: è un'estensione della Classe A e consente al nodo di aprire periodicamente delle finestre di ricezione. Vengono utilizzati a tal proposito dei segnali temporizzati, detti beacon, trasmessi dal gateway verso il nodo, il quale aprirà una finestra di ricezione indipendentemente dal fatto che abbia prima trasmesso o meno un pacchetto in Uplink.
- Classe C: è un'estensione della Classe A e permette al nodo di tenere sempre aperta una finestra di ricezione, a meno che il nodo non stia trasmettendo un messaggio in Uplink (Figura 1.7). I nodi della Classe C operano quindi con una latenza più bassa rispetto al Classe A, ma con un dispendio di energie superiore.

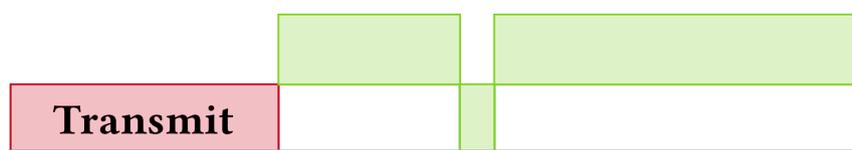


Figura 1.7: Funzionamento di un nodo in Classe C [11]

1.2.3 Indirizzamento e Sicurezza

Per quanto riguarda l'indirizzamento dei messaggi e la sicurezza, lo standard LoRaWAN offre un meccanismo di addressing, di crittografia e di contatori basato sui seguenti identificativi:

- Application Identifier (AppEUI): identificativo a 64 bit assegnato all'applicazione di cui il dispositivo fa parte;
- Device Identifier (DevEUI): identificativo statico a 64 bit proprio del nodo;
- Device Address (DevAddr): identificativo a 32 bit dinamico assegnato al nodo dal Network Server nel momento in cui il dispositivo entra a far parte della rete LoRaWAN;
- Application Key (AppKey): chiave a 128 bit condivisa dal nodo e dall'applicazione;
- Network Session Key (NetSKey): chiave a 128 bit generata nel momento in cui un dispositivo entra a far parte della rete ed utilizzata dal Network Server e dal nodo per generare il Message Integrity Code (MIC) che serve a controllare l'integrità del messaggio. Permette inoltre di mappare gli indirizzi dinamici dei dispositivi negli indirizzi unici (DevEUI e AppEUI);
- Application Session Key (AppSKey): chiave a 128 bit generata nel momento in cui il dispositivo entra a far parte della rete ed utilizzata per criptare e de-criptare il frame payload;
- Frame Counters: contatore dei messaggi in Uplink e Downlink. I messaggi che contengono un frame counter minore di quello aspettato vengono scartati in modo da evitare attacchi di rete di tipo Replay-Attack.

L'attivazione, ossia il processo con cui un nodo entra a far parte del network, può avvenire in due modalità diverse.

Nella modalità Over The Air Activation (OTAA) i nodi devono seguire una procedura di associazione prima di poter scambiare messaggi con il Network Server. Durante questa procedura vengono generate e scambiate le chiavi

sopracitate e vengono assegnati gli indirizzi dinamici. Ogni qualvolta che una nuova sessione dovrà essere aperta, delle nuove chiavi dovranno essere generate e scambiate.

Nella modalità di Activation By Personalization (ABP) invece non vi è alcuna procedura di associazione, in quanto gli identificativi e le chiavi sono già memorizzati nel nodo.

1.3 The Things Network

The Things Network (TTN) [12] è una piattaforma online che offre, gratuitamente e in modo open-source, l'intera infrastruttura di cui una comunicazione basata su protocollo LoRaWAN ha bisogno. La piattaforma si sviluppa secondo un modello detto "Crowdsourcing": un tipo di sviluppo basato cioè sulla collettività. Ogni utente registrato al sito può infatti installare uno o più gateway a proprie spese, per estendere la copertura della rete globale basata su LoRaWAN. Attraverso tali gateway, mappati in [Figura 1.8](#), i pacchetti in Uplink e in Downlink sono re-indirizzati da e verso il Network Server gestito dalla piattaforma nelle modalità descritte precedentemente. Si noti in [Figura 1.9](#) la presenza di un gateway appartenente ad ARCES (Advanced Research Center on Electronic Systems), in provincia di Reggio Emilia, presso una fattoria dell'azienda agricola "Il Girasole": punto in cui è installata la rete di nodi-sensore a cui il dispositivo progettato dovrà interfacciarsi.

The Things Network offre inoltre la possibilità di creare e gestire diverse applicazioni in base alla necessità di cui si dispone. Una volta creata una nuova applicazione, ad essa è assegnato un identificatore, attraverso cui è distinta dalle altre applicazioni del network. È possibile quindi assegnare diversi dispositivi alla stessa applicazione, e gestire i pacchetti in Downlink e Uplink attraverso un decoder, che si occupa di interpretare le informazioni contenute nel payload del frame.

La piattaforma consente di gestire la propria applicazione anche a livello di sicurezza, consentendo la generazione ed il controllo di tutti gli identificatori e le chiavi che caratterizzano il protocollo.

CAPITOLO 1. STUDIO DEL PROTOCOLLO LORAWAN



Figura 1.8: Mappa dei gateways attualmente connessi al network TTN [12]

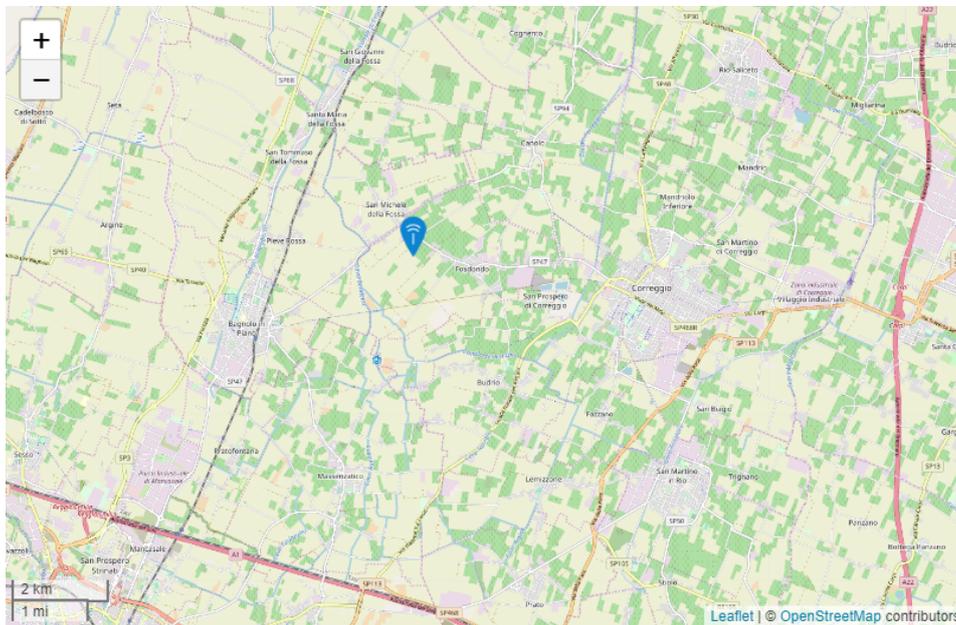


Figura 1.9: Mappa dei gateways attualmente connessi al network TTN, località Fosdondo (RE)[12]

1.3. THE THINGS NETWORK

Capitolo 2

Studio dei sensori e del relativo sistema di comunicazione

Il sensore adottato per la progettazione è un ECH₂O GS3 [13] della METER: azienda americana nata dalla fusione di Decagon Devices e UMS AG [14].

Il sensore, mostrato in Figura 2.1, consente la misura di alcuni parametri del terreno, quali la temperatura, la conducibilità elettrica e la permittività elettrica del suolo, attraverso cui può essere ricavato il Volumetric Water Content del terreno monitorato.

Il dispositivo è costituito da un corpo principale, che ospita l'unità centrale per la gestione e il controllo del sensore, dal quale fuoriescono tre aghi attraverso cui vengono effettuate alcune misurazioni. Il GS3 è dotato inoltre di un connettore di tipo Jack da 3.5mm con il quale viene alimentato e ci si può interfacciare ad esso.

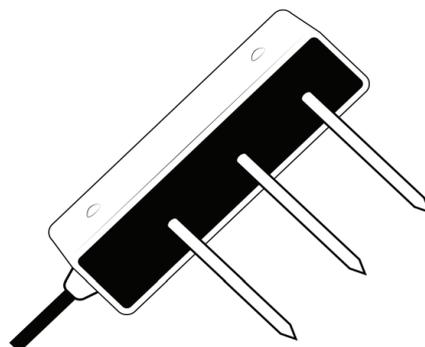


Figura 2.1: Sensore METER ECH₂O GS3 [13]

2.1 Teoria e funzionamento del sensore

In questa sezione analizzeremo il meccanismo attraverso il quale il sensore è in grado di estrarre i dati dal suolo, una volta che la procedura di installazione è stata eseguita (Figura 2.2).

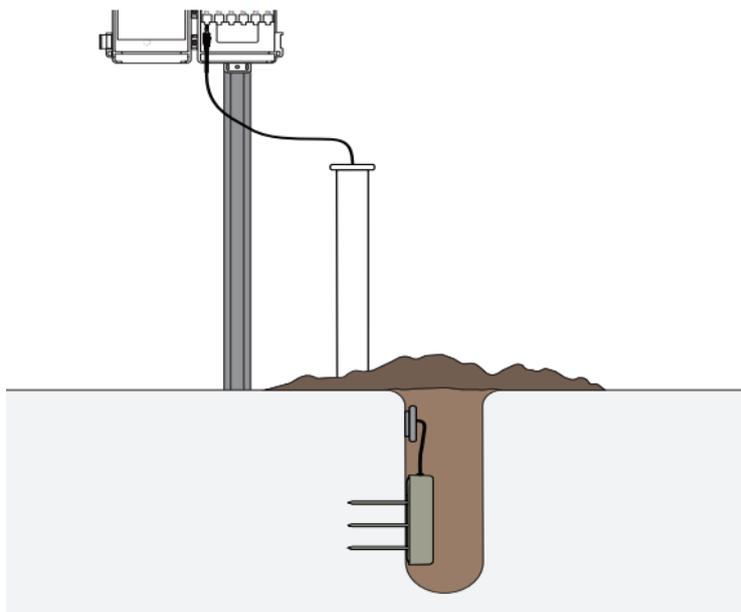


Figura 2.2: Esempio di installazione del sensore [13]

2.1.1 Temperatura

Per la misura della temperatura il sensore sfrutta un termistore posizionato vicino uno degli aghi, ed in grado di monitorare la temperatura superficiale dell'ago e fornirci il valore in gradi Celsius.

Il GS3 può effettuare una misura in un range di valori di temperatura che va da -40°C fino a 60°C con una risoluzione di 0.1°C e una precisione di $\pm 1^{\circ}\text{C}$ [13].

2.1.2 Conducibilità Elettrica

La conducibilità elettrica rappresenta la capacità di un corpo di lasciare fluire attraverso di esso della corrente elettrica, qualora sia sottoposto ad un campo elettrico. Per misurare il valore di conducibilità elettrica, il sensore applica

CAPITOLO 2. STUDIO DEI SENSORI E DEL RELATIVO SISTEMA DI COMUNICAZIONE

una corrente alternata a due elettrodi e misura la resistenza presente tra questi ultimi. Una volta trovato il valore di resistenza viene calcolata la conduttanza facendone il reciproco.

$$G = \frac{1}{R}[S]$$

Dopodiché la conduttanza viene moltiplicata per la distanza tra gli elettrodi e divisa per la loro area, restituendo così il valore di conducibilità elettrica.

$$EC = G * \frac{d}{A}[dS/m]$$

I valori misurati dal sensore possono variare tra 0 dS/m e 25 dS/m, con una risoluzione di 0.001 dS/m ed un'accuratezza del $\pm 5\%$ tra 0 e 5 dS/m e del $\pm 10\%$ tra 5 e 25 dS/m [13].

2.1.3 Volumetric Water Content (VWC) e Permittività

Per quanto riguarda la misura del Volumetric Water Content, questa è ottenuta a partire dal valore di permittività. Il sensore genera un campo elettromagnetico alla frequenza di 70 MHz attraverso gli aghi. La carica immagazzinata a causa del campo elettrico generato è quindi misurata dal sensore e sarà direttamente proporzionale alla permittività elettrica del mezzo nel quale il sensore è inserito. L'unità centrale del sensore provvederà quindi a calcolare il valore di permittività elettrica che potrà essere poi convertito nel Volumetric Water Content attraverso un'equazione di calibrazione che dipende dal mezzo in cui il sensore è inserito. Tale equazione è fornita nello User Manual del GS3 e permette di raggiungere una precisione del $\pm 5\%$. Se si vogliono effettuare delle misurazioni con una precisione maggiore (fino al $\pm 1\%$) vengono forniti da METER alcuni strumenti per la calibrazione manuale del sensore.

I valori ottenuti di VWC applicando l'equazione di calibrazione fornita nello User Manual hanno una risoluzione di $0.002 \frac{m^3}{m^3}$ per valori compresi tra 0 e $0.4 \frac{m^3}{m^3}$ e di $0.001 \frac{m^3}{m^3}$ per valori superiori a $0.4 \frac{m^3}{m^3}$ [13].

2.2 Caratteristiche tecniche ed elettriche del sensore

Il sensore è in grado di operare con una tensione di alimentazione che varia tra 3.6 VDC e 15 VDC e, a seconda dello sforzo operativo compiuto, può

2.2. CARATTERISTICHE TECNICHE ED ELETTRICHE DEL SENSORE

assorbire una corrente che va da un minimo di 0.5 mA ad un massimo di 30 mA durante una misura, ed è circa di 0.03 mA durante la fase inattività.

Una volta che il sensore è alimentato attraverso gli appositi pin di alimentazione e ground del connettore Jack (Figura 2.3), il dispositivo impiegherà al massimo 100 ms per accendersi, ed un intervallo di tempo compreso tra i 150 ms e i 200 ms per effettuare la rilevazione dei tre parametri sopra descritti.

I dati estratti dalla misurazione sono quindi inviati tramite un protocollo



Figura 2.3: Connettore di tipo Jack maschio per l'alimentazione ed il controllo del sensore GS3

seriale attraverso il pin dati del connettore Jack e possono essere quindi letti in diverse modalità. METER mette a disposizione un Data Logger per la raccolta dei dati, i quali sono quindi elaborati tramite alcuni tools distribuiti dalla stessa azienda.

Se non ci si volesse servire del Data Logger, METER fornisce anche le specifiche sul protocollo di comunicazione seriale, in modo che l'acquisizione e l'elaborazione dei dati possa essere gestita e personalizzata direttamente dall'utente.

I protocolli seriali con cui il sensore si interfaccia sono due, il protocollo DDI e il protocollo SDI-12 [15].

Il sensore GS3 è in grado di lavorare quindi in due modalità, a seconda del protocollo scelto.

2.2.1 Protocollo DDI

Il protocollo seriale DDI è ideale per sistemi che prevedono un collegamento a stella tra più sensori ed un Data Logger, e in cui i sensori sono attivati e controllati uno per volta. Permette l'implementazione di una comunicazione di tipo simplex, ossia mono-direzionale, dal sensore verso l'unità per l'acquisizione dei dati (che generalmente è costituita da un microcontrollore).

CAPITOLO 2. STUDIO DEI SENSORI E DEL RELATIVO SISTEMA DI COMUNICAZIONE

Quando è in uso il protocollo DDI, la modalità di funzionamento del sensore è quella illustrata in [Figura 2.4](#). Non appena il GS3 è alimentato viene inizializzata la procedura di avvio (durata massima 100 ms) ed una volta terminata, il sensore effettua la misurazione dei tre parametri precedentemente descritti (per massimo 200 ms). La data line è tenuta alta (livello logico alto) dal sensore per indicare che è in corso la lettura dei valori. Terminata l'acquisizione, i dati ottenuti vengono inseriti in un frame ed inviati attraverso la data line.

Finita la trasmissione del frame, il sensore abilita automaticamente la modalità di comunicazione SDI-12 e qualora si volesse procedere con una nuova misurazione è necessario riavviare il dispositivo per ripetere l'intera procedura di avvio, acquisizione ed invio dati.

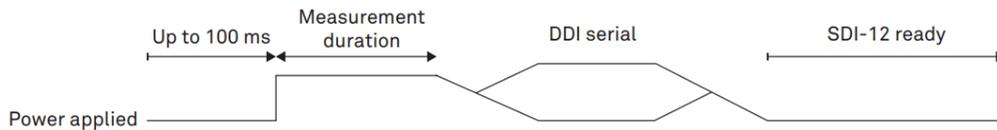


Figura 2.4: Modalità di comunicazione DDI [15]

La comunicazione in DDI mode è configurata in modo da avere una baud-rate di 1200 baud/s, un bit di start (basso), otto bit di informazione (codificati secondo lo standard ASCII) e un bit di stop (alto). È assente il bit di parità. Un esempio di simbolo DDI è riportato in [Figura 2.5](#).

Tale configurazione rende la comunicazione DDI compatibile alla gestione tramite UART.

A livello elettrico, un bit uguale a 1 corrisponde ad una tensione sulla data line che può andare da 2.8 V a 3.9 V; un bit uguale a 0 invece corrisponde ad una tensione compresa tra -0.3 V e 0.8 V.

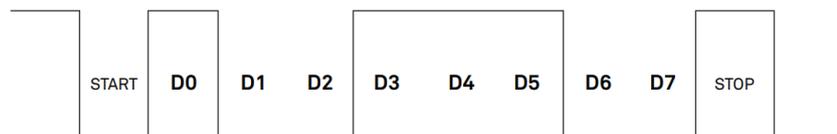


Figura 2.5: Esempio di simbolo in DDI mode, trasmissione del carattere 0x39 [15]

La trasmissione completa di un set di valori è costituita dall'invio del carattere <TAB> (0x09), succeduto dai tre valori di permittività, temperatura e

2.2. CARATTERISTICHE TECNICHE ED ELETTRICHE DEL SENSORE

conducibilità, separati da un carattere <Space> (0x20). Seguono in ordine un <Carriage Return> (0x0D), un carattere indicante il modello del sensore ed un carattere contenente il checksum per la verifica della correttezza del messaggio ricevuto.

In sintesi avremo:

<TAB><Perm> <Temp> <Cond><CR><Model><CS>

Il checksum è ottenuto sommando tutti i byte ricevuti, a partire dal <TAB> e fino al carattere contenente il modello. Se il checksum calcolato in ricezione è uguale a quello contenuto nel messaggio ricevuto allora quest'ultimo è considerato corretto.

2.2.2 Protocollo SDI-12

Il protocollo SDI-12 è generalmente impiegato in sistemi multi-slave in cui più sensori condividono lo stesso bus dati e la stessa alimentazione. Ad ogni sensore è collegato un indirizzo univoco alfanumerico, al quale il Data Logger fa riferimento per inviare dei comandi ai sensori e ricevere da essi i dati riferiti alle misure. Utilizzando tale protocollo quindi può essere implementata una comunicazione di tipo half-duplex, ossia bi-direzionale, in cui il bus dati può essere utilizzato o dal Data Logger o da uno dei sensori.

La modalità di funzionamento SDI-12 è, come già detto, differente dalla modalità DDI. La procedura di accensione sarà la stessa: non appena il sensore è alimentato, verrà completato un intero ciclo di funzionamento in modalità DDI. Dopodiché viene automaticamente attivata la modalità SDI-12. Una volta entrato in SDI-12 mode, una nuova procedura di misura può essere triggerata inviando un comando dal Data Logger (Figura 2.6) attraverso la linea dati ed indirizzato al sensore di cui si vuole attivare una lettura. Oltre alla lettura dei valori di permittività, temperatura e conducibilità,

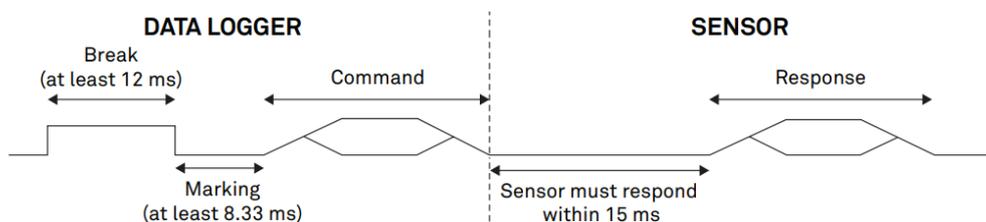


Figura 2.6: Comunicazione in SDI-12 mode tra Data Logger e sensore [15]

possono essere richieste altre informazioni al sensore, oppure una serie di azioni possono essere invocate attraverso una lista di comandi riportati nello User Manual. La configurazione della comunicazione, illustrata in [Figura 2.7](#), è diversa rispetto al protocollo DDI, in quanto prevede la presenza di un bit di parità, preceduto da 7 bit di informazione (8 bit nel DDI). A livello elettrico inoltre la logica è invertita: ad un 1 logico corrisponde un segnale basso, mentre ad uno 0 logico corrisponde un segnale alto. I valori di soglia tra segnale alto e basso sono li stessi indicati per il protocollo DDI.

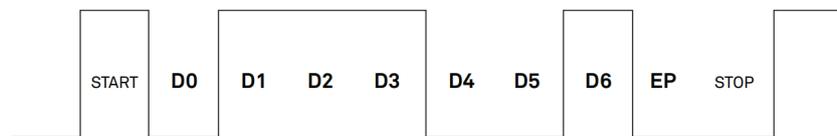


Figura 2.7: Esempio di simbolo in SDI-12 mode, trasmissione del carattere 0x31 [15]

Il protocollo SDI-12, rispetto al DDI, implica una gestione ed un controllo più difficili, in quanto implementa una comunicazione half-duplex, piuttosto che una mono-direzionale come il DDI. Inoltre in modalità SDI-12 si ha la necessità di triggerare una nuova misurazione attraverso l'invio di un comando al sensore. Per questo motivo i sensori devono essere costantemente alimentati. Ciò rappresenta un notevole svantaggio in applicazioni low-power, in quanto il sensore ha un consumo di corrente pari a 0.03 mA in fase di idle; consumo che deve essere quindi moltiplicato per il numero di sensori che il nodo ospita.

2.2.3 Esempio di comunicazione in modalità DDI

In [Figura 2.8](#) è mostrato un esempio di funzionamento del sensore in modalità DDI. La forma d'onda verde rappresenta la tensione di alimentazione fornita al sensore, mentre quella blu rappresenta il segnale sulla data line.

Si noti come, una volta fornita la giusta tensione di alimentazione al sensore, inizi la procedura di avvio, seguita dalla fase di misurazione, in cui la data line è tenuta alta dal dispositivo.

Trascorsi 73 ms dal momento in cui il sensore è stato attivato, si può osservare la trasmissione del frame contenente i valori misurati relativi ai tre parametri, nelle modalità descritte precedentemente.

2.2. CARATTERISTICHE TECNICHE ED ELETTRICHE DEL SENSORE

Nell'esempio in questione inoltre, la tensione di alimentazione fornita al sensore è di 3.8 V, ed un 1 logico sulla data line corrisponde ad un valore di tensione di 3.5 V, in linea con quanto specificato nella [sottosezione 2.2.1](#).

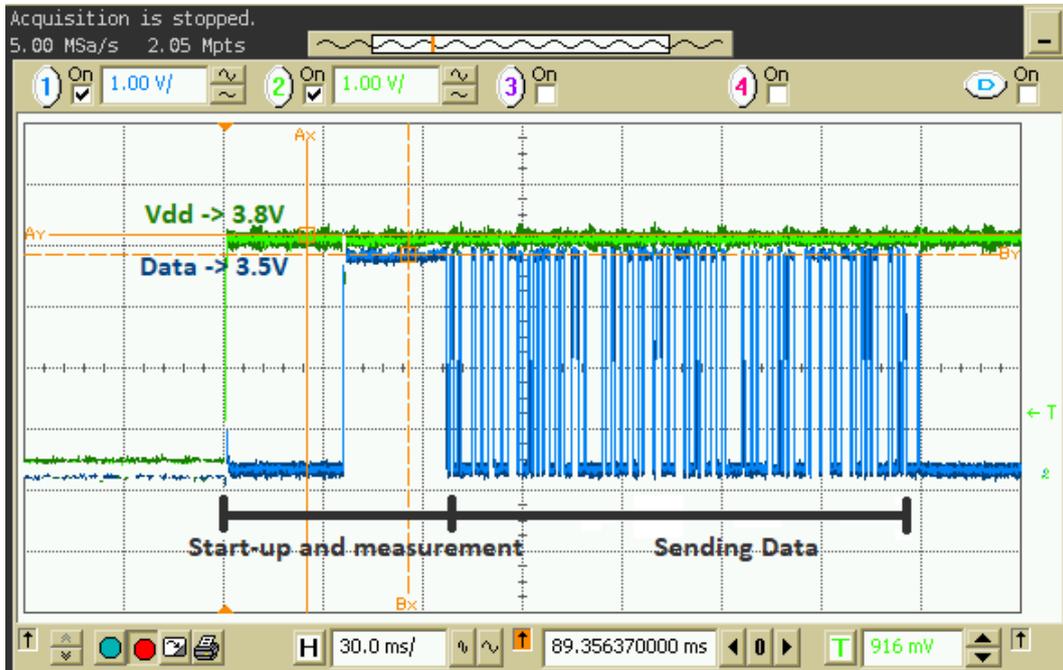


Figura 2.8: Trasmissione di un frame in DDI mode

Capitolo 3

Progettazione del nodo multi-sensore

A partire dallo studio del sensore e del protocollo LoRaWAN sono state quindi effettuate delle scelte progettuali che in questo capitolo verranno affrontate e spiegate.

3.1 B-L072Z-LRWAN1

Il dispositivo scelto come unità centrale del nodo multi-sensore è un B-L072Z-LRWAN1 [16] di STMicroelectronics che include un modulo all-in-one CMWX1ZZABZ-091 sviluppato da Murata, azienda giapponese produttrice di componentistica elettronica. Tale modulo integra al suo interno un MCU STM32L072CZ [17], sviluppato da ST, ed un tranciever SX1276, sviluppato da Semtech, che rende il dispositivo compatibile con la tecnologia LoRa e con la comunicazione di tipo Spread-Spectrum.

Il microcontrollore mette inoltre a disposizione dell'utilizzatore un ADC a 12-bit a 16 canali e numerosi sistemi di comunicazione (come ad esempio SPI, I²C, UART) per l'integrazione e l'interfacciamento con le periferiche. Permette l'attivazione di diverse modalità low-power per limitare il consumo di potenza sia quando il dispositivo sta eseguendo delle operazioni (run) che quando è in stato di riposo (idle). Tali modalità sono in realtà derivanti dalle funzionalità del core ARM Cortex-M0+ integrato nel device.

La discovery include inoltre un connettore RF SMA per la connessione

di un'antenna (inclusa), il debugger e diverse modalità per la gestione dell'alimentazione.

3.1.1 Modalità Low-power

Le modalità low-power messe a disposizione dal microcontrollore sono cinque: una per la fase di run e quattro per la fase di idle, ognuna delle quali garantisce prestazioni in termini energetici, e funzionalità differenti. Generalmente, più il risparmio energetico è invasivo, peggiori sono le prestazioni del device in termini di reattività (es. performance, wake-up time).

Le diverse modalità possono essere abilitate e disabilitate attraverso la scrittura del System Control Register (SCR).

Low-power Run mode

La Low-power Run mode è l'unica modalità low-power per la fase di run. La riduzione dei consumi in tale modalità è ottenuta essenzialmente abbassando drasticamente la frequenza di funzionamento del microcontrollore, che passa da un massimo di 32 MHz ad un massimo di 131 kHz.

A tale frequenza alcune delle periferiche di cui il microcontrollore è dotato non sono più utilizzabili, come ad esempio l'ADC o l'interfaccia USB.

Oltre a diminuire la frequenza di funzionamento del dispositivo e disattivare alcune periferiche, il regolatore di tensione interno viene inizializzato in low-power mode, settando il bit LPRUN del Power Control Register (PWR_CR). Il regolatore lavorerà pertanto in quello che è chiamato Range 2 di funzionamento (per distinguerlo dal Range 1 della normal mode). In tale range di funzionamento, l'uscita del regolatore sarà impostata a 1.5 V (1.8 V nella normal mode) e pertanto verrà abbassata la frequenza di funzionamento del microcontrollore. Ciò comporta un abbassamento della potenza massima fornita al core e alle periferiche ed un conseguente rallentamento nelle operazioni di lettura e scrittura dei registri. Il contenuto dei registri e della SRAM è preservato.

Se tutte le periferiche sono disabilitate e la frequenza di funzionamento è settata alla massima raggiungibile (131 kHz) il Current Consumption si aggira attorno ai 45 μA .

In Low-power Run mode quindi, al contrario delle altre modalità low-power,

la CPU continua a lavorare, anche se le sue performance sono ridotte.

Se, per quanto detto, un abbassamento della frequenza di clock comporta un calo della potenza dinamica, l'abilitazione della modalità low-power per il regolatore di tensione contribuisce ad una diminuzione della potenza statica, permettendo in sintesi di avere una notevole e generale riduzione dei consumi.

Sleep mode

In Sleep mode il core è arrestato, mentre tutte le periferiche sono abilitate e il regolatore funziona in normal mode (Range 1, con uscita a 1.8 V). La riduzione della potenza dissipata è pertanto minima e, tra le cinque modalità low-power, la Sleep mode è sicuramente quella in cui il power consumption è maggiore. Secondo il datasheet infatti, con tutte le periferiche disabilitate e ad una frequenza di clock di 16 MHz, il dispositivo ha un consumo di corrente di circa 1 mA, rispetto alle correnti dell'ordine del μA o del nA che possono essere raggiunte nelle altre modalità. Un vantaggio derivante dall'uso della Sleep mode è però la rapidità con cui il dispositivo può uscire da tale modalità per entrare in normal mode (quando un interrupt è ricevuto dal microcontrollore), che si aggira intorno ai $0.36 \mu s$, contro le tempistiche almeno 10 volte superiori delle altre low-power modes.

Low-power Sleep mode

La Low-power Sleep mode può essere vista come un miglioramento, in termini energetici, della Sleep mode, in quanto le funzionalità del core e delle periferiche rimangono identiche a quest'ultima, con la differenza che il regolatore di tensione è forzato a lavorare nel Range 2, come per la modalità Low-power Run, e che la frequenza di funzionamento è abbassata ad un massimo di 131 kHz. Per questo motivo non tutte le periferiche sono abilitate e quelle che hanno un funzionamento dipendente dalla frequenza devono essere re-inizializzate per lavorare correttamente.

Il consumo di corrente in Low-power Sleep mode, alla massima frequenza raggiungibile e con tutte le periferiche disabilitate, è di circa $19.5 \mu A$ (varia con la temperatura).

Stop mode

La Stop mode è senz'altro la modalità low-power che consente di raggiungere le prestazioni migliori in termini di risparmio energetico, pur preservando la funzionalità della SRAM e dei registri.

In Stop mode il regolatore di tensione è impostato in modalità low-power (Range 2 di funzionamento) e tutti i segnali di clock sono disabilitati, ad eccezione del LSE (Low Speed External Oscillator) e del LSI (Low Speed Internal Oscillator). Questi ultimi rimangono abilitati per dare la possibilità al dispositivo di risvegliarsi dalla Stop mode ed entrare in Run mode, allo scadere di un timer impostato tramite Real Time Clock (RTC), oltre che all'arrivo di un generico interrupt al microcontrollore.

Con tutte le periferiche disabilitate è previsto un consumo di circa $1 \mu A$.

Standby mode

In Standby mode il regolatore di tensione interno è spento e di conseguenza sono spenti tutti i dispositivi e le periferiche da esso alimentati (tra cui anche il core). Tutti i dati contenuti in SRAM e nei registri non sono mantenuti, ad eccezione di quelli contenuti nei registri per la gestione del risveglio. Quest'ultimo viene attivato quando sopraggiunge un segnale di RESET esterno al dispositivo, o quando viene ricevuto un segnale di Wake-up attraverso uno dei tre pin appositi.

Low-power Mode	Wake-Up Time	Current Consumption
Low-power Run	$3 \mu s$	$45 \mu A$
Sleep	$0.36 \mu s$	$1000 \mu A$
Low-power Sleep	$32 \mu s$	$19.5 \mu A$
Stop	$3.5 \mu s$	$1 \mu A$
Standby	$50 \mu s$	$0.29 \mu A$

Tabella 3.1: Confronto tra i wake-up time e il current consumption delle modalità low-power

Per quanto detto, la Stop mode è stata la modalità scelta per abbattere i consumi, in quanto permette una notevole riduzione del Current Consumption, pur preservando i dati contenuti nei registri e nella SRAM e mantenendo un

wake-up time breve, secondo solo alla Low-power Run mode e alla Sleep mode, che però non consentivano di raggiungere un risparmio energetico pari a quello della Stop mode.

Una sintesi di quanto descritto è mostrata in [Tabella 3.1](#).

3.2 Scelte architettureali

Il nodo ospiterà al suo interno fino a sei sensori; pertanto si è scelto di effettuare le misurazioni in successione, un sensore per volta, in modo che la procedura di misura di uno non venga influenzata da quella degli altri quando più sensori sono posti a diverse profondità nel medesimo punto del terreno. Tutto ciò va ovviamente a discapito di un aumento dei consumi da parte del nodo, poiché questo dovrà restare acceso per un tempo maggiore; aumento di poco conto se si tiene in considerazione che il dispendio di energie principale deriva dall'alimentazione stessa dei sensori durante la misura e dall'energia che serve al nodo per inviare un frame tramite protocollo LoRaWAN, come verrà spiegato nella [sezione 5.1](#).

È stato scelto inoltre di utilizzare i sensori in modalità DDI piuttosto che in SDI-12, in modo da poterli alimentare solo durante la misurazione e non avere un ulteriore dispendio di energie da parte di essi nel momento in cui alcuna rilevazione dei parametri è in corso.

3.2.1 Gestione e alimentazione dei sensori

Il controllo dell'alimentazione dei sensori è stato effettuato tramite l'utilizzo di due Analog Switch bidirezionali di tipo SP3T (Single Pole Three Throw), ossia dispositivi operanti come multiplexer/de-multiplexer in grado di collegare i terminali COM e NOx (NO0, NO1, NO2), selezionabili tramite due ingressi digitali (IN1 e IN2) e pilotati dalla board B-L072Z-LRWAN1 ([Figura 3.1](#)).

Il dispositivo designato è un TS5A3359 [18] di Texas Instruments, la cui scelta è stata effettuata a partire dalle caratteristiche del dispositivo e dalle performance che esso garantisce. In particolare, volendo che la caduta di tensione sullo switch fosse minima, uno dei parametri fondamentali per la scelta del dispositivo è stato quello della resistenza interna in stato di ON. Il TS5A3359 garantisce infatti una ON_State Resistance di 1Ω .

Altra caratteristica importante è quella riguardante il Current Consumption

3.2. SCELTE ARCHITETTURALI

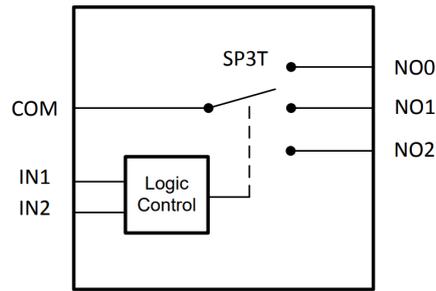


Figura 3.1: Schematico esemplificativo di un SP3T [18]

IN2	IN1	COM TO NO, NO TO COM
L	L	OFF
L	H	COM=NO0
H	L	COM=NO1
H	H	COM=NO2

Tabella 3.2: Tabella di funzionamento dello switch TS5A3359

del dispositivo stesso, che si è cercato di minimizzare. Quest'ultimo, per il device in questione, è dell'ordine di qualche decina di nA , dipendentemente dalle condizioni in cui lo switch opera.

Per quanto riguarda le caratteristiche elettriche, lo switch può operare con tensioni comprese tra 0 V e la sua tensione di alimentazione (massima a 5.5 V), e permette il passaggio di una corrente massima di 200 mA . In [Tabella 3.2](#) è riassunto il funzionamento del dispositivo.

3.2.2 Gestione della comunicazione

Anche per quanto riguarda la comunicazione si è optato per la scelta del componente TS5A3359, in modo da permettere la lettura dei sei sensori, attivi uno per volta, utilizzando una sola delle due UART presenti nel microcontrollore.

Due switch sono stati quindi utilizzati in modo indipendente (uno per i primi tre sensori, l'altro per i restanti) ma con i due COM cortocircuitati. È stato possibile fare ciò perché l'uscita del TS5A3359 è posta in uno stato di alta impedenza quando la configurazione degli ingressi digitali è "L L". Ciò gli permette quindi di non influenzare il segnale portato sul pin di ricezione della UART dallo switch antagonista.

Oltre alle caratteristiche precedentemente riportate (si rimanda al datasheet per eventuali precisazioni), di notevole importanza per quanto riguarda la comunicazione è il dato relativo alla bandwidth del dispositivo, che si aggira attorno ai 75 MHz, come mostrato in [Figura 3.2](#) (dipende dalla temperatura e dall'alimentazione).

Quanto detto rende il dispositivo compatibile a poter gestire i segnali derivanti dai sensori e contenenti i valori dei parametri misurati, i quali viaggiano ad una frequenza di circa 12 KHz (1200 simboli/s con 10 bit per simbolo).

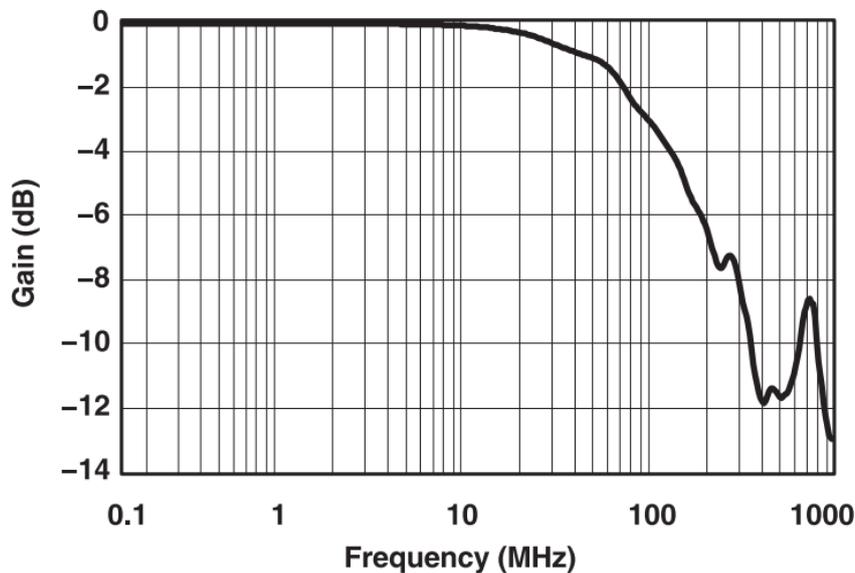


Figura 3.2: Larghezza di banda a 5 V di alimentazione [18]

3.2.3 Insertion Detection

Per coprire tutti i possibili casi di utilizzo del nodo multi-sensore si è implementato un meccanismo attraverso il quale è possibile capire quanti e quali dei sei sensori sono effettivamente connessi al nodo, in modo da permettere una riconfigurazione a runtime del firmware. Una volta verificata la presenza o meno di ciascun sensore, la procedura di misurazione è avviata solo per quelli connessi al nodo, mentre vengono completamente trascurati gli altri.

La realizzazione del meccanismo, chiamato "Insertion Detection", è stata resa possibile grazie all'uso di un connettore femmina di tipo Jack da 3.5 mm.

3.3. ALIMENTAZIONE

In particolare si è scelto un connettore dotato di tip-switch, ossia di un pin aggiuntivo la cui funzionalità è proprio quella di verifica della presenza o dell'assenza di un connettore Jack maschio inserito.

I pin di cui la presa dispone sono in totale quattro, tra i quali, oltre al tip switch, sono presenti un pin per l'alimentazione (tip), uno per il ground (sleeve) ed un pin per i dati (ring). Come si evince dallo schema in [Figura 3.3](#), quando nessun

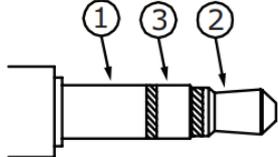
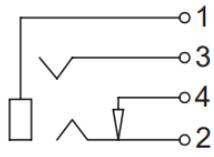
Model No.	SJ-3524-SMT
	
PIN	
1	sleeve
2	tip
3	ring
4	tip switch

Figura 3.3: Configurazione dei pin del modello SJ-3523-SMT [19]

senso è inserito nella presa Jack, allora il pin tip-switch è cortocircuitato al pin di alimentazione (tip). In questo modo se la presa è alimentata avremo una tensione sul tip-switch pari a quella di alimentazione.

Nel momento in cui un sensore viene connesso alla presa Jack, grazie ad un sistema meccanico a molla, il tip-switch verrà scollegato dal tip, e si troverà ad un potenziale indefinito. Per ovviare a questo problema è stata inserita una resistenza di pull-down tra il tip-switch e ground, in modo che non appena un sensore viene inserito, il pin 4 sarà cortocircuitato a massa e la tensione su di esso sarà nulla.

3.3 Alimentazione

Si è scelto di alimentare il nodo multi-sensore attraverso una serie di tre pile di tipo AA da 1.5 V, in grado di fornire quindi una tensione totale di 4.5 V nominali.

Benchè i sensori potessero essere alimentati ad una tensione compresa tra 3.6

VDC e 15 VDC, si è scelto di inserire un regolatore di tensione lineare di tipo LDO (Low Dropout) tra la tensione di batteria fornita dalle pile e i sei sensori. Si è scelto di utilizzare un regolatore LDO in quanto permetteva una dinamica della tensione in ingresso maggiore rispetto ad un normale regolatore di tensione lineare. Un regolatore di tensione lineare riesce a fornire una tensione in uscita pari al massimo alla differenza tra la tensione in ingresso e quella di dropout. Ad esempio, se la tensione di alimentazione è di 4.5 V e il dropout è di 500 mV, la tensione massima erogabile dal regolatore sarà di 4 V.

Un vantaggio derivante dall'utilizzo di un convertitore LDO è quindi la possibilità di raggiungere tensioni in uscita molto vicine a quelle in ingresso, poichè hanno una tensione di dropout molto minore rispetto ad un normale regolatore di tensione lineare.

Il dispositivo scelto per l'alimentazione dei sensori è un ADP163 [20] prodotto da Analog Devices. Le caratteristiche elettriche che differenziano questo regolatore dagli altri sul mercato sono una tensione di dropout di circa 30 mV (Figura 3.4), con una corrente erogata pari alla massima richiesta dai sensori (30 mA), ed una corrente di riposo pari a 860 nA con una corrente sul carico di 1 μ A.

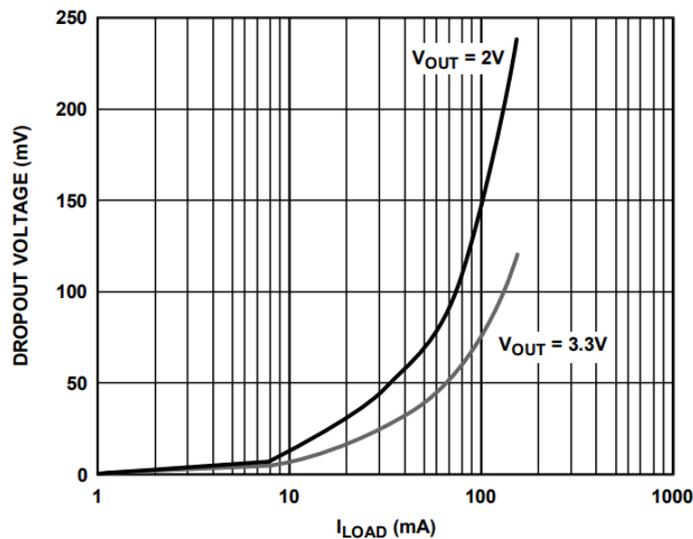


Figura 3.4: Caratteristica $V_{dropout}$ vs I_{load} del regolatore ADP163 [20]

L'ADP163 è inoltre regolabile, nel senso che permette di impostare una precisa tensione in uscita, selezionabile attraverso l'uso di un partitore resistivo (Figura 3.5), il cui valore di tensione risultante va in ingresso ad un apposito

3.3. ALIMENTAZIONE

pin del regolatore detto ADJ. Ciò permette di poter sfruttare tensioni in uscita che un convertitore LDO ad uscita fissa non può in genere raggiungere.

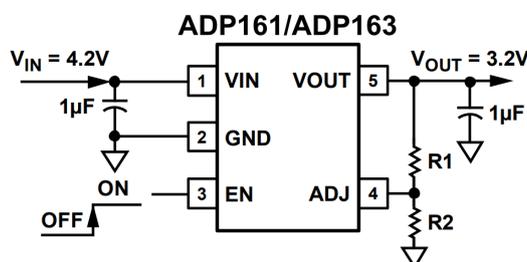


Figura 3.5: ADP163, schematico esemplificativo [20]

Il valore delle resistenze R1 ed R2 che compongono il partitore resistivo, può essere calcolato, tramite un'equazione fornita nel datasheet, a partire dal valore di tensione in uscita desiderato.

Secondo tale equazione:

$$V_{\text{out}} = 1.0V \left(1 + \frac{R1}{R2}\right) + (I_{\text{ADJ-BIAS}})(R1)$$

Dove $I_{\text{ADJ-BIAS}}$ è la corrente che fluisce verso il pin ADJ ed ha un valore tipico di 10 nA alla temperatura ambientale di 25°C.

I valori di resistenza scelti per R1 ed R2 sono rispettivamente di 732 kΩ e 270 kΩ, in modo da ottenere una tensione in uscita di 3.71 V. Ciò ci permette di allungare teoricamente la lifetime del nodo, in quanto il regolatore riuscirà a fornire la giusta tensione in uscita fino a quando la tensione di batteria non sarà scesa sotto al valore di circa 3.90 V a causa dello scaricarsi delle pile.

Infine l'ADP163 è dotato di un pin di enable (EN) attraverso il quale può essere abilitato e disabilitato, in modo da poter disattivare l'uscita nel momento in cui alcun sensore deve essere alimentato, e risparmiare quindi potenza che altrimenti sarebbe inutilmente sprecata.

Per quanto riguarda l'alimentazione del microcontrollore invece, si è scelto di non fare affidamento sul regolatore di tensione incluso nella discovery, ma di erogare la tensione di cui il microcontrollore ha bisogno (3.3 V) tramite un secondo regolatore LDO esterno. La scelta è ricaduta su un convertitore LDO a bassa corrente di riposo e ad uscita fissa a 3.3 V. In particolare si è scelto di utilizzare un STLQ015 [21] di STMicroelectronics che garantisce una Quiescent Current di 1 µA quando nessun carico è servito, contro i 20 µA

del regolatore incluso nella discovery (LD39050 [22] di STMicroelectronics) (Figura 3.6 e Figura 3.7). Per far sì che il regolatore interno fosse by-passato è stato necessario disconnettere la resistenza SB6 della board B-L072Z-LRWAN1 e connettere l'uscita del regolatore esterno direttamente al pin "3V3" del microcontrollore.

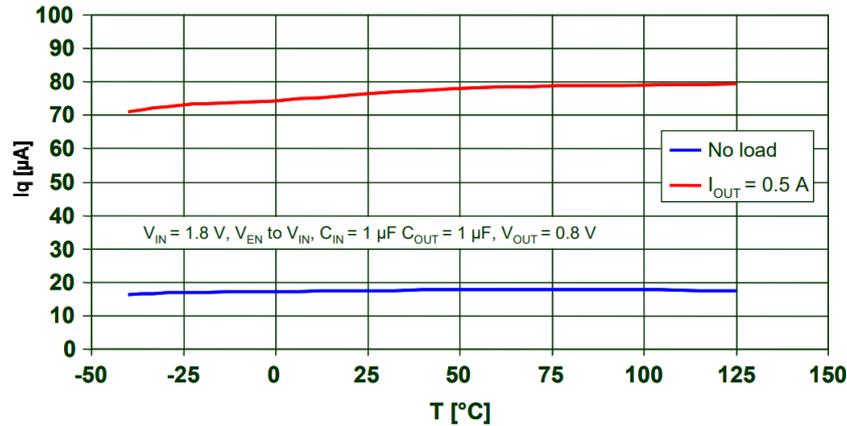


Figura 3.6: Caratteristica IQ vs Temperature del regolatore LD39050 [22]

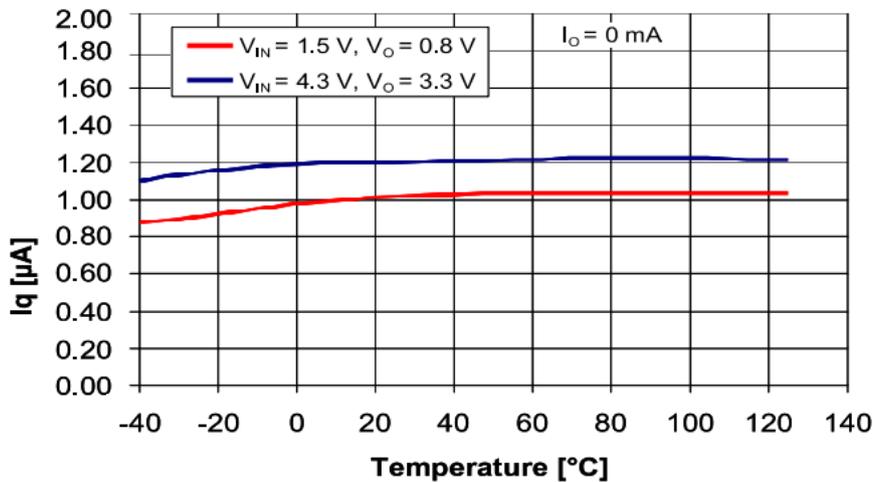


Figura 3.7: Caratteristica IQ vs Temperature del regolatore STLQ015 [21]

Anche regolatore LDO STLQ015 prevede un pin di enable attraverso il quale l'uscita può essere abilitata o disabilitata, tuttavia, dato che risulterebbe insensato disabilitare l'uscita e quindi l'alimentazione del microcontrollore, il pin EN è stato in questo caso cortocircuitato con la tensione di batteria, in

3.4. RILEVAZIONE DELLA PERCENTUALE DI CARICA DELLE BATTERIE

modo che l'uscita risulti costantemente abilitata. Il nodo multi-sensore è stato inoltre dotato di uno switch a scorrimento, il cui funzionamento meccanico permette di scollegare la tensione di batteria, fornita dalle pile, dal nodo e quindi spegnere completamente quest'ultimo.

3.4 Rilevazione della percentuale di carica delle batterie

Il nodo multi-sensore è in grado di fornire un'indicazione riguardante lo stato di carica delle batterie. Per far ciò si è ricorso all'uso di un ADC interno al microcontrollore, attraverso il quale è possibile leggere la tensione fornita dalle batterie, e risalire, attraverso un'apposita equazione di calibrazione, alla percentuale di carica ancora erogabile dalle pile.

L'ADC a 12 bit della board permette di leggere valori di tensione compresi tra 0 V e VDD, dove VDD rappresenta la tensione di alimentazione del microcontrollore (3.3 V, fornita dal regolatore precedentemente descritto).

Poiché la tensione fornita dalle pile (4.5 V nominali) va ben oltre la dinamica dell'ADC, è stato necessario partizionare la loro tensione attraverso un partitore resistivo. Il partitore però limita la corrente in ingresso all'ADC e, di conseguenza, la capacità interna dell'ADC potrebbe non caricarsi completamente, comportando una lettura di tensione sbagliata da parte dell'ADC. Per ovviare a questo problema è stata inserita una capacità di holding esterna in parallelo al partitore resistivo. La corrente circolante sul partitore caricherà quindi suddetta capacità. Una volta completata la carica verrà attivata la lettura della tensione tramite ADC e sarà proprio la capacità aggiunta a fornire l'energia necessaria alla capacità interna dell'ADC per la corretta lettura della tensione. Si è scelto quindi di utilizzare due resistenze di valore pari a $675\text{ K}\Omega$, in modo da avere in ingresso all'ADC una tensione massima di 2.25 V, evitando così il rischio di saturazione della dinamica dell'ADC. Il valore delle resistenze è stato scelto appositamente grande, in modo da abbassare la corrente che fluisce su di esso e che viene quindi persa, ma non dell'ordine del $M\Omega$ altrimenti la capacità avrebbe fatto fatica a caricarsi completamente. Per tali valori di resistenza si ha pertanto un flusso di $3.33\text{ }\mu\text{A}$ quando la tensione di batteria è a 4.5V.

Il valore scelto per la capacità è di $0.1 \mu F$ ed è stato ottenuto sperimentalmente, cercando di raggiungere un giusto compromesso tra velocità della lettura (250 ms) ed accuratezza.

Infine, per evitare che la corrente precedentemente calcolata continui a fluire anche quando alcuna misurazione della carica è in atto, è stato aggiunto a monte del partitore uno switch TS5A3359 (precedentemente descritto), la cui funzione è quella di collegare la tensione di batteria al partitore quando una nuova stima della carica deve essere effettuata, e scollegarla quando l'ADC non è in funzione.

3.5 PCB

Per la creazione del PCB è stato utilizzato il software EDA Kicad [23], distribuito con licenza open-source.

Kicad mette a disposizione un ambiente per la progettazione dello schematico ed uno per la creazione del layout. Per quanto riguarda il primo, si è cercato di interfacciare tutte le parti del circuito precedentemente descritte. A partire dallo schematico è stato poi generato il layout. Si è cercato di implementare quest'ultimo in modo da rendere l'utilizzo del modulo il più agevole possibile, tenendo presente inoltre anche i futuri sviluppi che potrebbero essere apportati al nodo multi-sensore. Ad esempio si è cercato di rendere il nodo compatto, affinché possa essere successivamente dotato di una protezione in grado di resistere alle intemperie ed alla polvere, secondo gli standard di protezione IEC 60529. Per raggiungere tale obiettivo si è cercato di progettare il PCB che ospita la circuiteria di controllo e gestione dei sensori e le batterie, in maniera tale che si potesse agganciare direttamente ai connettori di cui la board dispone (evidenziati in rosso in [Figura 3.8](#)).

Verranno di seguito illustrate i macro-blocchi che compongono il circuito, sia per quanto riguarda lo schematico che il layout.

3.5.1 Schematico

Lo schematico del nodo multi-sensore è mostrato in figura [Figura 3.9](#).

In [Figura 3.10](#) si possono notare i due regolatori LDO per la gestione delle alimentazioni dei sensori e del microcontrollore. Si noti la presenza dello switch

3.5. PCB

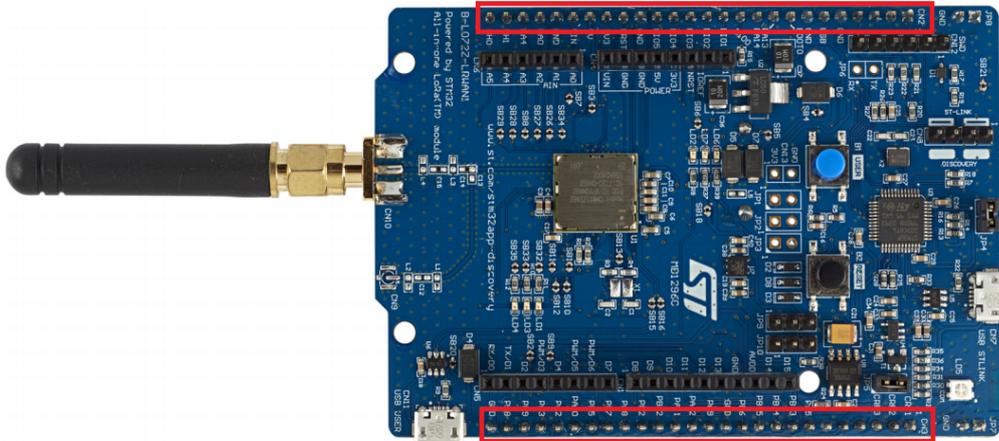


Figura 3.8: B-L072Z-LRWAN1 LoRa®/Sigfox™ Discovery kit [16]

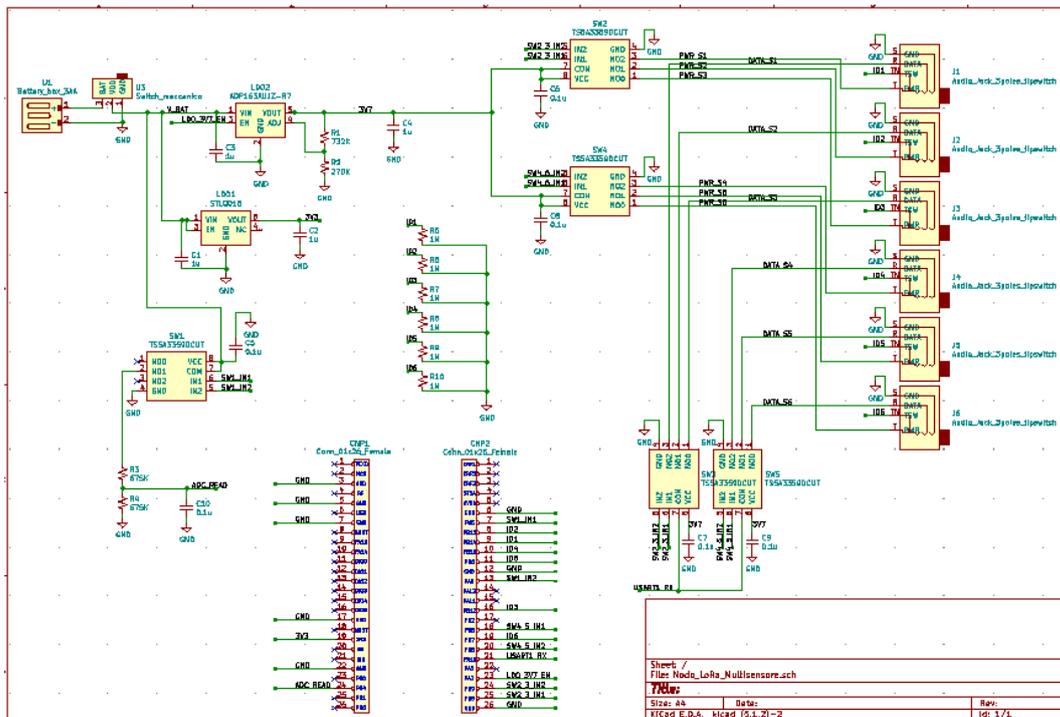


Figura 3.9: Schematico del nodo multi-sensore

a scorrimento e la presenza del vano batterie per l'inserimento delle pile. Le capacità C1, C2, C3 e C4 sono state aggiunte per limitare i disturbi e il loro valore è stato scelto in riferimento a quello consigliato sul datasheet.

In Figura 3.11 è mostrato il blocco di gestione e controllo dell'alimentazione

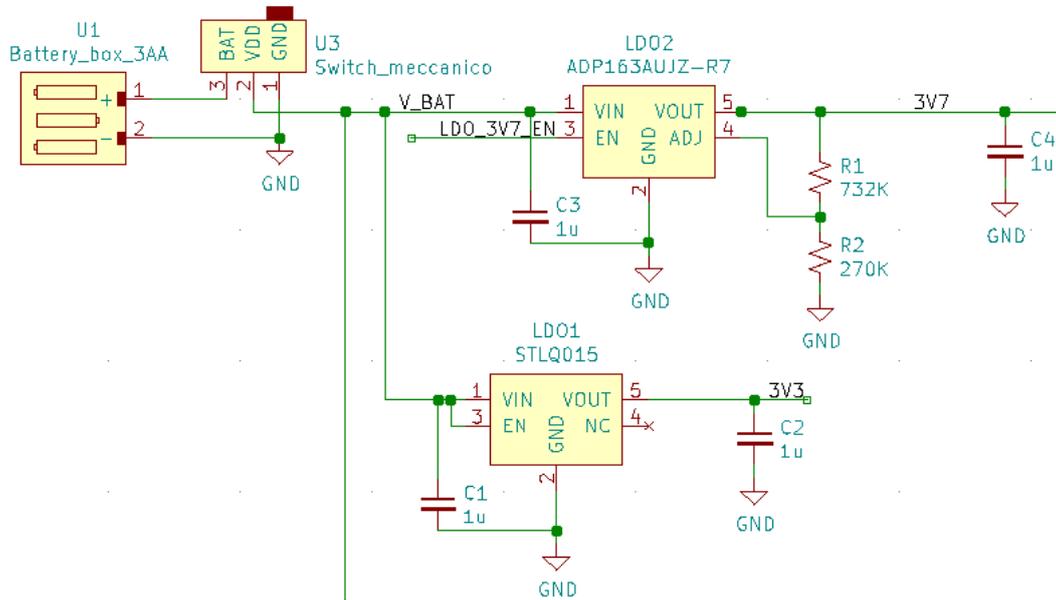


Figura 3.10: Schematico - Blocco di alimentazione

fornita ai sensori. Anche in questo caso sono state aggiunte le capacità (C6 e C8) di disaccoppiamento. Le uscite degli switch sono direttamente connesse alle prese di tipo Jack che ospiteranno i sensori, le quali verranno alimentate durante il funzionamento normale qualora il meccanismo di Insertion Detection abbia rilevato la presenza del sensore connesso.

Si passa quindi alla gestione della comunicazione tramite UART, in [Figura 3.12](#), in cui si può notare la presenza degli switch, il cui funzionamento è quello di selezionare l'uscita del sensore comunicante e connetterla al pin di ricezione dell'UART.

Per quanto riguarda la lettura del livello di carica della batteria, mostrata in [Figura 3.13](#), si notino le resistenze R3 ed R4 del partitore e la capacità di holding C10.

Infine, per la connessione al microcontrollore, si è dotato il PCB di due file di connettori di tipo header femmina ([Figura 3.14](#)) in maniera tale da poterlo collegare direttamente ai pin della board e permetterne un saldo ancoraggio.

3.5. PCB

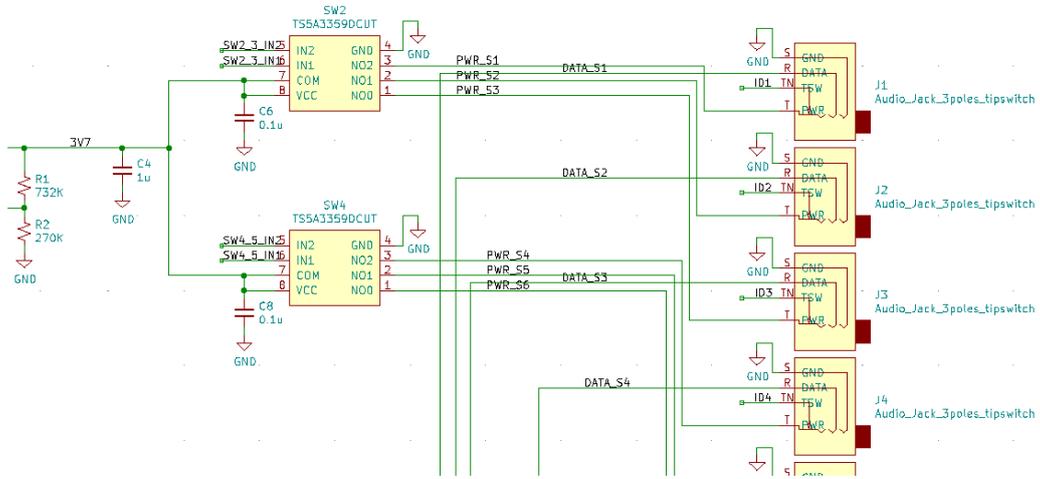


Figura 3.11: Schematico - Distribuzione e gestione dell'alimentazione dei sensori

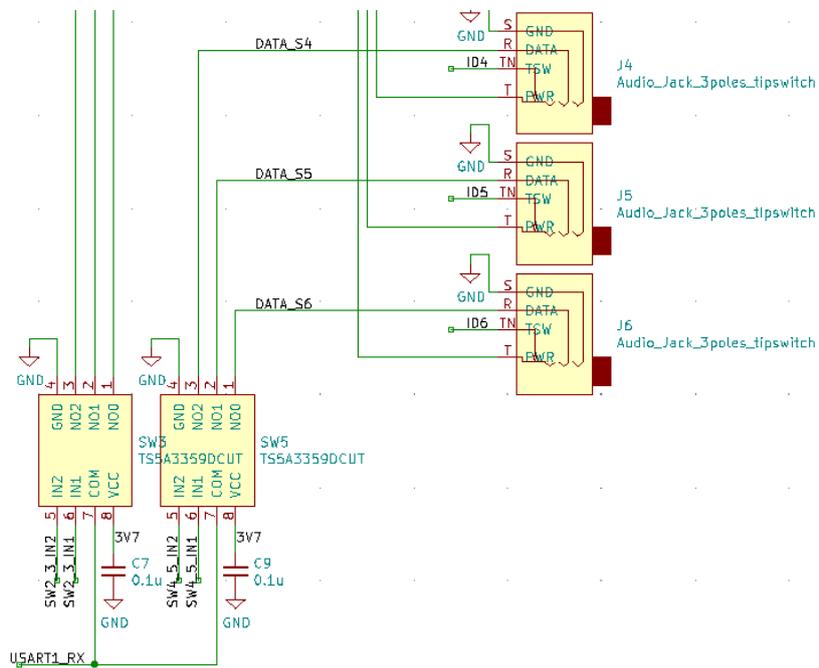


Figura 3.12: Schematico - Gestione della comunicazione tramite UART

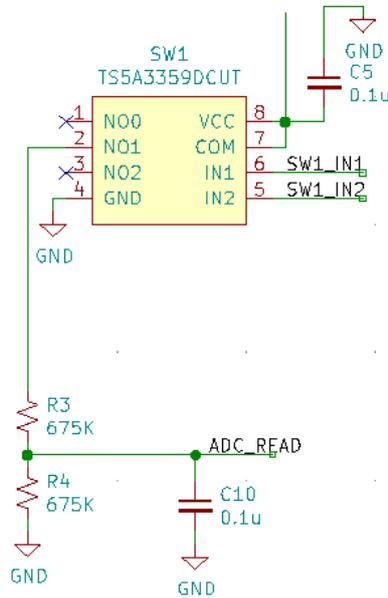


Figura 3.13: Schematico - Gestione della lettura del livello di carica della batteria

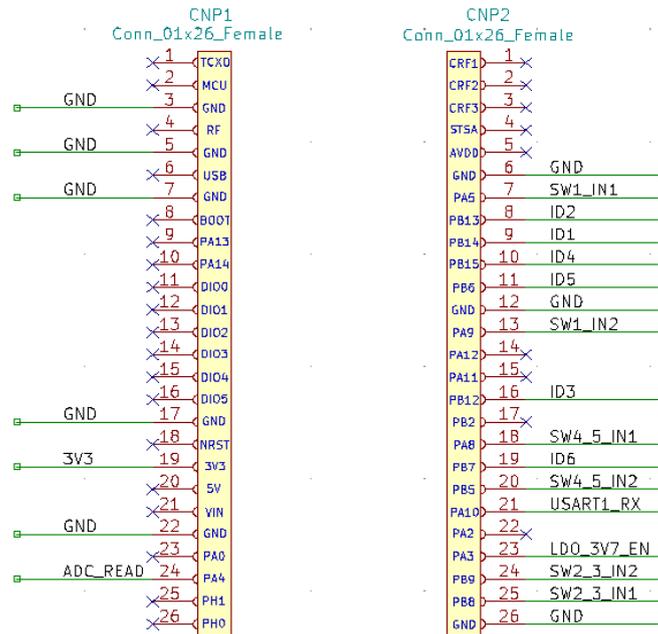


Figura 3.14: Schematico - Connettori per ancoraggio al microcontrollore

3.5.2 Layout

Ultimato lo schematico si è passati alla creazione del layout, in cui si è cercato di disporre i componenti del circuito in maniera più comoda possibile per

3.5. PCB

l'utente utilizzando soltanto due layer, corrispondenti alla faccia superiore ed inferiore del circuito stampato. Il vano per le batterie è stato riposto quindi nel piano superiore del PCB, insieme all'interruttore a scorrimento e alle prese Jack, posizione tutte sullo stesso lato del nodo. Nel piano inferiore del PCB invece sono stati posizionati tutti componenti per la gestione del nodo, tra cui i regolatori LDO per l'alimentazione, gli switch, e le relative resistenze e capacità. In [Figura 3.15](#) è possibile osservare entrambi i layer del PCB. Si noti che il verde rappresenta il livello inferiore mentre il rosso quello superiore. Nella stessa figura non è mostrato il piano di massa aggiunto in modo da renderne più chiara la comprensione.

In [Figura 3.16](#) e in [Figura 3.17](#) è mostrata la rappresentazione 3D del PCB.

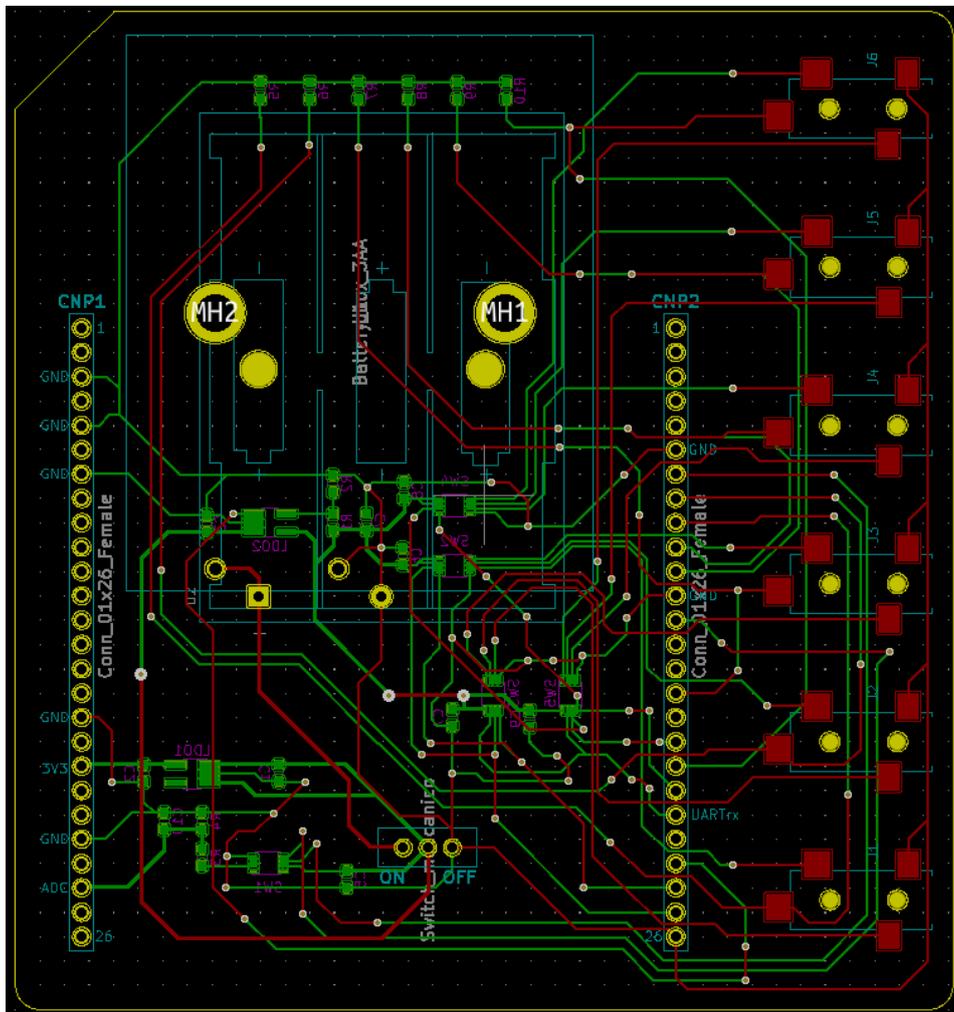


Figura 3.15: PCB del nodo multi-sensore

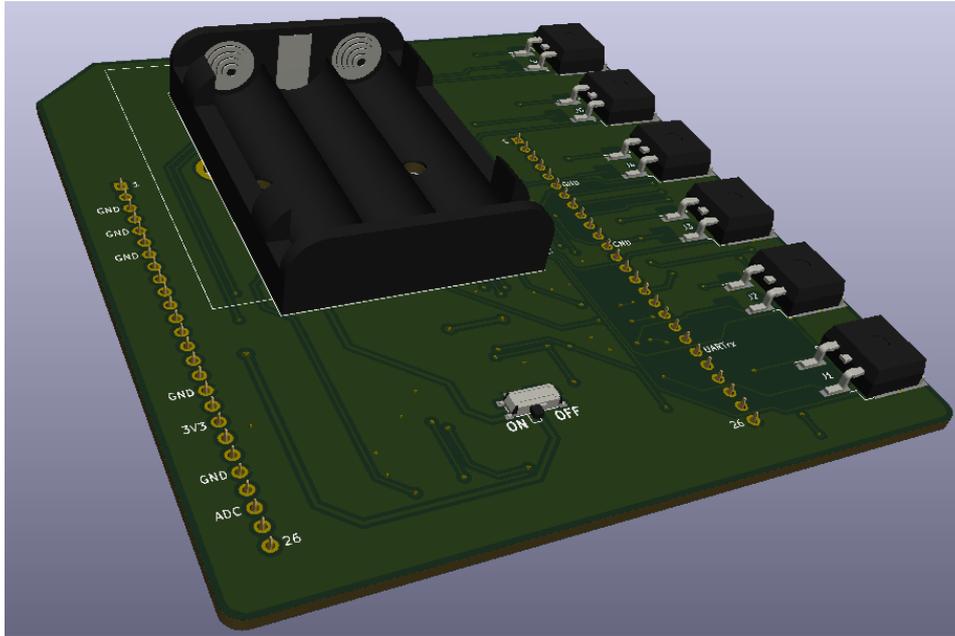


Figura 3.16: Rappresentazione 3D del PCB, livello superiore

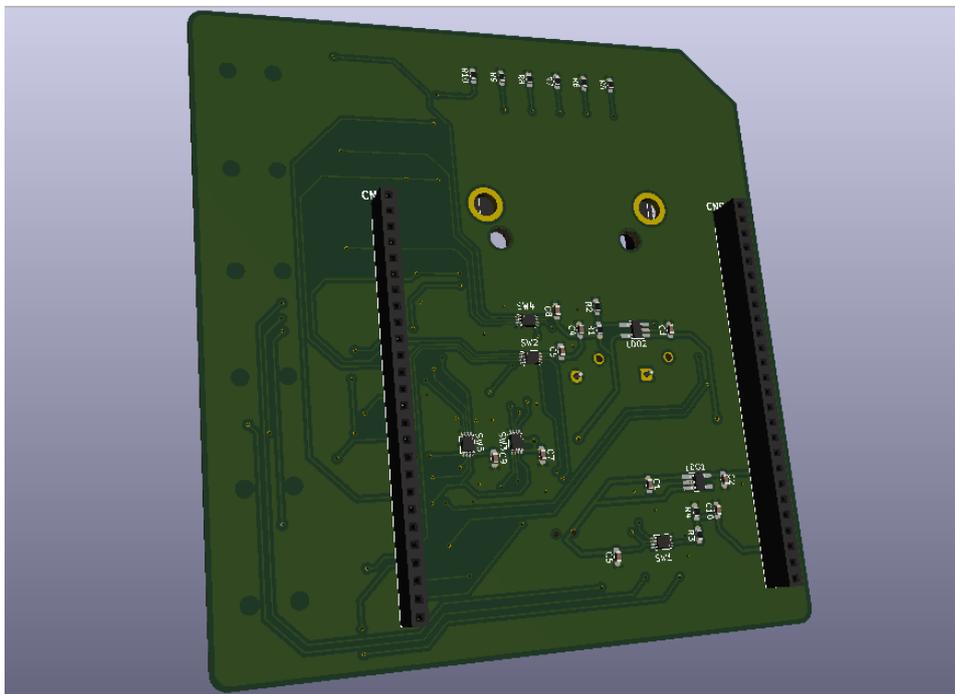


Figura 3.17: Rappresentazione 3D del PCB, livello inferiore

Capitolo 4

Firmware

Il firmware installato sul nodo multi-sensore è basato su quello distribuito dal produttore del microcontrollore, che è stato quindi riconfigurato e customizzato, in modo da inserire in esso tutte le funzioni utili al controllo e alla gestione dei dispositivi precedentemente descritti. In questo capitolo saranno illustrati tutti gli algoritmi e le funzioni che sono stati aggiunti al firmware originale, che prevedeva le funzioni per l'associazione del nodo alla rete LoRaWAN e per la gestione della comunicazione tra il device e l'application server.

Prima di inoltrarsi nella trattazione è bene però specificare alcune accortezze che sono state prese durante la fase di scrittura e di test del firmware. Anzitutto, si è scelto di configurare il nodo multi-sensore in modalità ABP (Activation By Personalization) per quanto riguarda il protocollo LoRaWAN, esclusivamente per rendere la fase di test più semplice e lineare e non preoccuparsi di eventuali problemi che sarebbero potuti incorrere durante il processo di associazione del nodo al network LoRaWAN. Una volta completata la fase di progettazione e di test, l'associazione in modalità OTAA è sicuramente lo strumento di attivazione migliore in quanto automatizza e facilita l'associazione di più nodi contemporaneamente.

La modalità ABP prevede quindi l'inserimento manuale degli identificativi e delle chiavi di sicurezza, che sono state impostate come segue.

```
..\Projects\Multi\Applications\LoRa\End_Node\inc\commissioning.h  
  
/*  
 * When set to 1 the application uses the OTAA procedure
```

```

* When set to 0 the application uses the ABP procedure
*/
#define OVER_THE_AIR_ACTIVATION    0

#define LORAWAN_DEVICE_EUI
{ 0x32, 0xEC, 0x31, 0x2F, 0xC3, 0x6E, 0xA7, 0x00 }

#define LORAWAN_APPLICATION_EUI
{ 0x5B, 0xA8, 0x01, 0xD0, 0x7E, 0xD5, 0xB3, 0x70 }

#define LORAWAN_APPLICATION_KEY
{ 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF }

#if ( OVER_THE_AIR_ACTIVATION == 0 )

#define LORAWAN_NETWORK_ID ( uint32_t )0

#define LORAWAN_DEVICE_ADDRESS ( uint32_t )0x260117D9

#define LORAWAN_NWKSKEY
{ 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF }

#define LORAWAN_APPSKEY
{ 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF }

```

Il valore di Spreading Factor scelto per la modulazione LoRa è stato il più basso (7), in modo che il tempo di attesa minimo tra l'invio di due frame successivi fosse il minore possibile, pur rispettando le limitazioni sul duty-cycle e sull'occupazione della banda imposte dalla normativa Europea ERC-REC-70-3E.

```

..\Projects\Multi\Applications\LoRa\End_Node\src\main.c

#define LORAWAN_DEFAULT_DATA_RATE DR_5 /* SF7 */
#define APP_TX_DUTYCYCLE                30000 /* ms */

```

L'algoritmo di funzionamento previsto per il nodo multi-sensore consta di due fasi principali: una fase di start-up, e una di funzionamento a regime.

4.1 Start-up

Durante la fase di start-up vengono eseguite tutte le operazioni che permettono al nodo di inicializzarsi e riconfigurarsi, a seconda dell'effettivo numero di sensori collegati alla board. Tali operazioni possono essere raggruppate in quattro diverse categorie:

- 1) Inizializzazione
- 2) Insertion Detection
- 3) Riconfigurazione
- 4) Associazione alla rete LoRaWAN

4.1.1 Inizializzazione

Durante l'inizializzazione vengono eseguite tutte le operazioni per la configurazione dei dispositivi, tra cui il modulo radio, l'UART, l'ADC e tutti i GPIO.

```
..\Projects\Multi\Applications\LoRa\End_Node\src\main.c

/* STM32 HAL library initialization*/
HAL_Init( );

/* Configure the system clock*/
SystemClock_Config( );

/*abilito i GPIO per i sensori, per l'ADC e per l'UART*/
GPIO_Configurazione ( );

/* Configure the debug mode*/
DBG_Init( );

/* Configure the hardware*/
HW_Init( );

/*abilito l'UART*/
USART1_Config();
```

Mentre la configurazione del modulo radio fa parte del pacchetto di funzioni di base contenuto nel firmware originale, UART, ADC e GPIO sono stati

4.1. START-UP

ricongigurati ed inizializzati per soddisfare le modalità di funzionamento volute. Per quanto riguarda l'UART ad esempio, era richiesta, per l'interfacciamento al sensore, una baud-rate di 1200 simboli/s, e la presenza di uno Start Bit e di uno Stop Bit.

```
..\B-L072Z-IRWANI\mlm32107x01\Projects\End_Node\translate.c

void USART1_Config(void)
{
    /* Peripheral clock enable */
    __HAL_RCC_USART1_CLK_ENABLE();

    /* Initialization of USART1 */
    UartHandle1.Instance      = USART1;
    UartHandle1.Init.BaudRate  = 1200;
    UartHandle1.Init.WordLength = UART_WORDLENGTH_8B;
    UartHandle1.Init.StopBits  = UART_STOPBITS_1;
    UartHandle1.Init.Parity    = UART_PARITY_NONE;
    UartHandle1.Init.HwFlowCtl  = UART_HWCONTROL_NONE;
    UartHandle1.Init.Mode      = UART_MODE_RX;

    if(HAL_UART_Init(&UartHandle1) != HAL_OK)
    {
        /* Initialization Error */
        Error_Handler();
    }
}
```

L'ADC è invece stato configurato in modo da acquisire una singola lettura a 12 bit attraverso il canale 4.

Per quanto riguarda invece i GPIO, si è scelta una diversa configurazione per ognuno, a seconda della funzione che è stata loro assegnata; ad esempio i GPIO per la logica di controllo degli switch sono stati configurati come Digital Output, mentre quelli adibiti al meccanismo di Insertion Detection sono stati configurati come Digital Input. Infine i GPIO per la lettura tramite ADC e per la ricezione dei dati tramite UART sono stati configurati come Alternate Function in modo che venissero gestiti direttamente dalle relative periferiche.

4.1.2 Insertion Detection

Una volta finita la procedura di inizializzazione, la prima azione eseguita dal nodo è la verifica della presenza o meno dei sei sensori, attraverso il meccanismo descritto nel [Capitolo 3](#).

L'alimentazione dei sensori viene abilitata, accendendo l'apposito regolatore di tensione, e attraverso gli switch viene distribuita a tutti e sei i sensori in successione. Il pin denominato "tip switch" viene quindi letto dal microcoltrollore e, se il valore letto corrisponde ad un 1 logico (tensione alta); questo significa che nel connettore non è inserito alcun sensore e il tip switch è cortocircuitato con l'alimentazione (3.7 V). Qualora invece il valore letto dal microcoltrollore fosse uno 0 logico la presenza del sensore sarebbe confermata, poiché il tip switch risulterebbe sconnesso dall'alimentazione grazie alla presenza del connettore inserito, ed sarebbe quindi cortocircuitato a massa dalla resistenza di pull-down.

```
..\B-L072Z-LRWAN1\mlm32l07x01\Projects\End_Node\translate.c

int Sensors_Detection (int sensor_number){
int sensor_plugged_in=0;
int Sen_Det=0; //sensor detected

/*abilito LDO*/
LDO_Enable();

if(sensor_number==1){
    /*alimento il primo connettore*/
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, GPIO_PIN_RESET);
    HAL_Delay(10);
    /*leggo il pin INSERT DETECTION*/
    Sen_Det=HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_14);
    if(Sen_Det==0){
        sensor_plugged_in=1;
    }
    else{
        sensor_plugged_in=0;
    }
}
...
}
```

4.1.3 Riconfigurazione

Verificata la presenza dei sensori, per ognuno di essi una variabile viene settata a 0, se il sensore è assente, o a 1 se il sensore è presente. Da queste variabili dipende quindi il funzionamento a regime del nodo multi-sensore che da ora in poi, durante la procedura di misurazione, fornirà l'alimentazione solo ai connettori di cui si è verificato l'avvenuto inserimento del sensore. Tutti gli altri connettori vengono tenuti spenti in primo luogo per non sprecare potenza, in secondo luogo in modo da permettere la costruzione di un frame la cui lunghezza varia dipendentemente dal numero di sensori inseriti, come vedremo nella sezione successiva.

```
..\Projects\Multi\Applications\LoRa\End_Node\src\main.c

    /*
    *Controllo se tutti i sensori sono collegati
    * Sensorx_connected=1 => sensore inserito
    * Sensorx_connected=0 => sensore non inserito
    */

    Sensor1_connected = Sensors_Detection (1);
    Sensor2_connected = Sensors_Detection (2);
    Sensor3_connected = Sensors_Detection (3);
    Sensor4_connected = Sensors_Detection (4);
    Sensor5_connected = Sensors_Detection (5);
    Sensor6_connected = Sensors_Detection (6);
```

4.1.4 Associazione alla rete LoRaWAN

Per la procedura di "Attivazione" sono state sfruttate le funzioni di libreria fornite dal produttore, attraverso le quali viene innanzitutto configurato il nodo per il funzionamento il Classe A; quindi viene chiamata la procedura per l'associazione al network.

```
..\Projects\Multi\Applications\LoRa\End_Node\src\main.c

    /* Configure the Lora Stack */
    LORA_Init( &LoRaMainCallbacks , &LoRaParamInit );

    LORA_Join( );
```

Una volta che l'associazione al nodo è stata eseguita, viene completato un intero ciclo di misurazione dei parametri monitorati dai sensori e l'invio di questi ultimi attraverso il protocollo LoRaWAN. Dopodiché viene settato un timer e abilitata la Stop mode.

4.2 Funzionamento a regime

Finita la procedura di Start-up, il sistema è correttamente configurato e pronto per eseguire il funzionamento a regime, che include tutte le operazioni per cui il nodo è stato effettivamente progettato. Le azioni eseguite in questa fase sono le seguenti:

- 1) Attivazione della Run mode
- 2) Accensione dei sensori
- 3) Lettura e controllo del dato ricevuto
- 4) Misura della percentuale di carica
- 5) Invio del frame LoRaWAN
- 6) Riconfigurazione del TX_TIMER
- 7) Attivazione della Stop mode

4.2.1 Attivazione della Run mode e accensione dei sensori

Allo scadere del timer settato viene generato un interrupt dal RTC, ricevuto il quale, il microcontrollore esegue la procedura di uscita dalla Stop mode. Dopodiché inizia il processo vero e proprio di controllo e gestione dei sensori. In successione, viene fornita l'alimentazione ai sensori presenti, quindi si aspetta che il dato inviato da ogni sensore venga correttamente ricevuto tramite UART.

La ricezione attraverso UART viene gestita a polling.

```
..\B-L072Z-LRWAN1\mlm32l07x01\Projects\End_Node\translate.c  
void Acquisizione_dati_sensori(uint8_t* valori, uint8_t sensor_n){
```

```

uint8_t BufferRx1[20]={0};

if(sensor_number==1){
    /*accendo LD0*/
    LDO_Enable();

    /*
    *fornisco l'alimentazione al primo sensore settando i
    *pin di controllo dello switch
    */
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, GPIO_PIN_RESET);
    HAL_UART_ReceiveEx(&UartHandle1, BufferRx1, 25, 500);

    /*
    *disattivo l'alimentazione del sensore settando lo
    *switch nella configurazione di OFF (IN1=IN2=L)
    */
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_8, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_9, GPIO_PIN_RESET);

    /*spengo LD0*/
    LDO_Disable();

    /* traduco il buffer ricevuto in valori decimali */
    Buffer_Translate(BufferRx1, valori);
}
...
}

```

4.2.2 Lettura e controllo del dato ricevuto

Il frame ricevuto attraverso il protocollo seriale viene quindi spaccettato, e vengono estratti i valori dei tre parametri, che sono successivamente tradotti in interi decimali attraverso la funzione `Buffer_Translate`. In particolare, ad ogni parametro sono associate due cifre: una per la parte intera ed una per la parte decimale.

```

..\B-L072Z-LRWAN1\mlm32107x01\Projects\End_Node\translate.c

void Buffer_Translate(uint8_t *pData, uint8_t *values){

```

```
    int k=0, d=0, i=0;
    indexArray=2;
    curs=0;
    float t=0;
    float decTempI=0;

/*
 *separo i valori di permittivita', temperatura e conducibilita'
 *in 6 diversi array, 3 interi e 3 decimali
 *termino ogni numero (parte intera o decimale) con una 0x7E in
 *modo da riconoscerne poi la lunghezza
 *
 *pData e' il buffer intero che contiene i valori di
 *permittivita' conducibilita' e temperatura (e altro)
 *parteintera e' l'array in cui copiamo la parte intera di uno
 *dei 3 valori partedecimale e' l'array in cui copiamo la parte
 *decimale (vuoto se non c'e' parte decimale)
 */

    unzip(pData, permI, permD);

    ...

/*
 *conto il numero di elementi di ogni array (incremento una
 *variabile fino a quando non leggo 0x7E (carattere terminatore
 *tilde) e traduco il valore in esadecimale con la
 *funzione ExaToDec
 */
    i=0;
    d=0;
    while(permI[i]!=0x7E){
        i++;
    }
    while(permD[d]!=0x7E){
        d++;
    }
    values[0]=ExaToDec(permI, i, 0);
    values[1]=ExaToDec(permD, d, 1);

    ...
}
```

4.2. FUNZIONAMENTO A REGIME

La funzione `Buffer_Translate` provvede anche a calcolare il checksum del frame ricevuto e confrontarlo con quello previsto, attraverso l'algoritmo fornito da METER. Nel caso in cui non si abbia corrispondenza tra i due checksum un messaggio di errore viene generato ed inviato attraverso LoRaWAN.

```
..\B-L072Z-LRWAN1\mlm32l07x01\Projects\End_Node\translate.c

uint8_t check(uint8_t* pData){
    uint8_t checkOk=0;
    uint16_t length=25;
    uint16_t sum = 0;

    for( i = 0; i < length; i++ )
    {
        sum += pData[i];
        if(pData[i] == '\r')
        {
            /*
             *Found the beginning of the meta data
             *section of the response
             */
            break;
        }
    }
    /* include the sensor type into the checksum */
    i++;
    sum += mirrored[i];

    /* Convert checksum to a printable character */
    sum = sum % 64 + 32;
    i++;
    if(sum==mirrored[i]){
        checkOk=1;
    }
    return checkOk;
}
```

4.2.3 Misura della percentuale di carica e invio del frame LoRaWAN

Una volta che tutti i dati sono stati acquisiti e opportunamente elaborati, sono incapsulati in un unico frame, contenente anche il livello di batteria misurato. Quindi il frame è inviato attraverso LoRaWAN.

```
..\ Projects\Multi\Applications\LoRa\End_Node\src\main.c

static void Send( void ){
...
    /*lettura livello batteria*/
    batteryLevel = GetBatteryVoltage();
    AppData.Buff[i++] = 0;

    if(Sensor1_connected==1){
        /*lettura primo sensore*/
        Acquisizione_dati_sensori(valori1,1);

        permissivitI=valori1[0];
        permissivitD=valori1[1];
        temperaturI=valori1[2];
        temperaturD=valori1[3];
        conducibilitI=valori1[4];
        conducibilitD=valori1[5];

        AppData.Buff[i++] = permissivitI & 0xFF;
        AppData.Buff[i++] = permissivitD & 0xFF;
        AppData.Buff[i++] = temperaturI & 0xFF;
        AppData.Buff[i++] = temperaturD & 0xFF;
        AppData.Buff[i++] = conducibilitI & 0xFF;
        AppData.Buff[i++] = conducibilitD & 0xFF;
    }
...
    AppData.Buff[i++] = batteryLevel;
    AppData.Buff[0] = i;
    AppData.BuffSize = i;

    LORA_send( &AppData, LORAWAN_DEFAULT_CONFIRM_MSG_STATE);
}
```

La funzione per il calcolo del livello di carica della batteria prevede l'attivazione del collegamento tra il partitore resistivo descritto nel [Capitolo 3](#) e la tensione di alimentazione. La corrente che fluisce nel partitore carica quindi la capacità di holding esterna, e non appena questa è totalmente carica viene abilitata la lettura della tensione tramite l'ADC. Per far in modo che la capacità abbia il tempo di caricarsi è stato inserito un ritardo di 250 ms (ottenuto sperimentalmente) prima che la lettura venga compiuta.

Il valore digitale letto viene quindi normalizzato rispetto al riferimento di tensione interno del microcontrollore e convertito in volt, attraverso un'equazione di calibrazione fornita nel Reference Manual. Dopodiché viene convertito in percentuale di carica, attraverso l'algoritmo descritto all'interno della funzione `GetBatteryVoltage` di seguito riportata.

```
..\B-L072Z-LRWAN1\mlm32l07x01\Projects\End_Node\translate.c

int GetBatteryVoltage (void){
    float SupplyVoltage, BatteryVoltage;
    int BatteryLevel;

    /*abilito il partitore*/
    Battery_Measurement_Enable();

    /*
    *aspetto che si carichi la capacita' (200 ms consentono alla
    *capacita' di caricarsi, ma per essere sicuri che sia
    *carica ne ho aggiunti altri 50 ms)
    */
    HAL_Delay(250);

    /*acquisizione tensione di riferimento per normalizzazione*/
    SupplyVoltage=HW_AdcReadChannel(ADC_CHANNEL_VREFINT);

    /*inizio la lettura*/
    BatteryVoltage=HW_AdcReadChannel(ADC_CHANNEL_4);

    /*fine lettura, disabilito il partitore*/
    Battery_Measurement_Disable();

    /*normalizzo e converto il valore letto in volt*/
    SupplyVoltage=VDDA_VREFINT_CAL*VREFINT_CAL/SupplyVoltage;
    BatteryVoltage=(BatteryVoltage/4096)*SupplyVoltage;
```

```
/*
 * tensione max 4500 mV (3 pile da 1500mV)
 * tensione min 3800 mV (piu' in giu' si spegne il regolatore)
 * (4500-3800)/100=7mV corrispondono all'1%
 * BatteryVoltage in realta' e' il valore misurato dopo la
 * caduta sul partitore resistivo.
 * Per risalire al vero valore della batteria basta dividere
 *per R2/(R1+R2)
 * R2/(R1+R2)=0.5 con R2=0.75 MOhm e R1=0.75 MOhm in modo da
 * abbassare la tensione batteria da 4500 mV a 2250 mV
 */
    BatteryLevel=(int)(( BatteryVoltage /0.5) -3800)/7;

    if(BatteryLevel >=100){
        BatteryLevel=100;
    }
    if(BatteryLevel <=0){
        BatteryLevel=0;
    }
    return BatteryLevel;
}
```

4.2.4 Riconfigurazione del timer e attivazione della Stop mode

Una volta che il frame contenente i dati delle misure è stato correttamente inviato viene riconfigurato il TX_TIMER e viene abilitata la Stop mode.

Il sistema entra così in stato di idle, fino a quando, allo scadere del timer, la Run mode verrà nuovamente abilitata e verrà intrapreso un nuovo ciclo di misurazioni.

4.3 Test mode

Oltre alle funzioni fin'ora descritte, sono state aggiunte nel codice alcune direttive per il compilatore che consentono di attivare e disattivare la modalità di test. Quando la test mode è abilitata, è possibile eseguire porzioni del codice in modo indipendente l'una dalle altre. Il tutto è stato implementato

4.3. TEST MODE

per rendere più agevole la verifica del corretto funzionamento dei dispositivi e del firmware.

```
..\Projects\Multi\Applications\LoRa\End_Node\src\main.c

/*
 *Di default la TESTING_MODE disabilita la lettura dei sensori
 *e della batteria
 */
#define TESTING_MODE

/*
 *In TESTING_MODE possiamo scegliere se tenere disabilitate
 *tutte le letture (sensori e batteria), oppure decidere se
 *abilitare la lettura sensori, attraverso la
 *direttiva TEST_GET_SENSOR_VALUE o se abilitare la lettura
 *della batteria con la direttiva TEST_GET_BATTERY_VALUE
 */
//#define TEST_GET_SENSOR_VALUE
//#define TEST_GET_BATTERY_VALUE

/*
 *Definire TEST_ONE_SENSOR solo se TESTING_MODE e
 *TEST_GET_SENSOR_VALUE sono già definiti.
 *In questo modo invece di 3 letture ne verrà fatta solo la prima
 */
//#define TEST_ONE_SENSOR
...
#ifdef TESTING_MODE
    AppData.Buff[i++] = 0xFF;
    ...

    AppData.Buff[i++] = 0xFF;

    AppData.BuffSize = i;
    LORA_send( &AppData, LORAWAN_DEFAULT_CONFIRM_MSG_STATE);
#endif
```

Capitolo 5

Stima dei consumi e dei tempi di vita del nodo multi-sensore

Finita la parte di progettazione sia hardware che software, si è passati alla fase di testing e alla misura dei consumi effettivi del nodo multi-sensore. In questo capitolo verranno analizzati i dati raccolti e si procederà ad eseguire una stima dei tempi di vita del nodo.

5.1 Misura dei consumi

Per eseguire le misure è stata utilizzata la Power Monitoring Board, progettata nel laboratorio ARCES (Advanced Research Center on Electronic Systems "Erode De Castro") [24] in collaborazione con STMicroelectronics. La board è dotata di tre canali indipendenti attraverso i quali è possibile acquisire e misurare valori di corrente e tensione. Ciò ci consente di tenere traccia delle tensioni e delle correnti che istante per istante vengono fornite al nodo multi-sensore dalle batterie.

Si tenga presente che le misure sono state eseguite considerando il caso in cui tre sensori sono collegati al nodo e il caso in cui è collegato un solo sensore. Si è scelto inoltre di ripetere le misure impostando diversi valori dello Spreading Factor (minimo e massimo, rispettivamente 7 e 12), in modo da ottenere una caratterizzazione dei consumi più completa. Infine sono state analizzate le tre diverse fasi di Start-up, funzionamento a regime e Stop mode.

Mentre per quanto riguarda la Stop mode non si ha alcuna differenza in

5.1. MISURA DEI CONSUMI

termini di consumi al variare dei diversi parametri, il funzionamento a regime è influenzato sia dal valore di SF che dal numero di sensori inserito. Idealmente, un aumento del numero di sensori si ripercuote sui consumi in quanto aumenta la potenza ad essi fornita durante le misure. Ad un aumento di SF invece corrispondono tempi ed energia dissipata durante la trasmissione del frame più alti.

Lo stesso discorso vale anche per la fase di Start-up, i cui consumi sono però più alti di quelli riferiti al funzionamento a regime in quanto oltre ad un intero ciclo di misure sono previste anche le fasi di inizializzazione, Insertion Detection e riconfigurazione del nodo.

Si esamini la [Figura 5.1](#), in cui sono mostrati i consumi durante il funzionamento a regime con SF impostato a 12 e tre sensori collegati. Si può notare innanzitutto la fase di uscita dalla Stop mode e la conseguente attivazione dei sensori ed elaborazione dei dati, a cui segue l'invio del frame e, a distanza di un secondo, l'apertura delle due finestre di ricezione descritte nella [sezione 1.2](#)



Figura 5.1: Misura di correnti e tensioni durante un ciclo di funzionamento a regime con SF12 e tre sensori collegati

La corrente media assorbita dal nodo sensore in un intervallo di 30 secondi, contenente un intero ciclo di misure, è di 2.578 mA. Mediando sui campioni

CAPITOLO 5. STIMA DEI CONSUMI E DEI TEMPI DI VITA DEL NODO MULTI-SENSORE

ottenuti durante la Stop mode è invece possibile calcolare la corrente media consumata dal nodo negli intervalli di non operatività. In questa fase si è misurata una corrente media di $4.3 \mu A$.

A partire da tali valori può essere fatta una stima del consumo di corrente medio per ora del nodo multi-sensore. Considerando infatti che il ciclo di lettura dei parametri e invio del frame si ripete una volta per ora, basterà effettuare la media dei valori precedentemente ottenuti, pesata sulla lunghezza degli intervalli di tempo nei quali essi si verificano. In sintesi avremo:

$$I_{\text{avarage-hour}} = \frac{3570[s] * I_{\text{stop}} + 30[s] * I_{\text{avarage-30sec}}}{3600[s]} = 25.79 \mu A$$

Dove 3570 s rappresenta l'intervallo di tempo in cui il nodo risulta in Stop mode, mentre 30 s è l'intervallo di tempo riferito alle misurazioni descritte precedentemente. 3600 s rappresenta il periodo di tempo totale su cui la stima è stata effettuata, vale a dire un'ora

Per quanto riguarda la fase di Start-up, i consumi di corrente misurati confermano quanto precedentemente detto. In tale fase si ha un consumo medio di 2.640 mA: maggiore rispetto a quello misurato durante il funzionamento a regime di $62 \mu A$. Il lieve aumento dei consumi è in linea con quanto previsto. Basti pensare infatti che il dispendio di energie maggiore deriva dall'alimentazione fornita ai tre sensori e dall'invio del frame attraverso la modulazione LoRa, che pertanto rimane tale per entrambe le fasi.

Analizzando i dati derivanti dal calcolo dei consumi quando soltanto un sensore è collegato al nodo ([Figura 5.2](#)), si può risalire non solo alla corrente media assorbita dal nodo in tale configurazione, bensì possono essere stimati i consumi derivanti da un qualsiasi numero di sensori inseriti. Per effettuare tale stima si potrebbe pensare di sottrarre o sommare, alla misura effettuata con tre sensori inseriti, i consumi relativi all'alimentazione del singolo sensore. Tuttavia si incorrerebbe in una stima errata in quanto così facendo non verrebbero considerati né l'allungamento dei tempi previsti per le misurazioni, né la potenza dissipata per trasmettere un frame di lunghezza diversa, contenente più o meno informazioni.

A partire invece dalla misura dei consumi nella configurazione con un solo sensore inserito, e sottraendo quest'ultima a quella effettuata con tre sensori inseriti e quindi dividendo per 2, è possibile stimare il consumo effettivo di potenza derivante dal controllo, l'elaborazione e la trasmissione dei dati riferiti

5.1. MISURA DEI CONSUMI

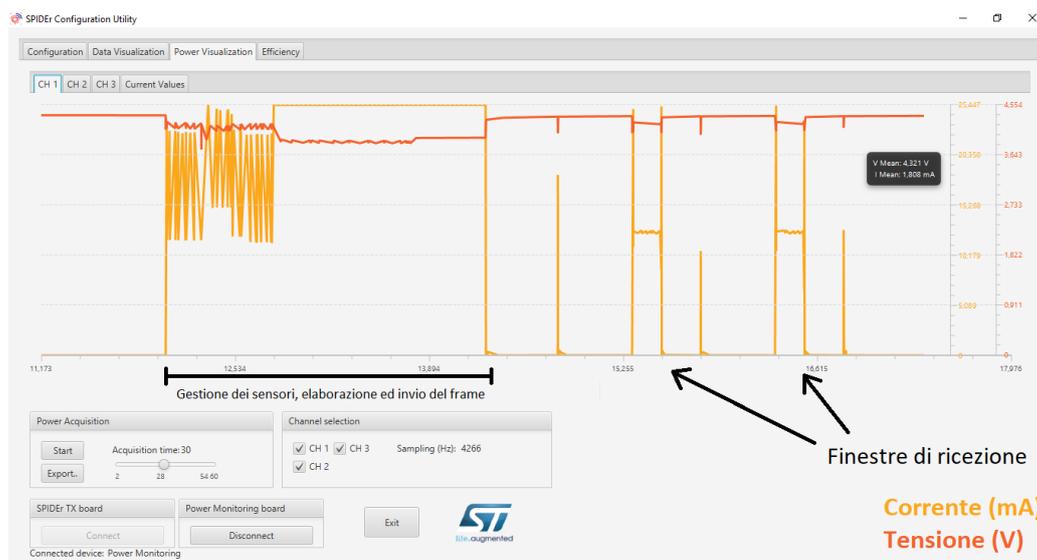


Figura 5.2: Misura di correnti e tensioni durante un ciclo di start-up con SF12 e un sensore collegato

ad un singolo sensore. Solo a quel punto può essere sottratta o aggiunta la nuova stima alla misura precedentemente effettuata, per risalire a tutte le configurazioni possibili. I valori ottenuti sono riassunti in [Tabella 5.1](#).

Si esaminino ora la [Figura 5.3](#) e la [Figura 5.4](#) in cui si tiene traccia dei



Figura 5.3: Misura di correnti e tensioni durante un ciclo di funzionamento a regime con SF7 e tre sensori collegati

CAPITOLO 5. STIMA DEI CONSUMI E DEI TEMPI DI VITA DEL NODO MULTI-SENSORE

valori di corrente e tensione forniti al nodo nella configurazione con SF7 e rispettivamente con tre e un sensore collegati. Applicando la stessa procedura precedentemente descritta per la configurazione con SF12, si è stimato un consumo medio di 1.511 mA calcolato in un intervallo di 30 secondi contenente un intero ciclo di misure. Si è calcolato quindi un consumo medio per ora di $16.63 \mu A$ quando tre sensori sono collegati al nodo.

Come previsto, aumentando il fattore di spreading aumentano i consumi del nodo multi-sensore. Si passa, ricapitolando, da un valore di corrente medio di $16.63 \mu A$, ad uno di $25.79 \mu A$: un aumento di $9.16 \mu A$.

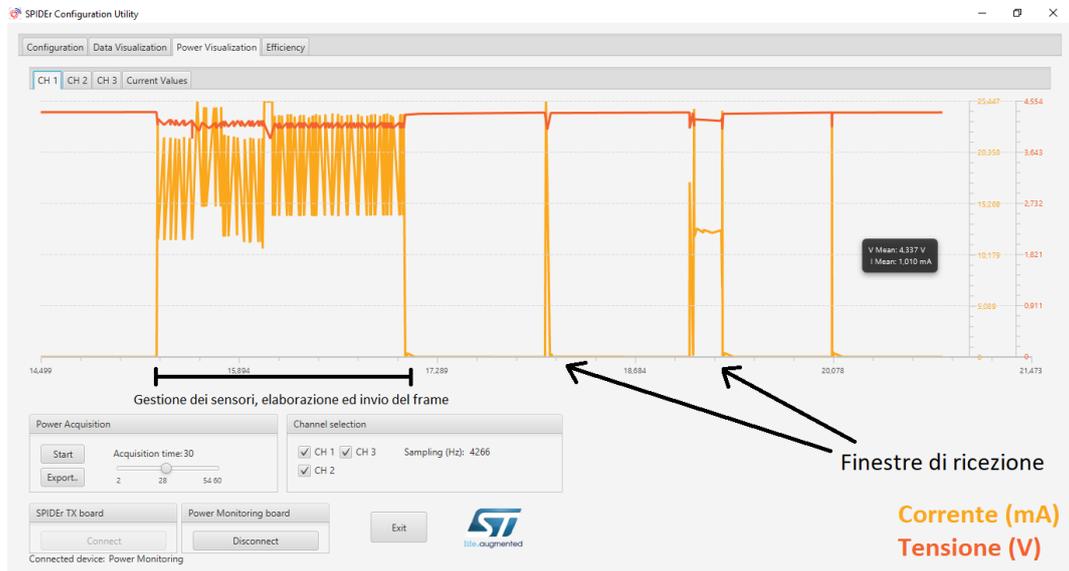


Figura 5.4: Misura di correnti e tensioni durante un ciclo di start-up con SF7 e un sensore collegato

N° di sensori	1	2	3	4	5	6
$I_{\text{average-SF12}}(\mu A)$	19.78	22.79	25.79	28.79	31.80	36.80
$I_{\text{average-SF7}}(\mu A)$	12.70	14.66	16.63	18.59	20.56	24.49

Tabella 5.1: Stima del consumo di corrente medio per numero di sensori inseriti a diversi valori di SF

5.2 Stima della lifetime

A partire dai dati relativi al consumo di corrente possiamo stimare i tempi di vita del nodo multi-sensore. È stato preso in considerazione il caso con Spreading Factor 12 e tre sensori inseriti. Inoltre, per la stima, si è tenuto conto della capacità media che caratterizza delle generiche pile commerciali di tipo AAA Alcaline. Le caratteristiche riferite alle batterie sono infatti standard e dipendono dal formato della batteria e dal processo chimico su cui è basato il loro funzionamento. Le pile considerate hanno una capacità di 1200 mAh e una tensione nominale di 1.5 V. Per determinare il tempo di vita del nodo multi-sensore basterà quindi dividere la capacità delle batterie per i valori medi per ora di corrente calcolati nella [sezione 5.1](#).

Nel caso sopracitato si avrà quindi una lifetime di 46531 ore: pari a 5.38 anni. Tale stima è stata fatta come segue:

$$Lifetime = \frac{1200[mAh]}{I_{average-hour}[mA]} = \frac{1200[mAh]}{0.02579[mA]} = 46531.72[h]$$

Se invece viene considerato il caso con SF7 e tre sensori collegati al nodo, i tempi di vita di allungano in quanto i consumi misurati sono sensibilmente più bassi. In tal caso è stata stimata una lifetime di 72159 ore di utilizzo, pari a 8.35 anni.

Capitolo 6

Esempio di funzionamento, ricezione e lettura dei dati tramite The Things Network

In questo capitolo verrà mostrato un esempio di funzionamento del nodo multi-sensore progettato, e di come questo si interfaccia alla rete LoRaWAN e all'infrastruttura di The Things Network.

6.1 Configurazione della rete LoRaWAN

Prima che il nodo sia effettivamente connesso alla rete e funzionante deve essere configurato.

Una volta creata l'applicazione sulla piattaforma TTN ed impostato l'Application ID, si può procedere con l'associazione del nodo. Durante la procedura vengono generate automaticamente gli identificativi e le chiavi richiesti dal protocollo LoRaWAN, come mostrato in [Figura 6.1](#). Tali parametri devono quindi essere importati nel firmware (come mostrato nel [Capitolo 4](#)) e verranno usati dal dispositivo nel momento in cui una nuova sessione di comunicazione dovrà essere intrapresa.

DEVICE OVERVIEW

Application ID	transmission_example		
Device ID	Nodo multi-sensore	Activation Method	ABP
Device EUI	<> ⇄ 00 A7 6E C3 2F 31 EC 32		📄
Application EUI	<> ⇄ 70 B3 D5 7E D0 01 A8 5B		📄
Device Address	<> ⇄ 26 01 17 D9		📄
Network Session Key	<> ⇄ 👁		📄
App Session Key	<> ⇄ 👁		📄

Figura 6.1: Esempio di associazione di un device all'applicazione sulla piattaforma TTN

6.2 Decoder

Il decoder è un applicativo fornito da The Things Network attraverso il quale è possibile interpretare ed elaborare i dati contenuti nel frame payload, in modo da agevolarne la comprensione. È programmato in linguaggio JavaScript ed è stato progettato in modo da riconoscere il numero di sensori inserito nel nodo. Una volta fatto ciò, il decoder legge i byte ricevuti in sequenza e associa le informazioni ai relativi parametri misurati.

Ad esempio, se il pacchetto ricevuto contiene le informazioni relative ad un solo sensore, lo script si comporterà come segue:

```
function Decoder(bytes) {
  ...
  /*
   *per ogni parametro si hanno a disposizione due byte,
   *uno per la parte intera ed uno per la parte digitale.
   *Quindi la parte digitale viene divisa per
   *100 e sommata alla relativa parte intera
   */
  if(bytes[0]==8){
```

CAPITOLO 6. ESEMPIO DI FUNZIONAMENTO, RICEZIONE E LETTURA DEI DATI TRAMITE THE THINGS NETWORK

```
var Permittivita1 = (bytes [1]);
var PermittivitaD1 = (bytes [2])/100;
var Temperatura1 = (bytes [3]);
var TemperaturaD1 = (bytes [4])/100;
var Conducibilita1 = (bytes [5]);
var ConducibilitaD1 = (bytes [6])/100;
var Bat = bytes [7];

Permittivita1=Permittivita1+PermittivitaD1;
Temperatura1=Temperatura1+TemperaturaD1;
Conducibilita1=Conducibilita1+ConducibilitaD1;

return {
  FS_Permittivity: Permittivita1 ,
  FS_Temperature: Temperatura1 ,
  FS_Conducibility: Conducibilita1 ,
  Battery_Level: Bat
};
}
...
}
```

6.3 Ricezione di un pacchetto LoRaWAN

Conclusa la procedura per l'associazione del nodo all'applicazione, e la configurazione del formato del payload, il nodo multi-sensore è pronto per la comunicazione. I dati contenuti nei pacchetti inviati dal nodo e ricevuti sul Network Server sono quindi decodificati e leggibili attraverso l'utility Console messa a disposizione da The Things Network, come mostrato in [Figura 6.2](#).

Nell'esempio in figura si è scelto di impostare un valore di SF pari a 12 e di collegare un solo sensore al nodo.

Si può notare quindi il payload del pacchetto, e come questo venga interpretato dal decoder per estrapolare i dati misurati dal GS3 (riferiti ai tre parametri sotto osservazione) e il dato riferito al livello di carica della batteria.

6.3. RICEZIONE DI UN PACCHETTO LORAWAN

The screenshot displays the 'APPLICATION DATA' interface of the TTN platform. At the top right, there are controls for 'pause' and 'clear'. Below this, a 'Filters' section contains buttons for 'uplink', 'downlink', 'activation', 'ack', and 'error', with 'uplink' currently selected. The main area shows a table of received packets with columns for 'time', 'counter', and 'port'. Three packets are listed, all from a device with ID 'new_device' and port '2'. The most recent packet is at 12:05:03 with counter 20 and payload '08 02 24 1A 32 01 00 39'. Below the table, the 'Uplink' section is expanded to show the 'Payload' as '08 02 23 1A 3C 01 00 39' and the 'Fields' as a JSON object:

```
{ "Battery_Level": 57, "FS_Conducibility": 1, "FS_Permittivity": 2.35, "FS_Temperature": 26.6 }
```

time	counter	port	dev id	payload	Battery_Level	FS_Conducibility	FS_Permittivity
12:05:03	20	2	new_device	08 02 24 1A 32 01 00 39	57	1	FS_Permittivity
12:02:02	19	2	new_device	08 02 22 1A 3C 01 00 39	57	1	FS_Permittivity
11:59:01	18	2	new_device	08 02 23 1A 3C 01 00 39	57	1	FS_Permittivity

Uplink

Payload

08 02 23 1A 3C 01 00 39

Fields

```
{ "Battery_Level": 57, "FS_Conducibility": 1, "FS_Permittivity": 2.35, "FS_Temperature": 26.6 }
```

Figura 6.2: Esempio di ricezione di un pacchetto tramite la piattaforma TTN

Capitolo 7

Conclusioni

In questa trattazione è stata descritta la progettazione di un nodo multi-sensore LoRaWAN per la stima del fabbisogno idrico in agricoltura.

Il flusso di progetto che ne ha segnato la realizzazione parte da un'iniziale fase di studio dello stato dell'arte, a cui è seguita una fase di approfondimento degli standard in uso e dei sensori che il nodo si propone di integrare.

Assodati i tratti salienti che caratterizzano sia il protocollo di comunicazione, sia i sensori, si è quindi cercato di progettare il nodo multi-sensore, facendo attenzione a rispettare le specifiche dettate sia dai dispositivi utilizzati, sia dalle modalità operative dell'intero nodo.

Parallelamente allo sviluppo hardware del nodo si è proceduto anche all'implementazione lato firmware, i cui aspetti fondamentali sono stati descritti nella trattazione.

In questa fase di progetto ci si è preoccupati di ottimizzare il nodo in termini di fabbisogno energetico, con l'obiettivo di estenderne il tempo di vita. A tal proposito si è intervenuti a tutti i livelli di progettazione: dal firmware alle soluzioni circuitali.

Il dispositivo è stato equipaggiato di tutta la componentistica elettronica necessaria a fornirgli l'alimentazione e ad effettuare il controllo e la gestione dei sensori.

Il corretto funzionamento del sistema è stato testato su breadboard e il dispositivo è stato quindi caratterizzato per quanto riguarda i consumi. I tempi di vita del nodo sono in linea con le aspettative.

La fase di progetto è stata conclusa con il disegno e lo sviluppo del PCB, il quale è quindi pronto per il processo di fabbricazione.

7.1 Sviluppi futuri

Al processo di fabbricazione del PCB seguiranno le fasi di montaggio dei componenti e di test, per verificare la correttezza del circuito stampato.

A partire dal progetto qui descritto, sono inoltre possibili future implementazioni per quanto riguarda le funzionalità del nodo e le sue modalità di utilizzo.

Il dispositivo dovrà essere equipaggiato con una protezione contro le intemperie e la polvere, secondo gli standard IEC 60529.

Il nodo multi-sensore prevede, secondo il progetto attuale, la possibilità di collegare fino a sei sensori contemporaneamente. Il firmware è ora compatibile con la gestione di un unico tipo di sensore, descritto nel [Capitolo 2](#), che dovrà essere posizionato a diverse profondità del terreno. La compatibilità del firmware potrà essere successivamente estesa anche ad altri sensori prodotti da METER per il monitoraggio delle falde acquifere e delle condizioni atmosferiche, in modo da stimare con maggiore precisione il fabbisogno idrico del suolo.

La presente tesi è stata svolta nel centro di ricerca ARCES (Advanced Research Center on Electronic System "Erocole De Castro") nell'ambito del progetto SWAMP, all'interno del programma di finanziamento dell'Unione Europea Horizon 2020, ed è stata motivata dall'esigenza di integrare, in una piattaforma IoT, sensori di monitoraggio del contenuto idrico nel suolo, al fine di trasmettere in tempo reale informazioni necessarie ad un sistema di ottimizzazione della distribuzione di acque irrigue agli agricoltori del Consorzio di Bonifica dell'Emilia Centrale (CBEC).

Bibliografia

- [1] OCSE. Managing water sustainably is key to the future of food and agriculture. <https://www.oecd.org/agriculture/topics/water-and-agriculture>.
- [2] ISTAT. Utilizzo della risorsa idrica a fini irrigui in agricoltura. https://www.istat.it/it/files/2014/11/Utilizzo_risorsa_idrica.pdf, 2014.
- [3] Consorzio di Bonifica dell'Emilia Centrale. Utilizzo della risorsa idrica a fini irrigui in agricoltura. <http://www.emiliacentrale.it/cosa-facciamo/tutela-dellambiente>.
- [4] <http://swamp-project.org>.
- [5] Unibo Magazine. Tecnologia a servizio dell'agricoltura: con swamp un sistema per ridurre lo spreco d'acqua, 2019.
- [6] LoRa-Alliance. A technical overview of lora® and lorawan™. <https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>, 2015.
- [7] Wikipedia. Modello OSI. https://it.wikipedia.org/wiki/Modello_OSI.
- [8] Semtech. An1200.22 - lora™ modulation basics. <https://www.semtech.com/uploads/documents/an1200.22.pdf>, 2015.
- [9] Augustin Aloÿs Yi Jiazi Clausen Thomas Heide Townsley William. A study of lora: Long range low power networks for the internet of things. <https://www.mdpi.com/1424-8220/16/9/1466/pdf>, 2016.

BIBLIOGRAFIA

- [10] <https://loro-alliance.org>.
- [11] The Things Network. Lorawan overview. <https://www.thethingsnetwork.org/docs/lorawan>.
- [12] <https://www.thethingsnetwork.org>.
- [13] METER Group. Gs3 user manual. http://library.metergroup.com/Manuals/20429_GS3_Web.pdf, .
- [14] <https://www.metergroup.com>.
- [15] METER Group. Gs3 integrator guide. <http://publications.metergroup.com/Integrator%20Guide/18256%20GS3%20Integrator%20Guide.pdf>, .
- [16] STMicroelectronics. Um2115 - user manual of b-l072z-lrwan1. https://www.st.com/resource/en/user_manual/dm00329995.pdf, 2018.
- [17] STMicroelectronics. Stm32l072cz - datasheet. <https://www.st.com/resource/en/datasheet/stm32l072cz.pdf>, 2017.
- [18] Texas Instruments. Ts5a3359 1- sp3t bidirectional analog switch - datasheet. <http://www.ti.com/lit/ds/symlink/ts5a3359.pdf>, 2016.
- [19] CUI Inc®. Sj-352x-smt audio jack - datasheet. https://www.mouser.it/datasheet/2/670/sj-352x-smt_series-1313220.pdf, 2018.
- [20] Analog Devices. Adp160/adp161/adp162/adp163 - datasheet. https://www.analog.com/media/en/technical-documentation/data-sheets/ADP160_161_162_163.pdf, 2013.
- [21] STMicroelectronics. Stlq015 - datasheet. <https://www.st.com/resource/en/datasheet/stlq015.pdf>, 2016.
- [22] STMicroelectronics. Ld39050 - datasheet. <https://www.st.com/resource/en/datasheet/ld39050.pdf>, 2017.
- [23] <http://kicad-pcb.org>.
- [24] <http://www.arces.unibo.it/en>.

Elenco delle figure

1.1	Modello ISO-OSI	9
1.2	Spreading Process [8]	11
1.3	De-spreading Process [8]	12
1.4	Segnale modulato con modulazione LoRa [9]	13
1.5	Topologia di una rete LoRaWAN [11]	14
1.6	Funzionamento di un nodo in Classe A [11]	16
1.7	Funzionamento di un nodo in Classe C [11]	16
1.8	Mappa dei gateways attualmente connessi al network TTN [12] .	19
1.9	Mappa dei gateways attualmente connessi al network TTN, località Fosdondo (RE)[12]	19
2.1	Sensore METER ECH ₂ O GS3 [13]	21
2.2	Esempio di installazione del sensore [13]	22
2.3	Connettore di tipo Jack maschio per l'alimentazione ed il controllo del sensore GS3	24
2.4	Modalità di comunicazione DDI [15]	25
2.5	Esempio di simbolo in DDI mode, trasmissione del carattere 0x39 [15]	25
2.6	Comunicazione in SDI-12 mode tra Data Logger e sensore [15] .	26
2.7	Esempio di simbolo in SDI-12 mode, trasmissione del carattere 0x31 [15]	27
2.8	Trasmissione di un frame in DDI mode	28
3.1	Schematico esemplificativo di un SP3T [18]	34
3.2	Larghezza di banda a 5 V di alimentazione [18]	35
3.3	Configurazione dei pin del modello SJ-3523-SMT [19]	36
3.4	Caratteristica V_{dropout} vs I_{load} del regolatore ADP163 [20]	37

ELENCO DELLE FIGURE

3.5	ADP163, schematico esemplificativo [20]	38
3.6	Caratteristica IQ vs Temperature del regolatore LD39050 [22]	39
3.7	Caratteristica IQ vs Temperature del regolatore STLQ015 [21]	39
3.8	B-L072Z-LRWAN1 LoRa®/Sigfox™ Discovery kit [16]	42
3.9	Schematico del nodo multi-sensore	42
3.10	Schematico - Blocco di alimentazione	43
3.11	Schematico - Distribuzione e gestione dell'alimentazione dei sensori	44
3.12	Schematico - Gestione della comunicazione tramite UART	44
3.13	Schematico - Gestione della lettura del livello di carica della batteria	45
3.14	Schematico - Connettori per ancoraggio al microcontrollore	45
3.15	PCB del nodo multi-sensore	46
3.16	Rappresentazione 3D del PCB, livello superiore	47
3.17	Rappresentazione 3D del PCB, livello inferiore	47
5.1	Misura di correnti e tensioni durante un ciclo di funzionamento a regime con SF12 e tre sensori collegati	64
5.2	Misura di correnti e tensioni durante un ciclo di start-up con SF12 e un sensore collegato	66
5.3	Misura di correnti e tensioni durante un ciclo di funzionamento a regime con SF7 e tre sensori collegati	66
5.4	Misura di correnti e tensioni durante un ciclo di start-up con SF7 e un sensore collegato	67
6.1	Esempio di associazione di un device all'applicazione sulla piattaforma TTN	70
6.2	Esempio di ricezione di un pacchetto tramite la piattaforma TTN	72

Elenco delle tabelle

3.1	Confronto tra i wake-up time e il current consumption delle modalità low-power	32
3.2	Tabella di funzionamento dello switch TS5A3359	34
5.1	Stima del consumo di corrente medio per numero di sensori inseriti a diversi valori di SF	67