

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CAMPUS DI CESENA
SCUOLA DI SCIENZE

Corso di Laurea in Ingegneria e Scienze Informatiche

INTERNET OF THINGS NEL SETTORE NAUTICO:
PROGETTAZIONE E SVILUPPO DI UN SISTEMA
PER IL TRACCIAMENTO E MONITORAGGIO
LOW POWER NELLA BANDA UHF

Elaborata nel corso di: Sistemi Embedded e Internet of Things

Tesi di Laurea di:
ANTONIO TAGLIENTE

Relatore:
Prof. ALESSADRO RICCI

ANNO ACCADEMICO 2016-2017
SESSIONE II

PAROLE CHIAVE

IoT

OTA Communication

Pervasive Computing

Low power device

GPS

Indice

Introduzione	vii
1 IoT e applicazioni nel settore nautico	1
1.1 Pervasive Computing e IoT	1
1.2 Privacy vs Servizi, sfide e dubbi del terzo millennio	2
1.3 BigData, la risorsa del futuro	3
1.4 Tracciamento: un'applicazione nel settore nautico	4
2 Il progetto OneMile	6
2.1 Introduzione	6
2.2 Requisiti e funzionalità	6
2.2.1 Obiettivi e fabbisogni	7
2.2.2 Studio di fattibilità: tecnologie a confronto	8
2.2.3 Analisi dei requisiti	12
3 Prototipazione dispositivi per il tracciamento	15
3.1 Progettazione sistema di comunicazione	15
3.2 Scelte tecnologiche	17
3.2.1 Microprocessori e microcontrollori	17
3.2.2 Comunicazione UHF	19
3.2.3 Dispositivo per il tracciamento	22
3.3 Realizzazione prototipale	24
3.3.1 Progettazione nodo	24
3.3.2 Implementazione nodo	32
3.3.3 Progettazione e implementazione collettore	38

4	Testing	43
4.1	Power Consumption	43
4.2	RF Interference Analysis	45
4.3	OTA transmission	51
5	Conclusioni e sviluppi futuri	53
5.1	Resoconto	53
5.2	Nuove sfide	53
	Ringraziamenti	55

Introduzione

L'Internet delle Cose (Internet of Things, IoT) è al centro di quella che oggi viene definita come quarta rivoluzione industriale, che vede l'utilizzo capillare e pervasivo di tecnologie ICT sia per innovare processi e sistemi con cui le imprese e industrie lavorano, sia per creare servizi e applicazioni innovative. In questo elaborato di tesi è stato considerato un caso di studio che vede l'applicazione di IoT al settore della nautica. Il caso di studio concerne lo sviluppo di un progetto - denominato OneMile - volto a risolvere una serie di problematiche proposte da un'azienda di noleggio imbarcazioni leggere (InterCortal SLU, con sede in Spagna). Tali problematiche spaziano dal controllo e monitoraggio dell'andamento lavorativo al tracciamento delle imbarcazioni. Il progetto nel suo complesso si propone di progettare e implementare un sistema per il tracciamento delle imbarcazioni, fornendo la possibilità di gestione e controllo mediante sito Internet e applicazione mobile. E' stato ideato, seguito e sviluppato da un team composto dall'autore di questa tesi (Antonio Tagliente) insieme a compagni di Università (Davide Del Vecchio ed Elia Bracci), ognuno dei quali ha approfondito e sviluppato aspetti specifici del progetto - descritti nelle relative tesi. Aspetto specifico approfondito e discusso in questa tesi è il sistema di comunicazione tra l'imbarcazione ed il sistema di stoccaggio dati. Nel primo capitolo il lettore verrà introdotto alle aree tematiche coinvolte e verranno mostrati alcuni esempi esemplificativi nella vita reale. Nel secondo capitolo si parlerà del progetto OneMile nella sua interezza, per arrivare al Capitolo 3 in cui, sulla base del quadro generale introdotto nel capitolo precedente, ci si concentrerà sulla comunicazione dei dati tra l'imbarcazione ed il sistema di elaborazione e salvataggio dati.

Gli ultimi due capitoli invece, tratteranno la fase di testing e conterranno un resoconto sui dati raccolti proiettando lo sguardo agli sviluppi futuri.

Capitolo 1

IoT e applicazioni nel settore nautico

1.1 Pervasive Computing e IoT

“The Internet of Things is not a concept; it is a network, the true technology-enabled Network of all networks.”

Edewede Oriwoh

IoT, acronimo inglese di Internet of Things, rappresenta ad oggi l'evoluzione quasi "obbligata" del vecchio modo di vedere le cose, trasformandole da meri oggetti passivi o strumenti volti allo svolgimento di una funzione mediante l'intervento dell'uomo, ad entità indipendenti e cooperanti in cui l'intervento dell'uomo è fortemente limitato se non inesistente. Con lo sviluppo della microelettronica si è arrivati a SoC e sistemi low power di piccolissime dimensioni rendendoli un prelibato derivato tecnologico adatto ai più svariati ambiti della ricerca e sviluppo. Dotando ogni oggetto della vita quotidiana di capacità computazionale si sancisce di fatto la nascita del Pervasive Computing o Ubiquitous Computing, dove al mondo fisico viene aggiunto un layer computativo che ne aumenta le potenzialità e permette l'interazione con l'uomo o con le tecnologie create da quest'ultimo in maniera autonoma.

Tale processo ha portato tutte le aree, dal business alle scienze sociali, a rivedere i modelli fino ad ora conosciuti e usati, aprendo nuovi interrogativi e nuove sfide su un terreno totalmete nuovo dove i limiti non sono netti o facilmente distinguibili. Basti pensare a come i social abbiano fuso insieme informatica, elettronica, le scienze sociali ed il business, creando nuovi processi di interazione e nuovi modelli di marketing.

1.2 Privacy vs Servizi, sfide e dubbi del terzo millennio

”E’ meglio essere ingannati
centinaia di volte che vivere una
vita di sospetti.”

Charles Haddon Spurgeon

Il tema della privacy è diventato molto delicato e ricorrente, poichè per avere servizi sempre più adatti alle nostre esigenze o di facile fruibilità, si necessita di una cessione dei nostri dati o delle nostre informazioni personali a terzi, permettendo, qualora utilizzati in modo deviato, di ledere la persona in maniera diretta o indiretta. Le Tecnologie trattate nel presente documento quali il posizionamento ed il tracciamento possono condurre il lettore verso una visione riduttiva e superficiale, fino ad arrivare alla negazione dell’utilità di tale tecnologia in ambiti quali la sicurezza ed il miglioramento di servizi già presenti sul territorio, sentendo ovunque odore di “fumus persecutionis”. Tuttavia tali paure non devono frenare l’evoluzione quasi naturale delle interazioni uomo macchina che sempre di più stanno diventando invisibili e pervasive, dotando l’essere umano di nuovi sensi e nuove potenzialità. Quindi, in questo documento ci soffermeremo e daremo maggior enfasi ai soli aspetti positivi e alle nuove capacità che tali tecnologie ci offrono.

1.3 BigData, la risorsa del futuro

“There were 5 Exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days.”

Eric Schmidt, Google’s CEO.

Il suddetto layer computativo produce in maniera sistematica enormi volumi di dati. Tali dati devono essere smistati, filtrati e stoccati in opportune strutture dati che ne permettano la fruibilità in tempi utili, e una concretezza semantica in grado di poter generare modelli di previsione o modelli comportamentali. Per poter utilizzare tali dati e trovare delle correlazioni negli stessi, si necessita di algoritmi di esplorazione performanti in grado di elaborare enormi moli di dati in tempi-uomo utili. A tal proposito ci vengono incontro le intelligenze artificiali e gli algoritmi di data mining un esempio concreto viene mostrato nell’articolo del Sole24Ore del 11 Marzo 2017 [1]:

“...ha sviluppato processi di analisi del linguaggio per «scandagliare» le conversazioni dei social media. La tecnologia permette di processare più di tre miliardi di messaggi al mese. L’obiettivo? Individuare eventuali correlazioni con l’andamento dei prezzi di Borsa...”

Il risultato raggiunto è strabiliante:

”Tra il 2014 e la fine del 2016 - dice Sastri - il ritorno cumulato dell’algoritmo basato sull’intelligenza artificiale è stato del 67,23%. Quello dell’approccio “buy and hold”, invece, si è assestato al 23,33%.”

Non possiamo quindi esimerci dal progettare o quanto meno predisporre, nei nuovi dispositivi, una componente comunicativa e di monitoraggio che renda quest’ultimi parti attive e interagenti in un sistema sempre più interattivo e connesso.

1.4 Tracciamento: un'applicazione nel settore nautico

"New technology is not good or evil in and of itself. It's all about how people choose to use it."

David Wong

Sulla base di quanto detto nelle sezioni precedenti, sempre più aree hanno adottato il tracciamento come soluzione principale a problemi quali la sicurezza, il monitoraggio remoto o semplicemente come mezzo di orientamento. Grazie ai nuovi dispositivi mobili come smartphone o smartwatch la geolocalizzazione è entrata di fatto nella nostra vita, basti pensare a quante volte tale tecnologia viene utilizzata rendendo la vita dell'utilizzatore più semplice e pratica. Settori come quello militare e quello dell'automotive sono stati i pionieri della geolocalizzazione aprendo la strada ad altri settori che oggi si stanno muovendo in questo senso. Settori come quello assicurativo, quello della sicurezza e quello ambientale stanno integrando nei loro dispositivi dei sistemi di geolocalizzazione e tracciamento per i motivi più disparati, grazie alla versatilità di questa meravigliosa tecnologia. Anche nel settore della nautica turistica è nata l'esigenza di tracciare e monitorare lo spostamento di imbarcazioni leggere in maniera del tutto invisibile all'utilizzatore che, ignaro di tutto, può godersi la sua attività preferita senza preoccuparsi di comunicare la propria posizione in maniera periodica. A questo si aggiunge un'altra esigenza in questo settore: la sicurezza. Monitorare la posizione permette inoltre di capire quando un'imbarcazione esce dalle zone di sicurezza o rimane troppo tempo ferma in una zona, oppure semplicemente supera le distanze consentite dall'azienda proprietaria delle stesse in un contesto di noleggio. Un progetto a riguardo, cofinanziato dall'Unione Europea, di nome Greenway si occupa proprio di monitorare, con le tecnologie offerte dall'IoT, le imbarcazioni registrate.

La descrizione del progetto infatti riporta: "L'applicazione GreenWay è un app IoT che funziona esclusivamente in ambiente Lysis. Non è un'applicazione mobile, ma è una applicazione cloud IoT che permette a qualsiasi dispositivo Lysis-enabled (cioè registrato sulla piattaforma Lysis) di comunicare con le imbarcazioni dotate di sistema Greenboxnautica. Greenway è

in grado di cercare in modo automatico tutte le barche dell'utente registrate e di impostare un inoltro degli alert dalla barca allo smartphone dell'armatore senza nessuno sforzo per l'utente" [2]. In tale contesto si colloca il progetto OneMile che verrà ampiamente discusso nei prossimi capitoli.

Capitolo 2

Il progetto OneMile

2.1 Introduzione

Sulla base di quanto detto nel capitolo precedente, l'azienda InterCortal SLU di Xavier Cortal, con sede in Spagna, ha ritenuto opportuno dotare le sue imbarcazioni leggere di sistemi geolocalizzanti in grado di tracciare le posizioni, memorizzarle e mostrarle a video nel modo più pratico possibile.

2.2 Requisiti e funzionalità

Di seguito verranno discusse le componenti e le scelte implementative che hanno portato alla realizzazione di un primo prototipo funzionante. Dall'intervista fatta al proprietario dell'azienda si è data maggiore enfasi al noleggio in spiaggia dei pedalò dove sono stati evidenziati gli step che un operatore compie durante lo svolgimento della sua mansione:

- L'operatore attende l'arrivo del cliente
- Al momento del noleggio il cliente paga la quota di listino e gli viene rilasciata la ricevuta
- Al rientro l'operatore riposiziona il pedalò, rendendolo di nuovo disponibile

2.2.1 Obiettivi e fabbisogni

Discutendo a lungo sul dominio applicativo, sono state stilate quelle le problematiche principali da risolvere, di seguito definite:

1. monitoraggio remoto delle attrezzature dislocate sul territorio
2. monitoraggio del corretto svolgimento delle attività lavorative
3. stima dei guadagni in tempo reale
4. dispositivi di trasmissione del pedalò a basso consumo

È possibile, sulla base di quanto detto, stilare un primo schema che mette in luce gli attori che entrano in gioco nel sistema.

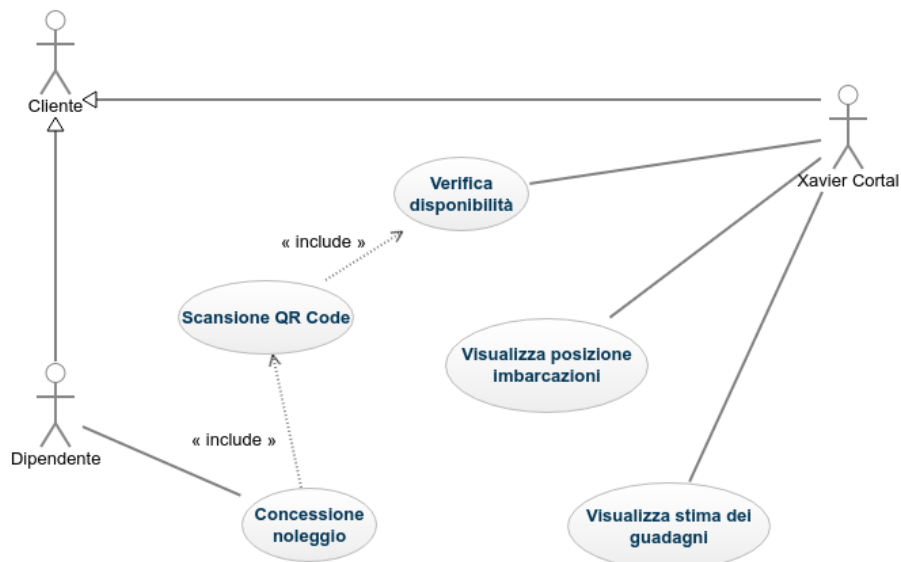


Figura 2.1: Schema dei casi d'uso

Nel punto uno si definisce la necessità di visualizzare la posizione approssimativa delle attrezzature, al fine di evitare frodi o atti di evasione da parte dell'operatore che noleggia il pedalò che, in caso di ricevana non rilasciata,

sottrae indebitamente profitti all'azienda. Nel secondo punto invece si chiede di creare una piattaforma per l'operatore monitorabile da remoto dove al posto del carnet di ricevute si ha una scansione del QR code associato all'imbarcazione. Grazie a questa piattaforma si ha una stima abbastanza attendibile di quanti pedalò sono disponibili e quanti invece sono in mare, ciò si traduce in un monitoraggio in tempo reale dei guadagni. Per l'ultimo punto invece, si tratta di creare un dispositivo che viene inglobato all'interno del pedalò la cui durata, in termini di batteria, deve essere la più lunga possibile.

2.2.2 Studio di fattibilità: tecnologie a confronto

In fase embrionale, per la comunicazione della posizione delle imbarcazioni, è stato aperto un ventaglio di opzioni e quelle prese in considerazione sono di seguito elencate:

- LoRa
- SigFox
- Moduli GSM
- LPD433

LoRa

LoRaWAN™ è una Low Power Wide Area Network (LPWAN). LoRaWAN dà particolare importanza ai requisiti fondamentali dell'IoT come la comunicazione bidirezionale sicura, la mobilità e i servizi di localizzazione. La specifica LoRaWAN permette l'interoperabilità tra gli smart dispositivi senza complesse installazioni locali, lasciando piena libertà a sviluppatori e user. Questa architettura di rete ha come topologia la star-of.star in cui i gateway sono dei bridge invisibili tra gli end-devices e i server del backend. I gateway sono connessi alla rete tramite un connessione con protocollo IP standard mentre gli end-devices una comunicazione wireless single-hop. Tutti gli end-devices supportano la comunicazione bi-direzionale ma in caso di necessità il sistema supporta anche operazioni di tipo multicast ad esempio per aggiornare il software o mandare messaggi in maniera distribuita,

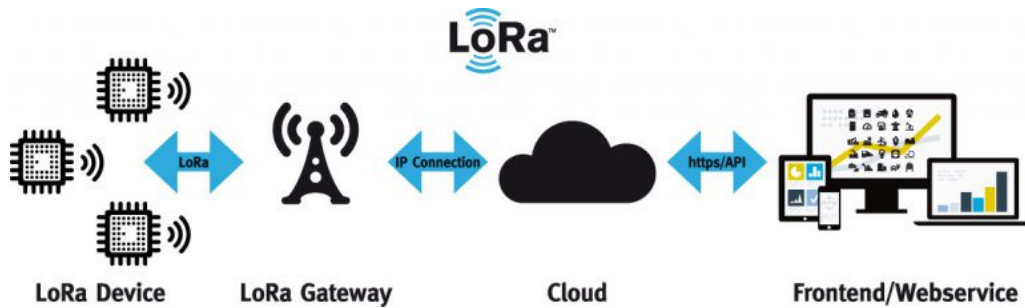


Figura 2.2: Schema generale di funzionamento [4]

riducendo il tempo di comunicazione totale [3]. Di seguito uno schema riassuntivo del sistema in tutte le sue componenti.

Questa tecnologia però, oltre ad avere un costo fisso ed una dimensione pacchetti limitata, non ha peraltro piena copertura in Spagna, come possiamo vedere in Fig 2.3, rendendola di fatto una scelta non applicabile ad un problema come il nostro.

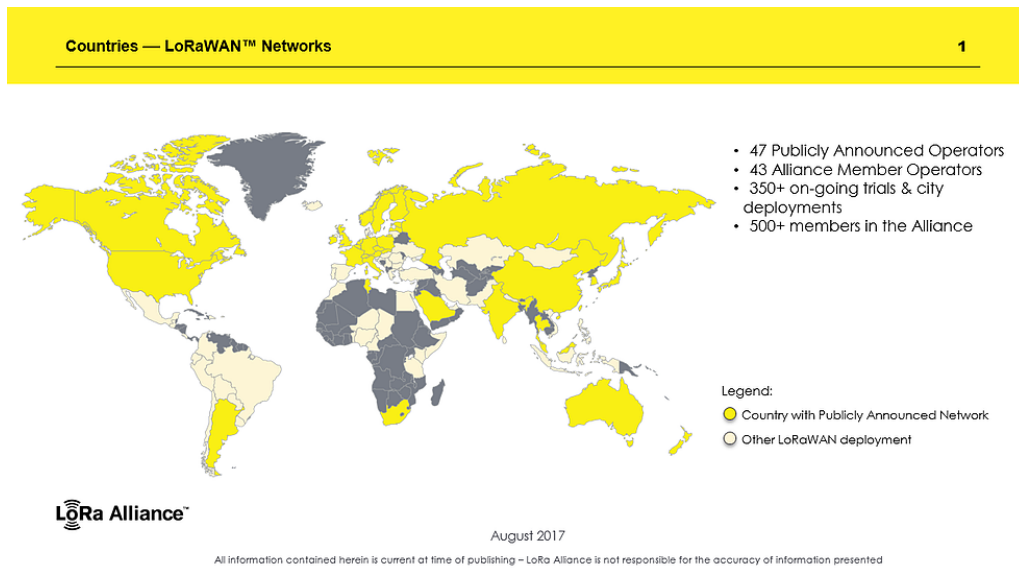


Figura 2.3: Schema generale di funzionamento [5]

SigFox

SigFox è un sistema di trasmissione wireless operante sulle frequenze ISM in grado di coprire notevoli distanze (long-range) grazie alla elevatissima sensibilità del ricevitore che sfrutta la tecnologia Ultra Narrow Band. I consumi sono stati notevolmente ridotti grazie al numero di messaggi in uscita dagli end-devices molto limitato, circa uno ogni quindici minuti.

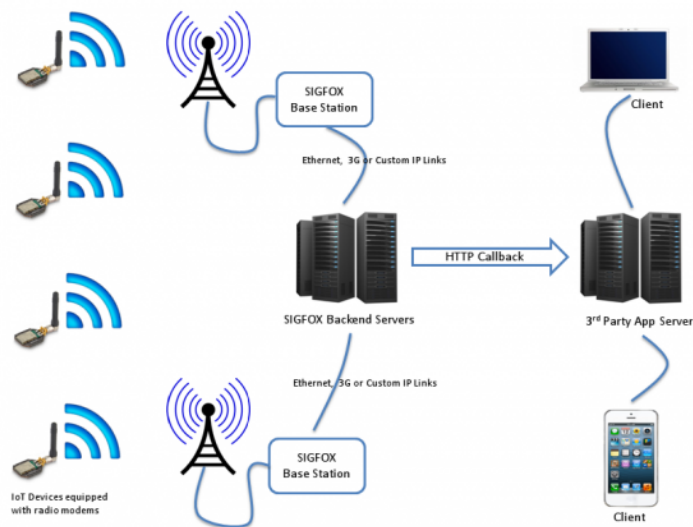


Figura 2.4: Schema architetturale [6]

Nelle applicazioni IoT che utilizzano la banda ISM non è possibile aumentare la potenza del TX che, al contrario, deve essere la più bassa possibile e deve rientrare nei limiti stabiliti dalle normative vigenti. Con queste potenze e con una sensibilità standard del ricevitore dell'ordine di -110 dBm la portata, in ogni caso, non supera i 100-300 metri. La sensibilità di un ricevitore è fortemente influenzata dal rumore intrinseco del dispositivo e da quello presente nell'etere. Questa tecnologia, oltre ad avere un costo fisso, ha un payload ancor più limitato per gli end-devices. D'altro canto aziende come STMicroelectronics, ON Semiconductor, Atmel e Texas Instruments stanno adottando SigFox come tecnologia nei loro dispositivi low-power rendendola di fatto una scelta appetibile. Tuttavia il costo dei dispositivi, il payload limitato e il costo fisso non la rendono la scelta migliore.

GSM e GPRS Connection

”Global System for Mobile Communications (in italiano Sistema globale per comunicazioni mobili) o GSM è lo standard di seconda generazione di telefonia mobile e al 2017 il più diffuso al mondo con più di 3 miliardi di persone in 200 Paesi. Si tratta di uno standard aperto sviluppato dalla conferenza europea delle amministrazioni delle Poste e delle Telecomunicazioni, formalizzato dall’Istituto europeo di standard telecomunicativi e reso operativo dal consorzio 3GPP” [7]- L’evoluzione della connessione GSM ha portato al GPRS, acronimo di General Packet Radio Service diventando di fatto il primo sistema cellulare progettato specificamente per realizzare un trasferimento dati a commutazione di pacchetto e a media velocità su rete cellulare per agganciarsi alla rete Internet. I moduli GSM-GPRS però, oltre ad avere un costo che si aggira intorno ai 30\$, ha un assorbimento, in termini di corrente, non compatibile con il sistema da noi progettato infatti in standby il modulo assorbe poco meno di 30mA riducendo l’autonomia complessiva. D’altro canto aziende come ”u-blox” [8] hanno lanciato moduli low power, ma rimane sempre il problema del costo di gestione mensile per il pacchetto dati fornito dal provider dei servizi.

LPD433 - Low Power Device 433

Gli LPD, acronimo di Low Power Devices, sono dei sistemi di comunicazione a breve raggio che operano nella banda UHF. ”UHF è la sigla di Ultra high frequency (frequenza ultra alta) e sta ad indicare i segnali a radiofrequenza trasmessi nella banda che va da 300 MHz a 3 GHz. Nell’ambito dell’ampia banda disponibile esistono vari tipi di applicazione per i segnali a queste frequenze” [9]. Questa tecnologia però non può comunicare direttamente con il server che andrà ad ospitare i dati, portandoci ad aggiungere una componente aggiuntiva che dovrà essere connessa ad internet e che gestirà più imbarcazioni dotate di modulo LPD433.

2.2.3 Analisi dei requisiti

Le macro funzionalità possono essere quindi riassunte in questi cinque punti:

- Reperimento posizione
- Comunicazione imbarcazione - stazione a terra
- Elaborazione
- Stoccaggio dati
- Visualizzazione

Reperimento posizione

Questa funzionalità si occupa di rilevare la posizione dell'imbarcazione.

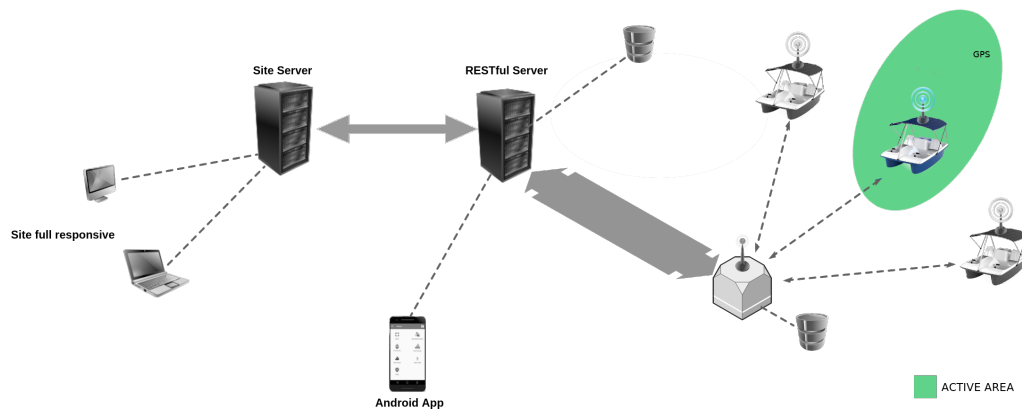


Figura 2.5: Reperimento posizione

Comunicazione imbarcazione - stazione a terra

In questa fase la posizione deve essere impacchettata e inviata a terra.

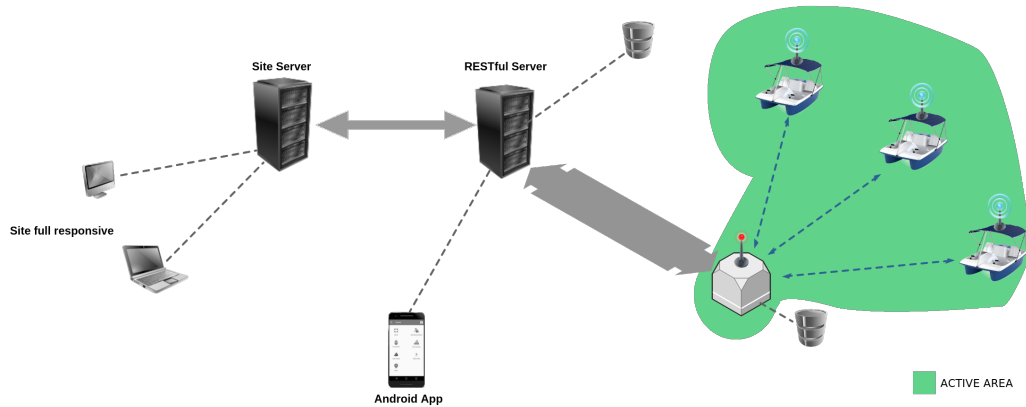


Figura 2.6: Comunicazione imbarcazione - stazione a terra

Elaborazione

La fase di elaborazione prende il messaggio in arrivo ed effettua un controllo degli errori preparandolo alla fase successiva.

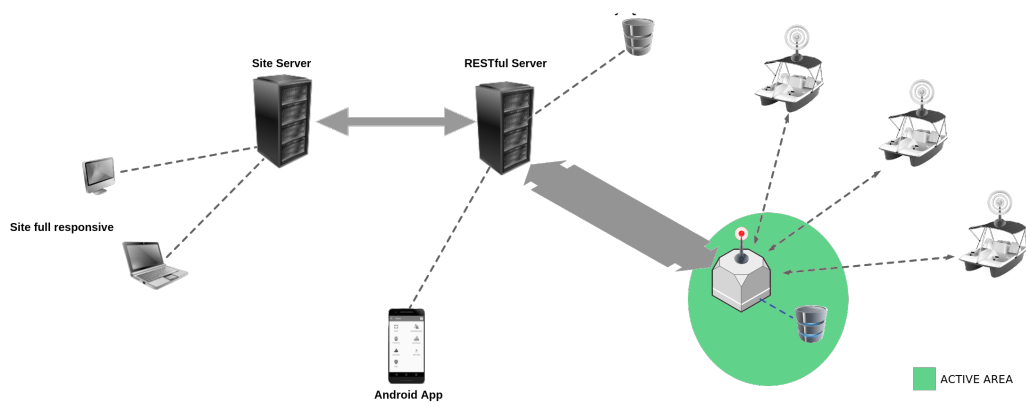


Figura 2.7: Elaborazione

Stoccaggio dati

I dati elaborati possono essere memorizzati in maniera permanente in apposite strutture dati.

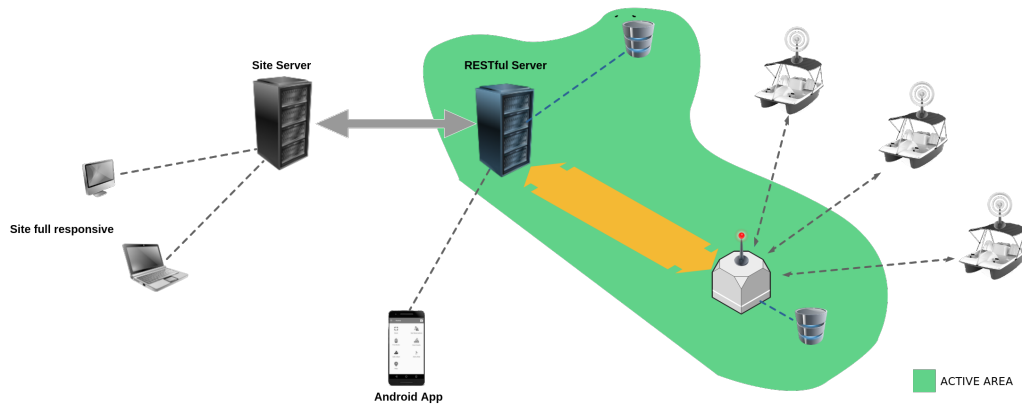


Figura 2.8: Stoccaggio dati

Visualizzazione

I dati così strutturati devono essere accessibili in maniera user-friendly dagli operatori da remoto, tramite sito e da App.

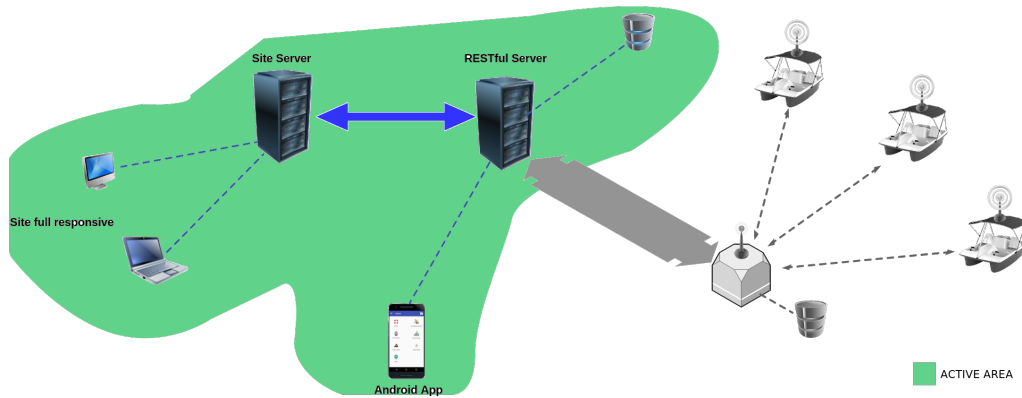


Figura 2.9: Visualizzazione

Capitolo 3

Prototipazione dispositivi per il tracciamento

3.1 Progettazione sistema di comunicazione

1. All'accensione della stazione a terra quest'ultima sincronizza il proprio DB locale con il DB online.
2. Prende la lista degli end-device registrati e li contatta uno alla volta.
3. Gli end-device trasmettono la posizione alla stazione a terra e, se il check del messaggio va a buon fine, lo salva nel DB locale.
4. Periodicamente la stazione a terra si sincronizza con il DB online.
5. I dati del DB online sono fruibili tramite sito e app
6. Grazie all'app dell'operatore è possibile noleggiare le attrezzature o le imbarcazioni scansando un QR code senza dover usare materiale cartaceo o mezzi non tracciabili.

Di seguito, in figura 3.1, possiamo vedere l'architettura del sistema con le relative tecnologie di comunicazione e le componenti principali.

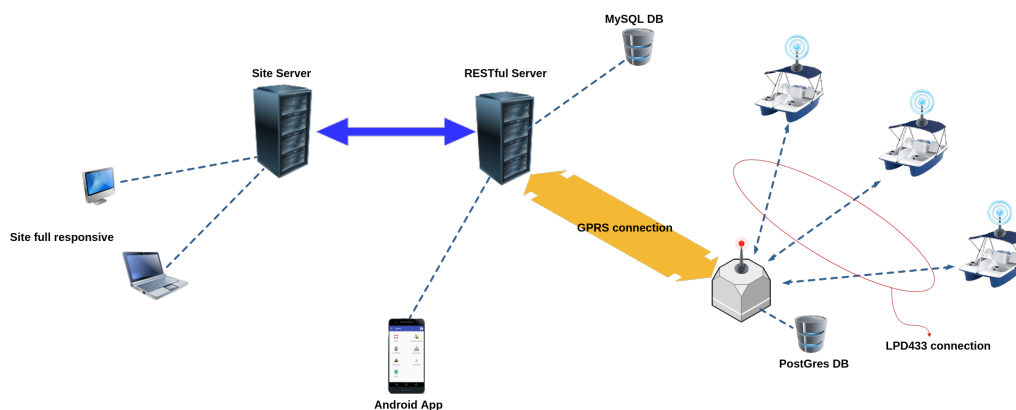


Figura 3.1: Schema architetturale

- Imbarcazioni dotate di tecnologia LPD433
- Stazione a terra con tecnologia GSM/GPRS e LPD433 e DB locale
- Server RESTful che contiene i dati di tutto il sistema
- Site Server che contiene il sito di monitoraggio

Tali tecnologie comunicative però devono essere inserite in device programmabili ed a basso consumo energetico che abbiano costi accessibili e siano facilmente reperibili. La scelta è ricaduta inevitabilmente su Arduino per quanto riguarda le imbarcazioni e Raspberry per quanto riguarda le stazioni a terra.

3.2 Scelte tecnologiche

3.2.1 Microprocessori e microcontrollori

Arduino

Arduino è una piattaforma hardware composta da una serie di schede elettroniche dotate di un microcontrollore. È stata ideata e sviluppata da alcuni membri dell'Interaction Design Institute di Ivrea come strumento per la prototipazione rapida e per scopi hobbistici, didattici e professionali [10].

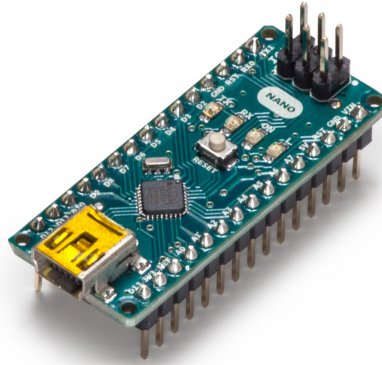


Figura 3.2: Schema Arduino nano [11]

Tale dispositivo è facilmente programmabile grazie all'ide di sviluppo multipiattaforma e all'enorme supporto da parte della community. Sensoristica e componentistica elettronica è facilmente integrabile grazie alla presenza di schede applicative chiamate "shields" che vengono agganciate ai connettori, permettendo di espandere le potenzialità di Arduino rendendolo molto versatile.

Raspberry

A differenza dell'Arduino questa piattaforma possiede un microprocessore Broadcom ARM ed un sistema operativo basato su kernel linux o RISC OS. Grazie a porte USB, bluetooth, wifi e hdmi è possibile utilizzarlo in innumerevoli campi anche grazie alla GPIO completamente programmabile.



Figura 3.3: Schema Raspberry Pi 3 [12]

Quello utilizzato per il progetto è il Raspberry Pi 3, il quale presenta alcune novità:

- primo modello con CPU a 64 bit da 1.2GHz
- cache da 512KB
- ram da 1Gb LPDDR2 da 900Mhz
- modulo Bluetooth 4.1 low energy
- modulo Wi-Fi 2.4Ghz 802.11n

3.2.2 Comunicazione UHF

Per la comunicazione su questa banda sono stati utilizzati, in una prima versione del prototipo, moduli comuni come MX-05V e MX-FS-03V che vengono largamente utilizzati nei comandi a distanza in ambito civile (apri-cancello, allarmi, antifurto). Grazie a questi componenti è possibile estendere la seriale al mondo delle connessioni senza filo con pochissime righe di codice ed una componentistica aggiuntiva quasi nulla.

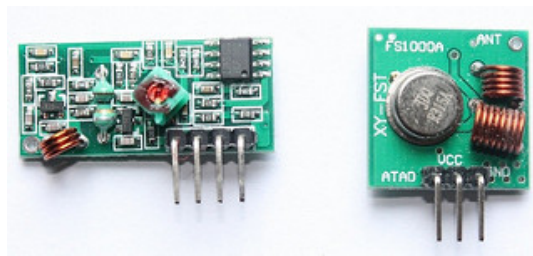


Figura 3.4: Ricevitore a sinistra e trasmettitore a destra [13]

Ricevitore MX-05V

Il ricevitore ha dimensioni 30mm x 14mm x 7mm con un costo veramente esiguo. Dal datasheet possiamo estrapolare i seguenti dati:

- Tensione di lavoro: DC5V
- Assorbimento: 4mA
- Frequenza di lavoro: 315Mhz
- Sensibilità: -105Db

MX-FS-03V

Il trasmettitore ha dimensioni più piccole 19mm x 19mm con un costo sempre bassissimo.

- Tensione di lavoro: 3.5-12V
- Modulazione: AM
- Velocità di trasferimento: 4KB / S
- Potenza in trasmissione: 10mW
- Frequenza di lavoro: 315Mhz

A queste componenti è stata aggiunta un'antenna stilo auto-costruita di lunghezza pari $\frac{\lambda}{4}$ dove λ corrisponde alla lunghezza d'onda di 315MHz. Di seguito vediamo i moduli con l'antenna ricavata da filo in rame con impedenza di 50Ω .

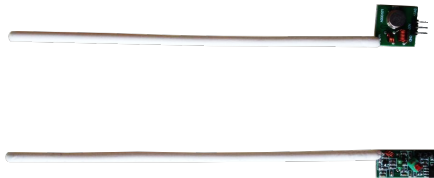


Figura 3.5: Moduli equipaggiati con antenna autocostruita

LPD433 modulo HC-12

Questo modello, molto più performante dei precedenti, contiene sia il trasmettitore che il ricevitore e permette di raggiungere distanze ben maggiori, questo però a scapito dell'autonomia della batteria.

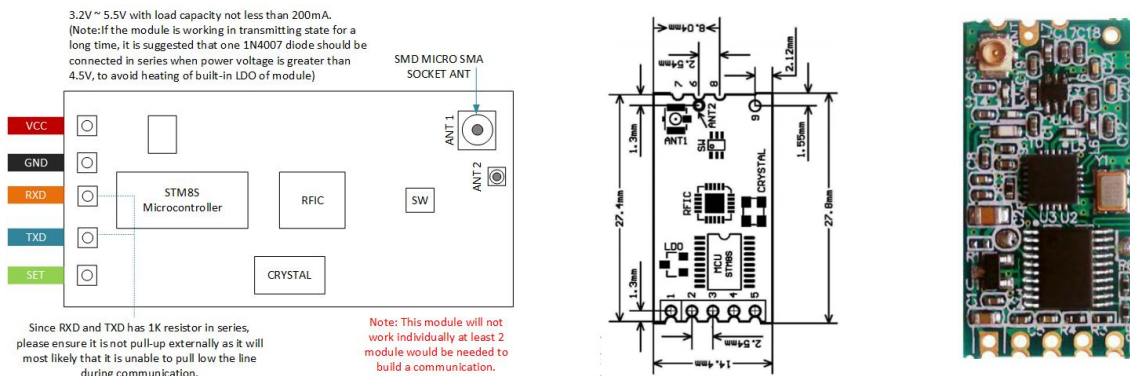


Figura 3.6: HC12 Module scheme [14]

Tale modulo ha un ampio spettro di funzionalità tra cui annoveriamo:

- Trasmissione a lunga distanza (1000m in campo aperto a 5000bps)
- Range di frequenze comprese tra 433.4 e 473.0MHz con oltre 100 canali di comunicazione.
- 8 possibili livelli di trasmissione con un massimo di 100mW
- Tre modalità di lavoro per adattare il modulo ai vari campi di utilizzo
- MCU incorporato, permette la comunicazione con il dispositivo esterno tramite la porta seriale
- The number of bytes transmitted unlimited to one time
- Update software version

Antenne e connettori

Le antenne utilizzate sono da esterno ed offrono 5Db di guadagno con dei convertitori SMA - micro SMA che permettono la connessione al modulo HC-12.



Figura 3.7: Antenna da 433Mhz con convertitore SMA - micro SMA

3.2.3 Dispositivo per il tracciamento

Modulo GPS

Per reperire la posizione si è optato per un chip neo-6 series della u-blox. Le caratteristiche di questo chip sono:

- UART, USB, DDC (conforme I2C) e interfaccia SPI
- Disponibile con oscillatore a cristallo o TCXO
- Cristallo RTC per una ricerca dei satelliti più veloce
- Versione da 1.8 e 3.0 V
- Assorbimento 100mA

Tutte le componenti sono montate su un PCB dove sono presenti componenti quali antenna, componentistica di stabilizzazione dell'alimentazione ed altri controlli sull'input. Purtroppo questo modello non contempla la modalità power-save, tale problematica è stata risolta e trattata più avanti.

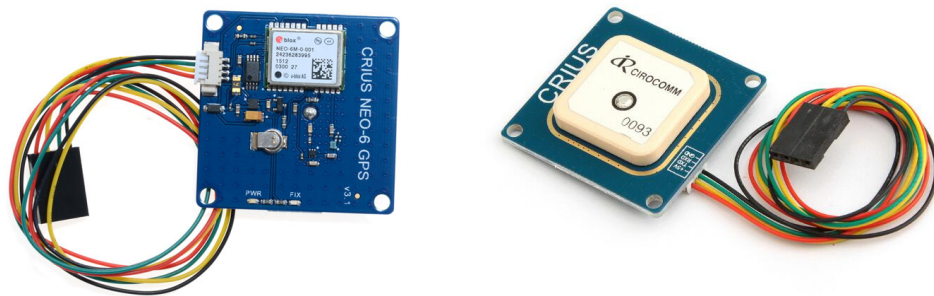


Figura 3.8: Modulo GPS

Mosfet IRF520

Questo mosfet di potenza è stato utilizzato per ovviare al problema dei consumi, controllando l'alimentazione su determinate aree del sistema, riducendo notevolmente il fabbisogno di corrente.

Questo componente permette di controllare tensioni fino a 100V e correnti fino a 9.2A con switching time nello'ordine del nano secondo. La resistenza interna inoltre è 0.270Ω , riducendo i fenomeni resistivi parassiti e di conseguenza l'effetto joule sul componente.

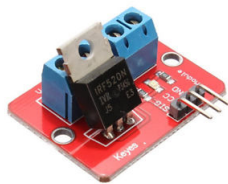


Figura 3.9: Modulo IRF520

3.3 Realizzazione prototipale

Come possiamo vedere nella sezione precedente i componenti in gioco sono vari e diversificati. Di seguito verranno mostrate e commentate le scelte progettuali che hanno portato ad un primo dispositivo tracciante che rispecchia le specifiche iniziali. Per semplicità e generalizzazione del sistema si è scelto di chiamare **nodo** il sistema tracciante sull'imbarcazione e **collettore** la stazione a terra.

3.3.1 Progettazione nodo

Il nodo è suddiviso in macro unità ognuna con le sue caratteristiche e funzionalità, di seguito elencate e schematizzate.

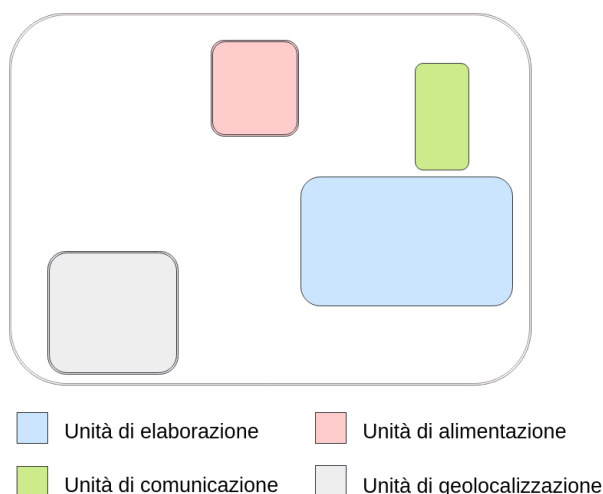


Figura 3.10: Schema delle macro unità

Elenchiamo quindi le unità con i rispettivi componenti utilizzati:

- **Alimentazione** IRF520
- **Geolocalizzazione** Modulo GPS neo-06
- **Comunicazione** HC-12 con antenna omni-direzionale
- **Elaborazione** Arduino Nano

Unità di alimentazione

Questa unità si occupa di gestire l'alimentazione delle componenti in maniera controllata. Ad essa infatti è delegato il controllo e la gestione dell'alimentazione del gps, questo ci permette di avere una maggiore efficienza controllando i consumi.

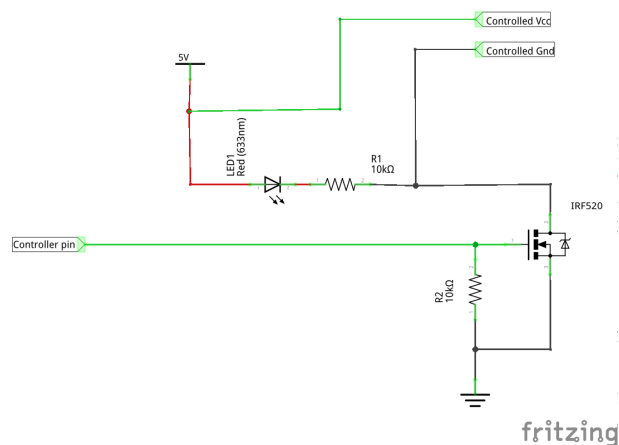


Figura 3.11: Schema elettrico

Unità di geolocalizzazione

Come unità di geolocalizzazione è stata utilizzata, come detto precedentemente, la scheda Crius 0093 con chip Neo-06. Di seguito è mostrato il collegamento e libreria per la gestione di quest'ultimo. Per chiarezza verranno mostrate solo le signature e spiegate uno alla volta.

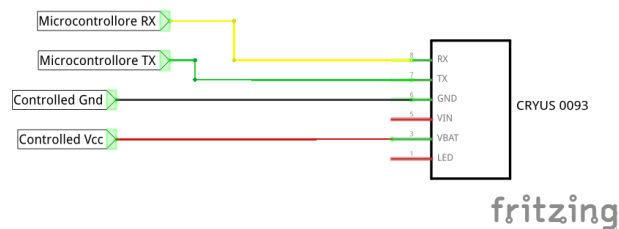


Figura 3.12: Schema elettrico GPS

GPS.hpp

```

/*
 * File:    HC12.hpp
 * Author:  Antonio Tagliente
 *
 * Created on July 4, 2017, 10:54 PM
 */

#ifndef __GPS_HPP__
#define __GPS_HPP__
#include <Arduino.h>
#include <SoftwareSerial.h>
#include "TinyGPS.h"

class GPS {
public:
    GPS(uint8_t rxPin, uint8_t txPin);
    GPS(uint8_t rxPin, uint8_t txPin, int baudRate);
    ~GPS();
    float getLatitude();
    float getLongitude();
    float getAltitude();
    void enableLog();
    void disableLog();
private:
    uint8_t _rxPin = 2;
    uint8_t _txPin = 3;
    int _baudRate = 9600;
    bool _Log = false;
    void _gpsDump();
    void _update();
    int _listenTimeout = 3000;
    int _refreshTimeout = 1000;
    int _lastPositionTime = 0;
    float _latitude = -1;
    float _longitude = -1;
    float _altitude = -1;
    TinyGPS* _gps;
    SoftwareSerial* _ss;
};

#endif

```

Dal file header possiamo notare, oltre ai due costruttori, i metodi pubblici:

- `getLatitude()`;
- `getLongitude()`;
- `getAltitude()`;
- `enableLog()`;
- `disableLog()`;

Possiamo facilmente immaginare il loro comportamento. I metodi `enableLog()` e `disableLog()` abilitano e disabilitano le stampe a video per mostrare le fasi intermedie altrimenti nascoste all'utente. Variabili come `_listenTimeOut` e `_refreshTimeout` gestiscono rispettivamente il tempo massimo in cui la porta deve rimanere in ascolto e il tempo minimo che deve intercorrere per leggere una nuova posizione. Tutta la libreria si poggia su un'altra libreria che è **TinyGPS++** creata da Mikal Hart e pubblicata su Github. L'insieme di queste parti ci ha permesso di gestire il gps con un buon livello di astrazione, delegando all'algoritmo la gestione delle varie fasi che portano alla lettura della posizione. Quando si usa una `getLatitude()` o una `getLongitude()` all'interno viene richiamato il metodo `_update()` visibile di seguito.

```
void GPS::_update() {
    long timeout = millis() - _lastPositionTime;
    if (_latitude == -1 || _longitude == -1 || timeout > _refreshTimeout)
    {
        _lastPositionTime = millis();
        if (_Log) Serial.println("—_UPDATE_—");
        _ss->listen();
        bool newdata = false;
        while (!newdata && millis() - _lastPositionTime
                < _listenTimeOut)
        {
            if (_ss->available())
            {
                char c = _ss->read();
                if (_gps->encode(c)) newdata = true;
            }
        }
    }
}
```

```

    if (newdata)
    {
        if(_Log) {
            Serial.println("\nAcquired Data");
            Serial.println("_____");
        }
        _gpsDump();
        if(_Log) Serial.println("\n_____");
    }
}

```

Il funzionamento di questo metodo è molto semplice, legge la seriale fino a quando non riceve dati consistenti o fino al momento in cui non si raggiunge il tempo di timeout. Se il dato è stato ottenuto ed è consistente si invoca il metodo `_gpsDump()`:

```

void GPS::_gpsDump() {
    unsigned long age;
    _gps->f_get_position(&_latitude, &_longitude, &age);
    _altitude = _gps->f_altitude();
    if(_Log) {
        Serial.print("Lat/Long(float): ");
        Serial.print(_latitude);
        Serial.print(", ");
        Serial.print(_longitude);
        Serial.print("Alt(float): ");
        Serial.print(_altitude);
    }
}

```

Questo metodo privato permette di estrapolare i dati e salvarli nelle apposite variabili.

Unità di comunicazione

In questa unità si gestisce la comunicazione tra il nodo ed il collettore. Per farlo è stata creata ad hoc una libreria per il modulo HC-12 che ci permette di usare le sue funzionalità in maniera più pratica e ad alto livello. possiamo vedere lo schema di collegamento e la succitata libreria.

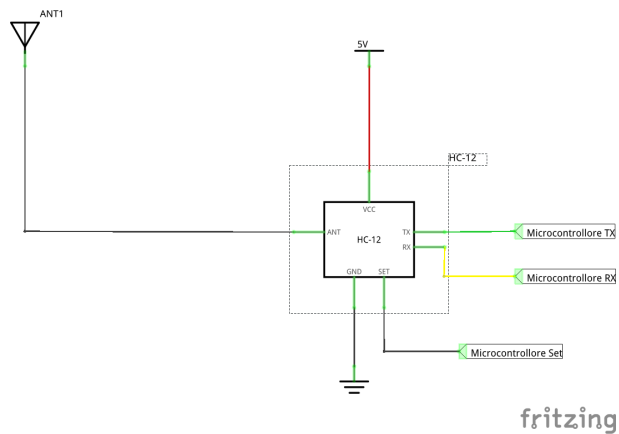


Figura 3.13: Schema elettrico HC-12

HC12.hpp

```

/*
File: HC12.hpp
Author: Antonio Tagliente

Created on July 31, 2017, 9:48 AM

*/

#ifndef HC12_HPP
#define HC12_HPP
#include "Arduino.h"
#include <SoftwareSerial.h>

class HC12 {
public:
    HC12(int rxPin, int txPin, int setPin);
    ~HC12();
    void setBaudrate(int baud);

```

```

void setTxPower(int power);
void setChannel(int channel);
void setWorkingMode(int mode);
void sleep();
void wakeUp();
void setTimeout(long milliSec);
int getBaudrate();
String getTxPower();
String getActualSetting();
int getChannel();
int getWorkingMode();
void enableLog();
void disableLog();
void resetDefault();
long getActualTimeOut();
void update();
String readString();
void sendString(String string);
private:
    /*
     * Default value
     */
    int _rxPin = 2;
    int _txPin = 3;
    int _setPin = 4;
    int _baudRate = 2400;
    long _timeOut = 200;
    boolean _Log = false;
    int _numberOfBauds = 9;
    long _availableBaudrates[9] = {1200, 2400, 4800,
                                   9600, 19200, 38400, 57600,
                                   74880, 115200};

    SoftwareSerial* _ss;
    /* Utility functions */
    String _commandBuilder(String prefix, String value);
    void _sendCommand(String command);
    bool _sleep = false;
};

#endif /* HC12_HPP */

```

Sulla base del datasheet del componente sono stati impostati i parametri interni alla libreria permettendo di avere un ottimo grado di controllo grazie ai vincoli che tali parametri impongono. Tentare, ad esempio, di

inserire baudrate fuori range o settare un canale non contemplato è pressochè impossibile grazie ai controlli sugli input. Nella fase di esecuzione dei comandi AT è la libreria ad entrare in **transparent mode** lanciando il comando e ritornando successivamente alla modalità ascolto, rendendo tutto più semplice ed intuitivo.

Unità di elaborazione

Tale unità si occupa di gestire e controllare i vari componenti ad esso associati tramite uno "sketch" che li fa interagire tra loro. Gestisce anche le tempistiche di accensione e spegnimento dei dispositivi ad esso associati aumentando vertiginosamente l'efficienza del nodo.

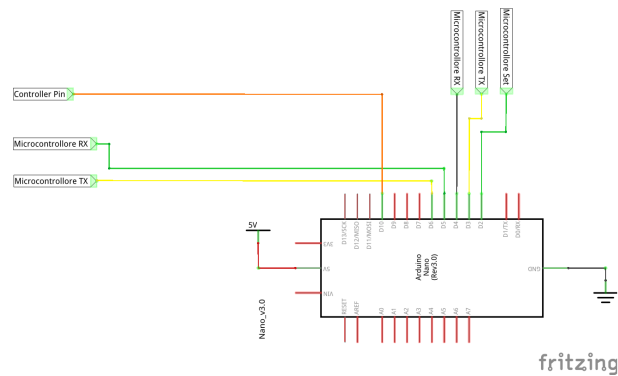
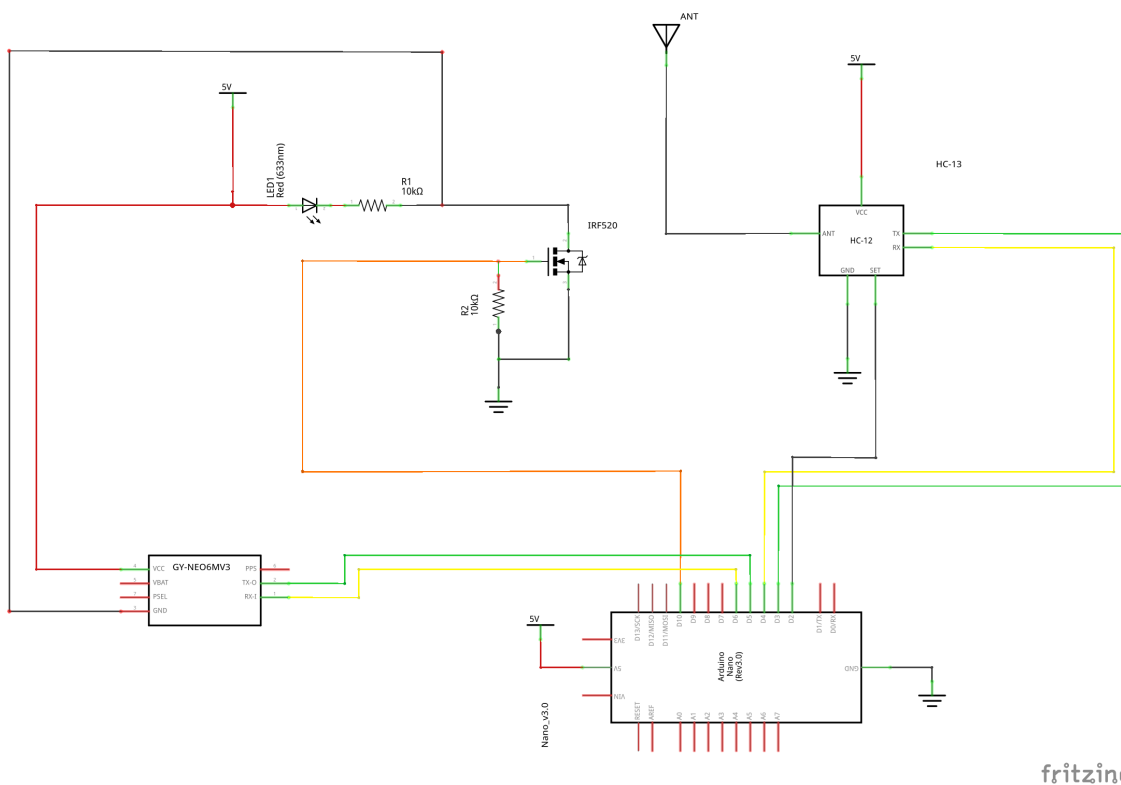


Figura 3.14: Schema elettrico Arduino

3.3.2 Implementazione nodo

Di seguito verranno mostrati due schemi: il primo è relativo al nodo completo e tracciante, mentre l'altro ad un nodo dummy che simula lo spostamento nello spazio e permette di testare connessioni multiple.



fritzing

Figura 3.15: Schema elettrico completo Nodo Tracciante

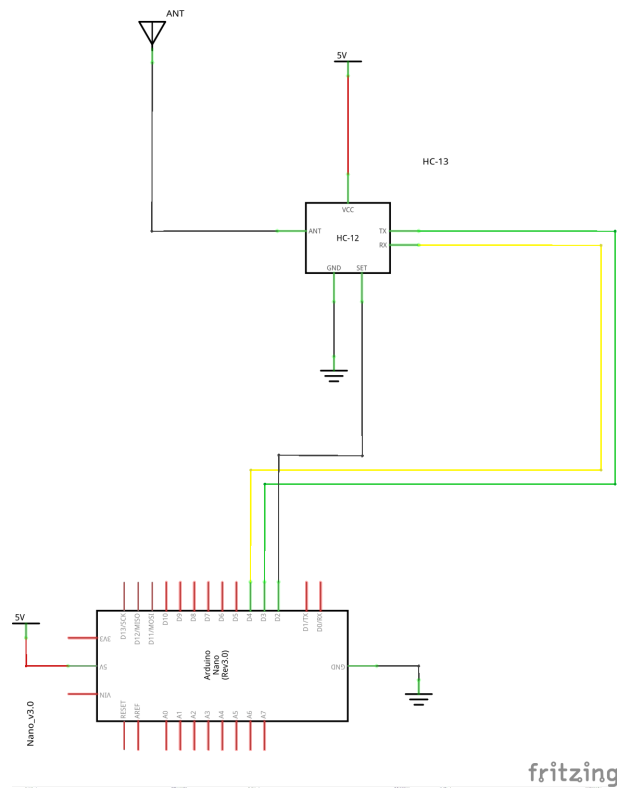


Figura 3.16: Schema elettrico completo Nodo dummy

Sulla base di queste configurazioni andiamo ad analizzare il comportamento dei due nodi soffermandoci sulle fasi principali, senza entrare nel dettaglio. Per fare ciò ci avvarremo di una FSM che mostra in maniera grafica i macro stati in cui si può trovare il nodo.

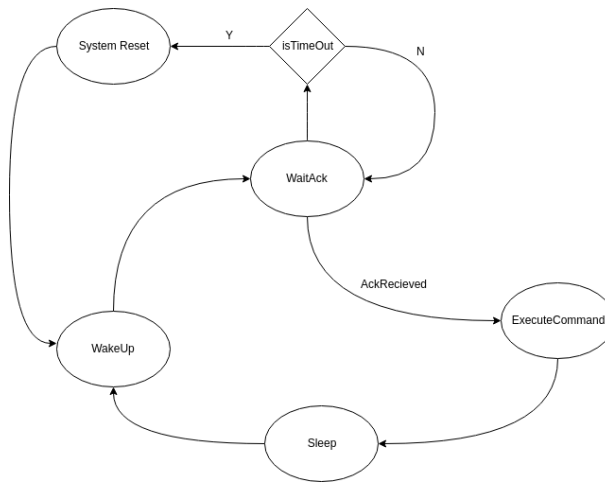


Figura 3.17: FSM dei nodi

Analizziamo ora i vari stati, dando enfasi alle funzionalità di ognuno.

WakeUp

Questa fase iniziale prevede il wakeUp del modulo HC12 e l'accensione del modulo con l'IRF520 che alimenta il gps, così da avere tutti i sistemi online. La gestione controllata del gps ovvia alla mancanza della modalità power-save di quest'ultimo permettendo una maggiore efficienza nei consumi. Quando tutti i sistemi sono attivi si passa alla fase successiva.

WaitAck

In questa fase il nodo aspetta l'ack del collettore a terra, tale ack consta in un JSON costituito da:

- **id** contenente il nome univoco del nodo
- **command** contenente il comando da inviare
- **value** contenente un eventuale parametro

Di seguito è possibile vedere un ack message inviato dal collettore:

```
{ "id": "ibiza#01", "command": "get_position", "value": "" }
```

All'arrivo di questo messaggio il nodo confronterà il proprio nome con quello presente nell'ack e, se questo corrisponde, si passerà allo stato successivo, altrimenti rimarrà in attesa fino a quando non riceverà un ack. Un altro caso potrebbe essere un Timeout del sistema. In questa fase il sistema non ha ricevuto un ack per un periodo di tempo elevato, procedendo al riavvio per ripristinare eventuali crash di sistema.

ExecuteCommand

Ricevuto l'ack, il sistema estrae dal JSON il comando e l'eventuale valore. Controlla se questo comando è gestibile ed in caso affermativo procede alla creazione del JSON con l'hash di controllo. Una volta compiuti questi passi procede con l'invio delle informazioni. Lo schema qui di seguito chiarifica quanto detto.

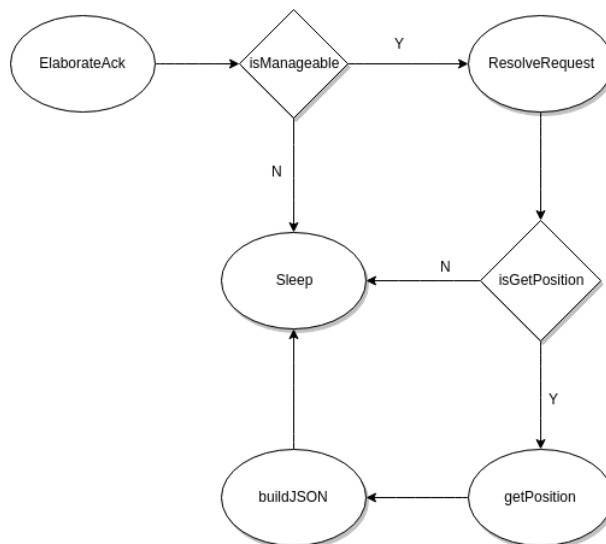


Figura 3.18: FSM della gestione dei comandi

Nel caso della richiesta della posizione avremo come risposta un JSON di questo tipo:

```

{
message:{id:"ibiza#01",latitude:"45.83214",longitudo:"13.34307"},
hash:"c6b913cd9cc47a95f1a34654c8d80596"
}

```

Il campo **message** rappresenta il JSON contenente il nome e la posizione del nodo mentre **hash** è un codice MD5 generato sul primo campo. Questa generazione è stata fatta tramite la libreria **MD5** di Vasilis Georgitzikis. Il collettore, ricevuto il messaggio, non dovrà far altro che generare a sua volta il codice dal campo message e compararlo con il campo hash. Se la procedura è andata a buon fine otterremo due codici hash uguali.

Sleep

In questa fase il modulo HC12 viene posto in sleepMode tramite l'apposito metodo presente nella libreria, mentre il GPS viene spento portando a LOW il pin che comanda il mosfet IRF520. Il GPS però viene riacceso in anticipo di un SAT_TIME rispetto all'accensione programmata. Questo è necessario poichè il GPS necessita di un preambolo per agganciarsi ai satelliti e questo tempo è impostato tramite il parametro SAT_TIME.

Prototipi Nodo

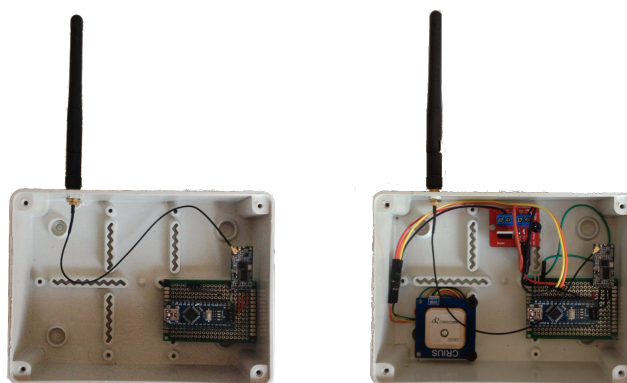


Figura 3.19: Sulla sinistra abbiamo il nodo dummy mentre sulla destra abbiamo il nodo dotato di GPS e sistema power save

Riportiamo inoltre le aree evidenziate in fase di progettazione.

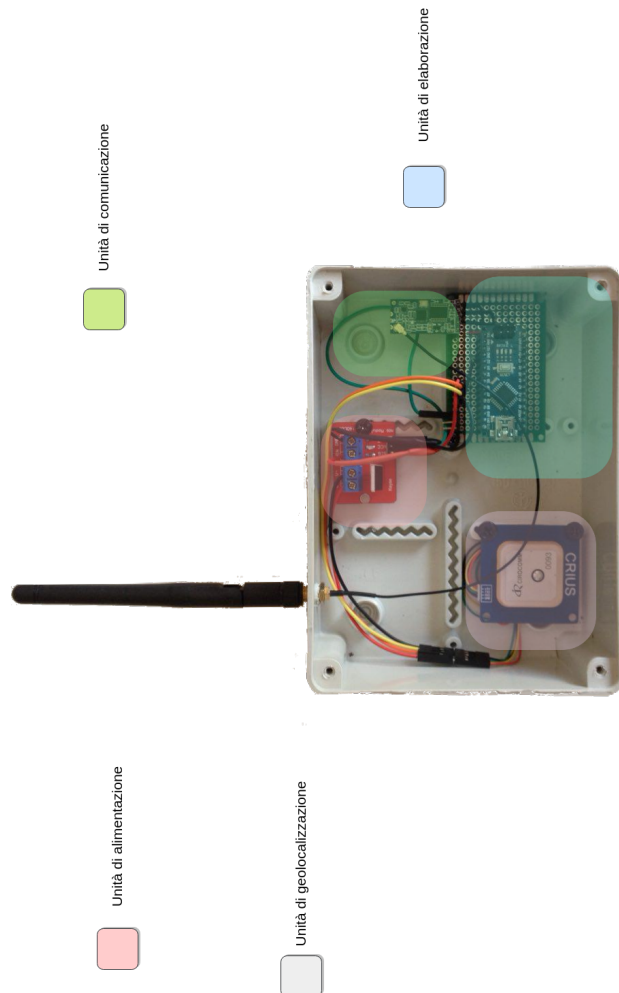


Figura 3.20: Schema delle macro unità

3.3.3 Progettazione e implementazione collettore

Anche se il collettore è l'unico collegamento con internet, noi ci soffermeremo soltanto sulla modellazione e costruzione di un dispositivo dotato di software minimale, in grado di gestire una comunicazione bidirezionale completa con controllo degli errori, demandando ad altri il compito di salvare, sincronizzare e visualizzare da remoto i dati raccolti. Tale dispositivo è composto da un'unità di elaborazione e da un'unità di comunicazione, quest'ultima uguale al nodo, ovvero composta da un HC-12 e da un'antenna omnidirezionale, mentre come unità di elaborazione si è preferito utilizzare un Raspberry Pi 3. Di seguito possiamo vedere lo schema di collegamento.

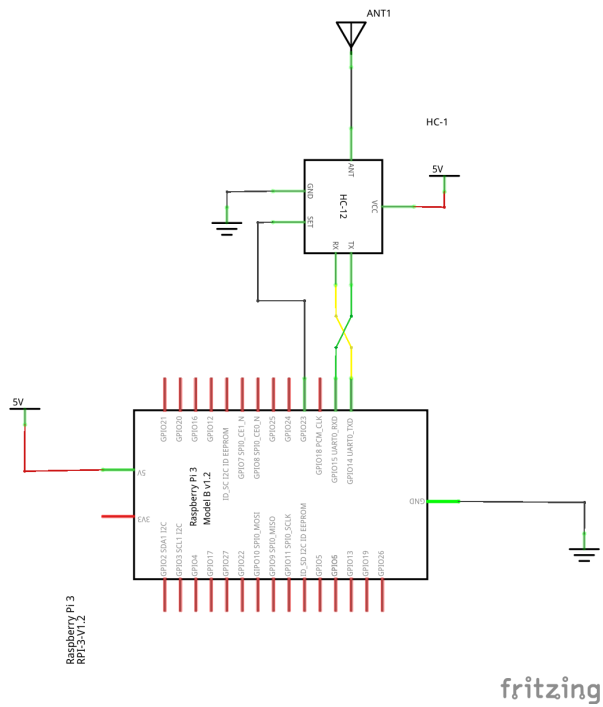


Figura 3.21: Schema elettrico collettore

Comunicazione Nodo - Collettore

Mostrate le due componenti principali, possiamo introdurre un primo schema che mostra la comunicazione tra il nodo ed il collettore:

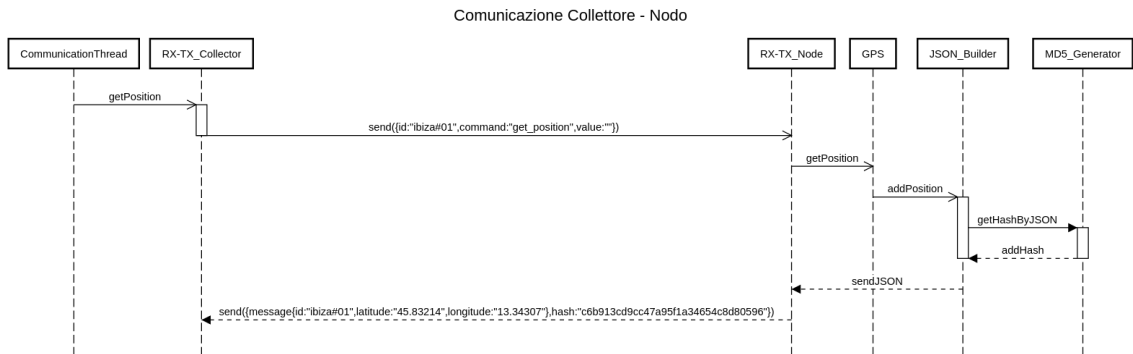


Figura 3.22: Diagramma di sequenza di una richiesta di posizione

Questo diagramma mostra le fasi principali, tralasciandone la composizione interna, che permettono di reperire la posizione del nodo. Ovviamente ci sono anche i messaggi di comando che permettono di settare il tempo di sleep dell'intero dispositivo oppure quello del gps, ma essendo il medesimo funzionamento trascureremo questi dettagli per mostrare in toto il sistema.

Gestione delle risposte

Una volta ricevuto il JSON questi viene mappato in un oggetto chiamato **HashedMessageJSON**, al suo interno contiene un altro oggetto chiamato **MessageJSON** che andrà a contenere il campo del JSON **message** mappato anch'esso. Per la mappatura da JSON ad oggetto si è utilizzata la libreria **Jackson** che attraverso annotazioni permette di linkare ogni campo del JSON con il campo dell'oggetto, anche se si tratta di un JSON innestato, come nel nostro caso. Per farlo basta predisporre l'oggetto con getter e setter di ogni campo annotato e lasciare un costruttore vuoto, inoltre è possibile aggiungere altre annotazioni che evitano errori in caso di mancanza di campi.

HashedMessageJSON e MessageJSON

```
public interface HashedMessageJSON {
    MessageJSON getMessage ();
    String getHash ();
    void setMessage (MessageJSON messageJSON);
    void setHash (String hash);
    String toJSON ();
}
```

Spicca subito un **getMessage()** di tipo `MessageJSON`. La potenza di Jackson infatti sta nel fatto che la sua mappatura può arrivare a molti livelli di profondità. È anche presente il metodo **toJSON()** che restituisce una stringa con il JSON validato a partire dall'oggetto ed il metodo **getHash()** che restituisce il contenuto del campo hash.

```
public interface MessageJSON {
    String getId ();
    String getLatitude ();
    String getLongitude ();
    void setId (String id);
    void setLatitude (String latitude);
    void setLongitude (String longitude);
    String toJSON ();
}
```

Qui invece possiamo notare, oltre al **toJSON()**, anche il **getId()** che nell'esempio avrebbe contenuto `ibiza#01`, mentre sarebbe pleonastico spiegare l'utilità degli altri getters.

Controllo errori

Mappati gli oggetti, è ora necessario controllare l'attendibilità degli stessi. Per farlo si è scelto di utilizzare la tecnica MD5 hash tramite la libreria `java.security.MessageDigest`. Di seguito mostreremo la parte di codice preposta al controllo del messaggio:

```
public boolean isCorrect (final HashedMessageJSON hMJ) {
    if (hMJ != null) {
        String mJ = hMJ.getMessage ().toJSON ();
        return hMJ.getHash ().equals (this.calculateHash (mJ));
    }
    return false;
}

public String calculateHash (final String msg) {
    MessageDigest m;
    try {
        m = MessageDigest.getInstance ("MD5");
        m.update (msg.getBytes (), 0, msg.length ());
        return new BigInteger (1, m.digest ()).toString (16);
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace ();
    }
    return "";
}
```

Il metodo `isCorrect()` compara la stringa hash del messaggio con un hash generato a partire dal messaggio innestato e, se ciò da esito positivo, possiamo dire con sufficiente certezza che il messaggio è corretto. Questa certezza è determinata dalla complessità dell'algoritmo di hashing. L'algoritmo MD5, sebbene sia ampiamente utilizzato ancora oggi, è sfavorito sul lato sicurezza, se confrontato con algoritmi più complessi come WHIRLPOOL, SHA-1 o RIPEMD-160. Si parla di sicurezza debole o forte negli algoritmi di hashing quando:

- **Sicurezza debole:** dato un messaggio M , è computazionalmente difficile trovare un secondo messaggio M' tale che $h(M)=h(M')$.
- **Sicurezza forte:** è computazionalmente "difficile" trovare una coppia di messaggi M, M' tali che $h(M)=h(M')$.

Le vulnerabilità del MD5

I seguenti casi sono stati estrapolati da Wikipedia [15]

1. Nel 1993 Der Boer e Bosselaers ottennero un primo risultato, anche se limitato, trovando una "pseudo-collisione" dell'algoritmo MD5; cioè due diversi vettori di inizializzazione I e J con 4 bit di differenza tali che:
2. $MD5compress(I,X) = MD5compress(J,X)$ Nel 1996 Dobbertin annunciò una collisione della funzione di compressione MD5. Anche se non rappresentava un attacco alla funzione hash MD5 completa, fu sufficiente per molti crittografi a raccomandare di passare ad un sostituto come il WHIRLPOOL, SHA-1 o RIPEMD-160.
3. La dimensione dell'hash a 128 bit era abbastanza piccola per contemplare un attacco birthday. Nel marzo 2004 ha avuto inizio il progetto distribuito MD5CRK con lo scopo di dimostrare che l'MD5 era un algoritmo insicuro, trovando una collisione e usando un attacco birthday.
4. MD5CRK durò fino a che, il 17 agosto 2004, fu trovata una collisione da Xiaoyun Wang, Dengguo Feng, Xuejia Lai e Hongbo Yu[1]. Il loro attacco su un cluster IBM p690 impiegò solo un'ora.
5. Il 1^o marzo 2005 Arjen Lenstra, Xiaoyun Wang e Benne de Weger dimostrarono[2] la costruzione di due certificati X.509 con differenti chiavi pubbliche e lo stesso MD5 hash, di fatto una collisione. La costruzione funzionava sia per le chiavi private che per quelle pubbliche. Dopo pochi giorni Vlastimil Klima descrisse[3] un algoritmo migliorato, capace di costruire collisioni MD5 in poche ore su un singolo computer. Il 18 marzo 2006, Klima pubblicò un algoritmo[4] che riusciva a trovare una collisione in un minuto su un singolo computer, usando un metodo che chiamò "tunneling".

La scelta di continuare ad utilizzare tale algoritmo sta nel fatto che è facilmente generabile anche da microcontrollori con hardware limitato e che le informazioni in gioco non sono sensibili.

Capitolo 4

Testing

4.1 Power Consumption

In questa fase compareremo i dati forniti dalle aziende produttrici con quelli degli assorbimenti reali. Naturalmente verrà preso in considerazione il solo elemento nodo, essendo l'unico ad essere alimentato a batteria.

Arduino Nano v3.0

In pieno funzionamento l'assorbimento dell'arduino nano v3.0 si attesta intorno ai 20mA o, se si rimuove il power led, a 17mA. Dalle misurazioni effettuate, risulta un assorbimento che oscilla tra 19.07mA e i 21.13mA, ne risulta che le caratteristiche rispecchiano appieno i parametri rilasciati dall'azienda.

HC-12

Dai dati forniti dall'azienda produttrice, risulta un assorbimento massimo in trasmissione di 200mA e 20 μ A in sleep. Testando il componente risultano \sim 23 μ A in sleep ed un assorbimento in trasmissione molto al di sotto delle aspettative, tale assorbimento infatti oscilla tra 152mA e 175,8mA a piena potenza, con configurazione:

- AT+C001 Canale 001 con frequenza di 433.4MHz
- AT+P8 Potenza massima 20dBm

CRIUS 0093 - GPS

Questo componente non possiede documentazione specifica quindi ci rifaremo alla documentazione del chip on-board ublox NEO-6. Nel datasheet a corredo è presente una vasta gamma di parametri, tra cui il power consumption. Tra le voci prenderemo in considerazione quello in modalità continua che è di 111 mW a 3.0V. A questo valore va aggiunta tutta la componentistica di protezione e di stabilizzazione presente sulla scheda Crius, che dovrebbe assorbire $\sim 10\text{mA}$ tenendo in considerazione led e regolatore di tensione, arrivando ad un assorbimento totale di $\sim 120\text{mA}$. I dati da noi rilevati sono molto vicini alle stime fatte, attendendosi intorno a quel valore con campionamenti che vanno da un minimo di 115,01mA ad un massimo di 119.21mA.

IRF520

Anche se è pressochè nullo l'assorbimento di questo componente, con valori che vanno da $25\mu\text{A}$ a $250\mu\text{A}$, verrà citato per completezza, considerato che il led on-board assorbe $\sim 3\text{mA}$.

Risultati

Sulla base dei dati raccolti, possiamo riassumere il tutto in una tabella riepilogativa dove, per valori reali, sono stati presi i valori massimi:

COMPONENT	THEORETICAL POWER CONSUMPTION (mA)	REALISTIC POWER CONSUMPTION (mA)
Arduino nano v3.0	20	21,13
HC-12	200	175,8
CRIUS 0093	120	190,21
IRF520	3	3
TOTAL	343	390,14

Figura 4.1: Tabella comparativa power consumption

Possiamo quindi definirci soddisfatti del risultato, se teniamo presente che avremo questi consumi solo in brevissimi periodi, cioè durante la fase di trasmissione. Questo tempo è calcolabile tenendo conto del baudrate, nel nostro caso 2400, e del tipo di codifica della porta che nel nostro caso, trattandosi di un arduino, sarà:

[1 bit Start] [8 bits payload] [1 bit stop]

Da qui deriviamo il bitrate della porta espresso con \mathbf{R} secondo la seguente formula:

$$R = \text{baudrate} * \log_2 S$$

Il logaritmo è in base 2, poichè abbiamo solo due livelli logici 0 e 1. La lettera \mathbf{S} rappresenta le possibili configurazioni:

$$S = 2^N$$

La lettera \mathbf{N} rappresenta i bit usati per la codifica, che nel nostro caso sono 10, calcolando header e trailer rispettivamente di 1 bit ed un payload di 8 bits. Da un baudrate pari a 2400 ne risulta un bitrate di 24Kbit per secondo, ora, calcolando un JSON di 200 caratteri (stima per eccesso) il tempo di trasmissione sarà meno di un decimo di secondo, mentre per il restante tempo il sistema assorbirà una corrente totale di $\sim 20\text{mA}$ (Arduino e HC-12 in sleep).

4.2 RF Interference Analysis

Questa sezione sarà dedicata all'affidabilità del sistema in presenza di fonti di interferenza nella banda UHF tramite un disturbatore creato ad hoc e posizionato tra nodo e collettore. Il disturbatore è un semplice modulo RF operante a 433Mhz che genera segnali casuali con periodo e baudrate variabili. Per apprezzare meglio il comportamento è stato montato un analizzatore di spettro tarato ad una sensibilità tale da dare maggior enfasi ai segnali che superano una certa potenza, permettendoci di studiare meglio l'interazione nodo collettore.

I test eseguiti sono stati:

1. Analisi senza fonti di disturbo
2. Analisi con fonte di disturbo con $\omega = 100\text{ms}$ e baudrate = 1200
3. Analisi con fonte di disturbo con $\omega = 50\text{ms}$ e baudrate = 2400
4. Analisi con fonte di disturbo con $\omega = 50\text{ms}$ e baudrate = 19200
5. Analisi con fonti di disturbo 3 e 4 in contemporanea

Parametri di comunicazione Nodo - Collettore

In questa parte verranno elencati i parametri di comunicazione quali tempi e numero di messaggi inviati da ambe due le parti.

Nodo:

- **WakeUp Time** illimitato fino all'arrivo dell'ack del collettore
- **SleepTime** 10sec
- **Numero di risposte** singola risposta

Collettore:

- **WakeUp Time** illimitato
- **SleepTime** 3sec
- **Numero di richieste** 3 richieste

Analisi senza fonti di disturbo

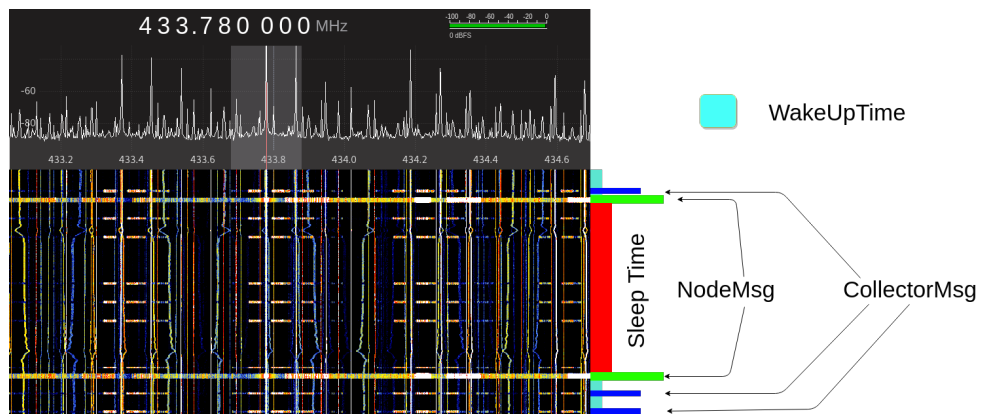


Figura 4.2: Analisi priva di disturbi

In questa immagine sono state aggiunte delle etichette per una maggiore chiarezza e per meglio comprendere quali sono le tempistiche nella comunicazione tra nodo e collettore. Essendo un pattern che si ripete, tali etichette verranno omesse nei test successivi commentando solo il comportamento e

i dati rilevati. In questo caso abbiamo la catena di chiamate `getPosition` da parte del collettore a cui viene data una risposta da parte del nodo ben visibile e netta, questo perchè al nodo è stata montata un'antenna con un guadagno da 5db ed è configurato per trasmettere a piena potenza. Il collettore invece trasmette alla minima potenza ed è equipaggiato con un antenna a spirale, questo per simulare distanze maggiori.

In questo test sono arrivati 97 messaggi corretti su 100.

Analisi con fonti di disturbo singole

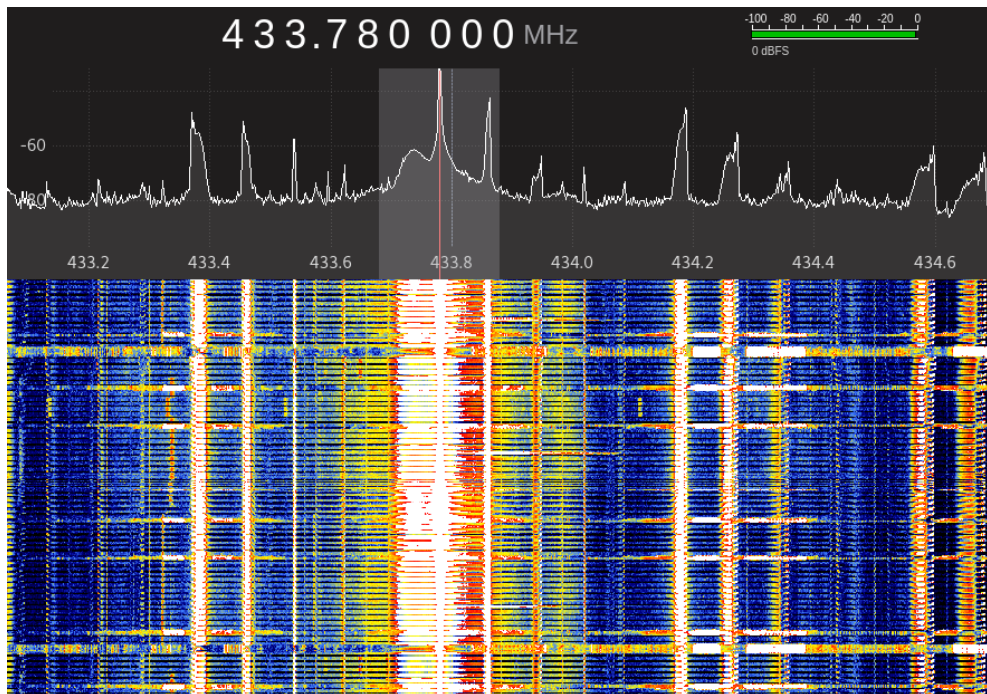


Figura 4.3: Analisi con fonte di disturbo con $\omega = 100\text{ms}$ e baudrate = 1200

In questo test è subito evidente la fonte di disturbo ma è altrettanto visibile il solito pattern di invio-ricezione, tale test infatti ha ottenuto un ottimo risultato non molto distante dal primo.

In questo test sono arrivati 94 messaggi corretti su 100.

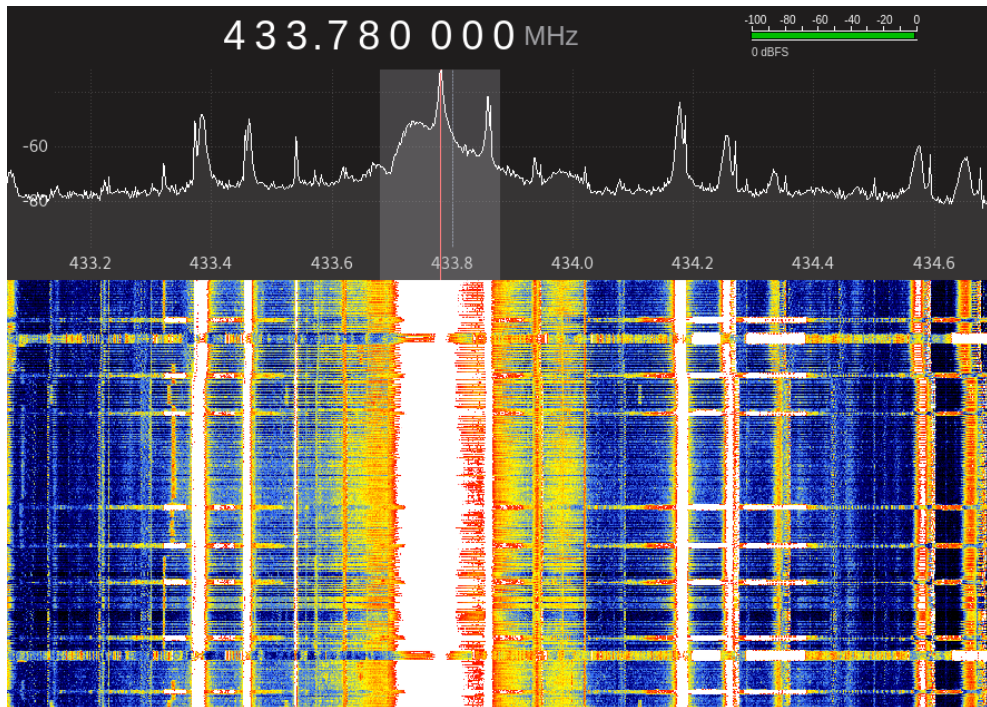


Figura 4.4: Analisi con fonte di disturbo con $\omega = 50\text{ms}$ e baudrate = 2400

In questo test, che come vediamo non si differenzia molto dal precedente, possiamo notare una catena di impulsi di disturbo più fitta, ciò era auspicabile visto il passaggio da 100ms a 50ms.

Qui però il baudrate è uguale al nostro sistema e questo ha abbassato di molto il numero di messaggi corretti.

In questo test sono arrivati 85 messaggi corretti su 100.

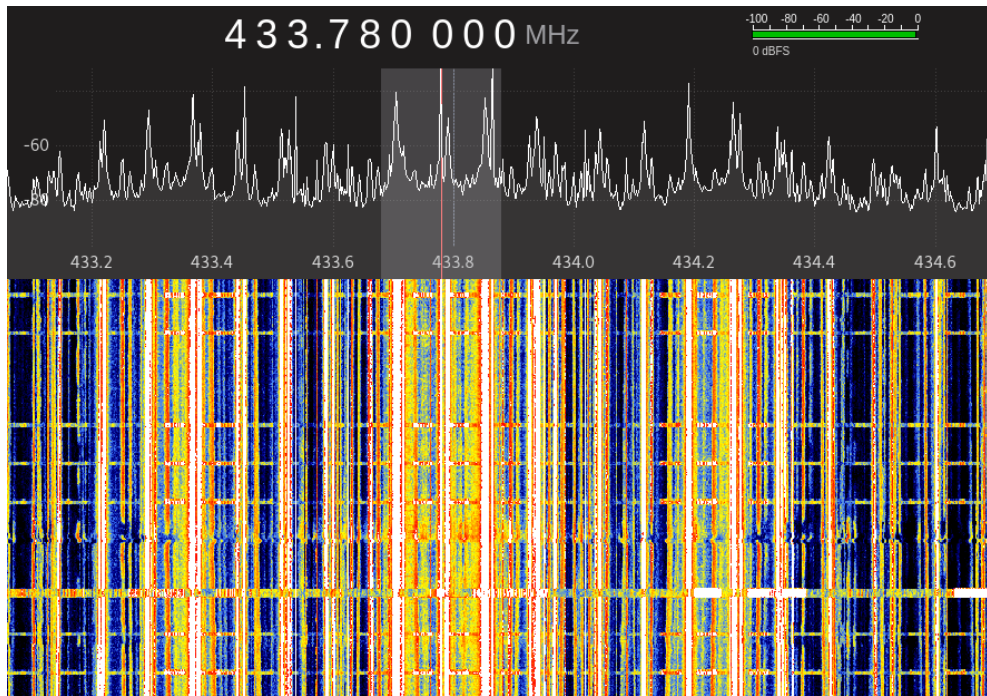


Figura 4.5: Analisi con fonte di disturbo con $\omega = 50\text{ms}$ e baudrate = 19200

In questo test sono state inviate catene di impulsi con un baudrate elevato (questi parametri sono simili a quelli presenti nei radio comandi per cancelli) ma con una potenza di trasmissione molto maggiore.

In questo test sono arrivati 71 messaggi corretti su 100.

Analisi con fonti di disturbo multiple

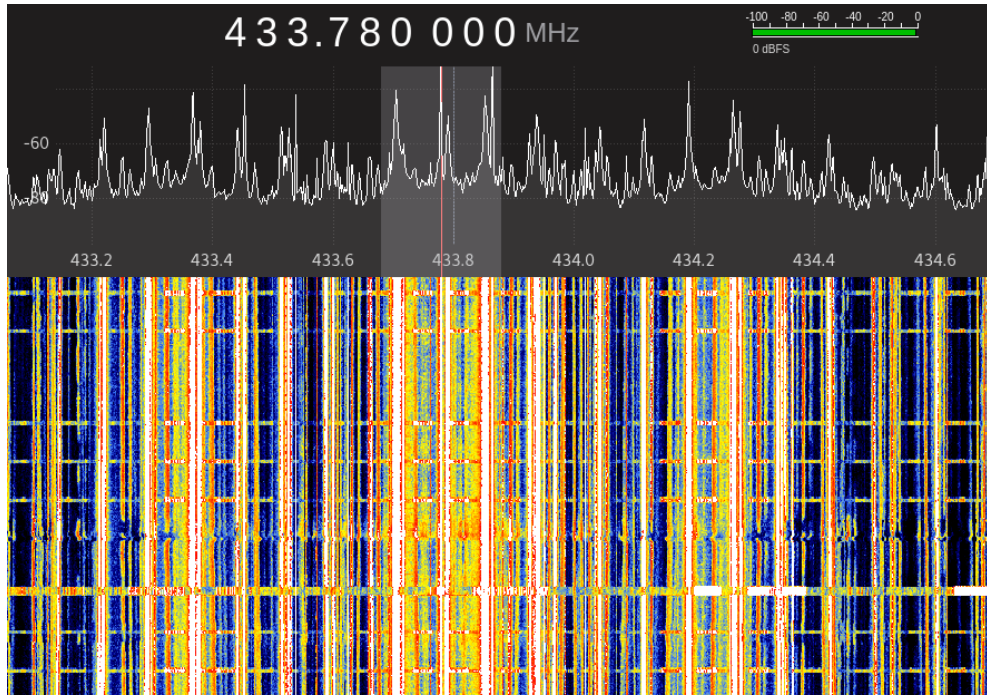


Figura 4.6: Analisi con fonti di disturbo $\omega = 50\text{ms}$ e baudrate = 2400 e $\omega = 50\text{ms}$ e baudrate = 19200

In questo test si è estremizzata la fonte di disturbo combinando i due casi precedenti, al fine di testare il sistema nelle condizioni peggiori. Il risultato infatti non si è fatto attendere con solo 62 messaggi corretti su 100.

4.3 OTA transmission

Tale sezione sarà dedicata al test di trasmissione in campo aperto, monitorando la qualità e l'affidabilità del sistema. Per farlo è stata scelta come località marittima Bellaria-Igea Marina-

Sia sul nodo che sul collettore sono state usate le seguenti configurazioni.

- **Antenna** omnidirezionale da 5db
- **Baudrate** 2400
- **Potenza trasmissione** massima consentita dal modulo, 20db
- **Canale 001** con frequenza 433.70Mhz

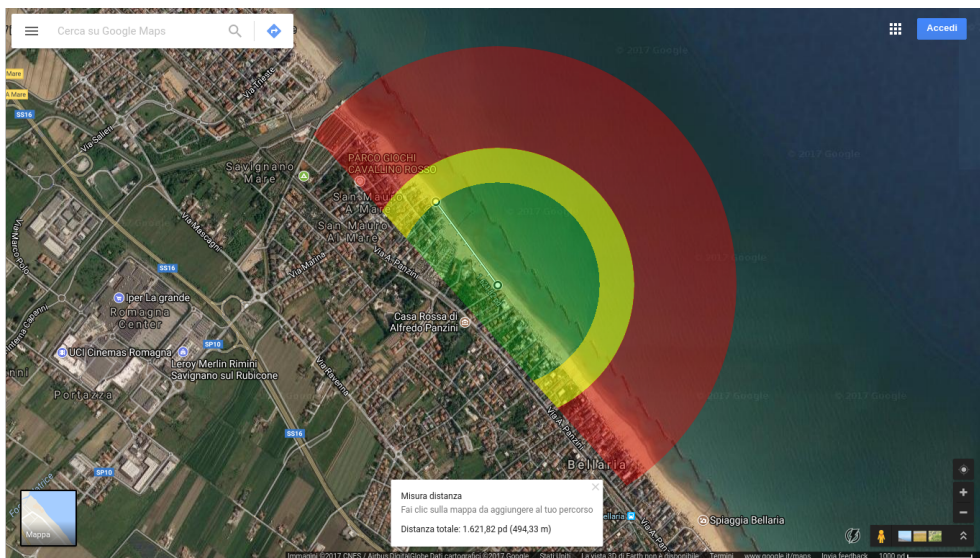


Figura 4.7: Trasmissione OTA con monitoraggio del segnale

Sulla mappa sono state segnate le aree in cui ‘e stato eseguito il test, colorandole di colori differenti in base alla percentuale di messaggi correttamente ricevuti.

I dati relativi a tale aree sono i seguenti:

- **Verde:** 100%
- **Giallo:** 30%
- **Rosso:** 5%

Dai dati rilevati possiamo notare come ci sia un netto cambiamento tra la zona verde e le zone più distanti, passando da una percentuale di messaggi corretti pari a $\leq 100\%$ ad una percentuale ben più bassa che ci porta ad ammettere l’inaffidabilità del sistema superati i 500 metri circa.

Capitolo 5

Conclusioni e sviluppi futuri

Il presente capitolo contiene un resoconto del lavoro fatto, nella stesura di quest'ultimo si è cercato di essere quanto più obiettivi possibile. Verranno inoltre presentati i punti su cui attualmente ci si sta concentrando e quelli su cui è possibile migliorare.

5.1 Resoconto

Dai test svolti è emerso fin da subito quanto sia delicato il problema dei consumi e quanto sia complesso ridurre questi ultimi mantenendo caratteristiche performanti. Nel nostro caso abbiamo dei nodi che in sleep assorbono meno di 20mA e che in piena trasmissione superano di poco i 170mA, raggiungendo una distanza di trasmissione di 500 metri circa.

Tale risultato ci rende molto soddisfatti del lavoro svolto.

5.2 Nuove sfide

Attualmente si sta già lavorando sull'apparato di alimentazione dotando il sistema di un pannello solare di piccole dimensioni e di un real time clock che spegne automaticamente il nodo nelle ore della notte, sincronizzandosi automaticamente tramite il modulo gps. È anche in sviluppo un algoritmo che riconosce le imbarcazioni in acqua calcolandone automaticamente le corse e adattandosi automaticamente al variare delle maree. Altro tema al centro di dibattiti è la distanza. L'obiettivo del gruppo è infatti quello

di raggiungere il miglio marino e per farlo si sta considerando di sostituire l'antenna del collettore, passando da un'antenna omnidirezionale ad una a pannello calcolando che, salvo rari casi, l'apertura del fascio d'antenna 'e inferiore a 180° .

Tanti passi sono stati fatti ma molti di più se ne potranno fare se la collaborazione tra uomini continuerà ad essere il carburante della ricerca, come lo è stato per questo progetto, sperando che questo sia solo l'inizio di un viaggio pieno di scoperte.

“Nessun uomo dovrebbe sfuggire
alle nostre università senza
sapere quanto poco conosce.”

Julius Robert Oppenheimer

Ringraziamenti

Inzierò ringraziando la mia compagna Enrica a cui debbo non solo questo traguardo ma tutte le cose belle che da sette anni mi regala e che solo lei mi ha saputo dare. Seguono i miei colleghi e amici Davide ed Elia con cui ho svolto questo progetto tra risate e momenti bui e a tutti gli altri amici che mi hanno supportato e sopportato tra mille peripezie e sbalzi di umore. Ma un grazie particolare va al mio relatore che grazie alle sue lezioni e i suoi seminari dà, giorno dopo giorno, un grosso contributo alla scienza non solo come ricercatore ma anche come divulgatore per chi non è addetto ai lavori. Per finire ringrazio tutta l'Università di Bologna che, da i professori allo staff tecnico e amministrativo, fa crescere professionisti in tutti gli ambiti accogliendoli e accompagnandoli fino alla laurea.

Bibliografia

- [1] Vittorio Carlini. Borse: così social, algoritmi e intelligenza artificiale sono le mani invisibili dei mercati. *Il Sole24Ore*, 17 Marzo 2017.
- [2] Guida all'uso dell'applicazione greenway . <http://www.sardegnaicerche.it>, 07 mar 2017.
- [3] LoRa Technology, accessed 07/09/2017. <https://www.lora-alliance.org/technology>.
- [4] LoRa System Scheme, accessed 07/09/2017. <http://blog.microtronics.at/lora-and-2g-in-one-module>.
- [5] LoRa Coverage Scheme, accessed 07/09/2017. <https://www.lora-alliance.org>.
- [6] SigFox Architecture, accessed 09/09/2017. <http://cz.iot-nn.com/blog/2017/06/27/sitove-technologie-lpwan-pro-internet-veci-5-dil>.
- [7] Global System for Mobile Communications, accessed 07/09/2017. https://it.wikipedia.org/wiki/Global_System_for_Mobile_Communications.
- [8] U-Blox, accessed 07/09/2017. <https://www.u-blox.com/en/press-release/u-blox-launches-low-power-version-2g-m2m-module-europe-and-asia>.
- [9] Ultra high frequency, accessed 07/09/2017. https://it.wikipedia.org/wiki/Ultra_high_frequency.

-
- [10] Arduino, accessed 07/09/2017. [https://it.wikipedia.org/wiki/Arduino_\(hardware\)](https://it.wikipedia.org/wiki/Arduino_(hardware)).
 - [11] Arduino nano v3.0, accessed 09/09/2017. <https://store.arduino.cc/arduino-nano>.
 - [12] Raspberry, accessed 10/09/2017. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b>.
 - [13] Modulo 433, accessed 12/09/2017. <http://www.buildcircuit.com/how-to-use-rf-module-with-arduino>.
 - [14] Hc12, accessed 12/09/2017. <http://www.14core.com/wiring-the-hc11-hc12-434433mhz-transceiver>.
 - [15] MD5, accessed 15/09/2017. <https://it.wikipedia.org/wiki/MD5>.