

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

SCUOLA DI INGEGNERIA E ARCHITETTURA · SEDE DI CESENA  
DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE  
- GUGLIELMO MARCONI -

CORSO DI LAUREA IN  
INGEGNERIA ELETTRONICA PER L'ENERGIA E L'INFORMAZIONE

# INTERFACCIAMENTO DI SISTEMI RFID E UWB

TESI DI LAUREA IN  
COMUNICAZIONE DIGITALE E INTERNET

RELATORE:  
**CHIAR.MO PROF. ING.  
DAVIDE DARDARI**

CORRELATORE:  
**ING.  
DAVIDE FABBRI**

PRESENTATA DA:  
**LUCA ARCANGELONI**

I APPELLO - II SESSIONE  
ANNO ACCADEMICO 2018/2019



**Parole chiave:**

RFID

UWB

EPC

Tag

Reader

Sincronizzazione

Trigger



**Dedica**

*Alla mia famiglia...*



## Abbreviazioni:

<b>AI</b>	Antenna Inventory
<b>CW</b>	Continuos Wave
<b>EPC</b>	Electronic Product Code
<b>FCC</b>	Federal Communications Commission
<b>FDX</b>	Full Duplex
<b>FOV</b>	Field Of View
<b>FPGA</b>	Field Programmable Gate Array
<b>GPIO</b>	General Purpose Input/Output
<b>HDX</b>	Half Duplex
<b>ISO</b>	International Organization for Standardization
<b>LLRP</b>	Low Level Reader Protocol
<b>NFC</b>	Near Field Communication
<b>OOK</b>	On-Off Keying
<b>PMU</b>	Power Management Unit
<b>RFID</b>	Radio Frequency IDentification
<b>RO</b>	Reader Operation
<b>RTLS</b>	Real Time Locating System
<b>SNR</b>	Signal to Noise Ratio
<b>SRD</b>	Short-range devices
<b>TCP</b>	Transmission Control Protocol
<b>TDOA</b>	Time Difference Of Arrival
<b>TOA</b>	Time Of Arrival
<b>UHF</b>	Ultra High Frequency
<b>UWB</b>	Ultra Wide Band
<b>WRU</b>	Wake-Up Receiver





# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Tecnologie UWB e RFID</b>	<b>3</b>
2.1	Tecnologia UWB . . . . .	6
2.2	Soluzione UWB-RFID . . . . .	6
<b>3</b>	<b>Sistema UWB-RFID</b>	<b>9</b>
3.1	Architettura . . . . .	9
3.2	Tag . . . . .	12
3.3	Sistema di localizzazione UWB . . . . .	14
3.4	Reader RFID . . . . .	17
<b>4</b>	<b>EPCglobal Class1 Generation2</b>	<b>19</b>
4.1	Low Level Reader Protocol . . . . .	22
4.1.1	Messaggi LLRP . . . . .	24
<b>5</b>	<b>Sincronizzazione</b>	<b>29</b>
5.1	Generazione del trigger tramite LLRP . . . . .	32
5.1.1	Risultati e Problemi . . . . .	35
5.2	Generazione del trigger tramite seriale . . . . .	38
5.2.1	Risultati e Problemi . . . . .	44
<b>6</b>	<b>Test finale</b>	<b>49</b>
6.1	Accorgimenti . . . . .	54
6.2	Risultati sperimentali . . . . .	55

X

*INDICE*

**7 Conclusioni**

**59**

**Bibliografia**

**67**

# Capitolo 1

## Introduzione

La tesi qui introdotta si inserisce in un lavoro molto più ampio nato da un primo progetto finanziato dall’Agenzia spaziale europea: LOST (Localisation of objects in space through RF Tags, [8]). L’obiettivo di tale progetto consisteva nella realizzazione di un sistema di localizzazione con precisione centimetrica per aiutare gli astronauti in assenza di gravità a individuare gli oggetti sparsi nella stazione spaziale. A fronte di ciò è stata utilizzata la tecnologia radio a banda ultra larga (ultrawide bandwidth - UWB) che non è nuova nell’impiego in sistemi di localizzazione ad elevata precisione per ambienti chiusi, dove il GPS non funziona. Il problema però è che la tecnologia UWB funziona grazie a Tag a radiofrequenza equipaggiate con batterie che andrebbero ricaricate dopo qualche ora di funzionamento, procedura che si sarebbe rilevata molto complicata e dispendiosa all’interno della stazione spaziale internazionale. La soluzione adottata, che ha dato vita al sistema alla base di questa tesi, fu quella di sostituire il generatore di impulsi UWB nel Tag con un circuito che riesce a modulare la riflessione dei segnali ricevuti dall’antenna, si parla di modulazione a retro-diffusione (backscatter modulation), in questo modo l’energia consumata cala drasticamente ed è possibile eliminare la batteria trasferendo l’energia necessaria via radio dal dispositivo di interrogazione (Reader). Nei capitoli successivi tale sistema verrà approfondito spiegandone il funzionamento complessivo, dall’energizzazione dei Tag alla loro localizzazione.

Successivamente al progetto LOST nasce l’esigenza di standardizzare il si-

stema realizzato sostituendo i protocolli di indirizzamento concepiti ad hoc per il suo funzionamento con protocolli standard utilizzati dalle etichette elettroniche (Tag) presenti in commercio sotto il nome di sistemi RFID (radiofrequency identification). In altre parole un lavoro di integrazione del sistema UWB per la gestione del posizionamento ad alta precisione con il protocollo RFID Gen2, operante nella banda UHF al fine di garantire una retro-compatibilità con i Reader standard esistenti in modo da assicurarne l'utilizzo in contesti aziendali e commerciali. Le applicazioni possono essere le più varie, dalla gestione logistica di magazzini alla localizzazione dei bagagli dei passeggeri negli aeroporti.

Il lavoro di tesi prende parte a quest'ultima fase di standardizzazione ponendosi l'obiettivo di sincronizzare il sotto-sistema UWB di localizzazione con il sotto-sistema RFID. In particolare il ricevitore è il dispositivo che capta gli impulsi UWB retro diffusi dal Tag e deve sapere quando quest'ultimo è attivo e pronto a riflettere per aprire la finestra di campionamento, così come anche il generatore che genera la sequenza di interrogazione costituita da impulsi UWB necessita di questa informazione in modo da iniziare a trasmettere impulsi solo a Tag acceso. Ciò rende indispensabile una sincronizzazione tra questi dispositivi e il Reader RFID responsabile dell'energizzazione e indirizzamento del Tag quindi l'unico a sapere quando è attivo o meno.

La tesi si svilupperà in una prima parte strettamente teorica in cui si approfondiranno i vari protocolli utilizzati, dal protocollo RFID ad altri più pratici e di basso livello come LLRP e l'EPC GEN 2, inoltre verrà esaminato il sistema in esame spiegandone il funzionamento generale. A questa prima parte seguirà la seconda dove verrà illustrato l'aspetto pratico della tesi descrivendo le varie soluzioni adottate con i relativi risultati e problemi.

# Capitolo 2

## Tecnologie UWB e RFID

Negli ultimi anni i sistemi di identificazione a radio frequenza (RFID) sono diventati comuni in diversi ambiti, dall'industria manifatturiera alla logistica, sostituendo procedure di identificazione automatiche più datate come, per esempio, le etichette a codici a barre ormai diventate inadeguate a causa della bassa capacità di archiviazione e alla non riprogrammabilità. I sistemi RFID consentono l'identificazione univoca di oggetti, animali o persone tramite l'uso di due componenti: Transponder (Tag) e Reader, Fig. 2.1. Il Tag, posizionato nell'oggetto da identificare, memorizza informazioni (ID univoco, dati di varia natura) che il Reader può leggere/scrivere. Il vantaggio di tali sistemi è che per il trasferimento di potenza e dati tra i due componenti non è richiesto il contatto, ma si adoperano campi elettromagnetici.

I Tag RFID più economici sono passivi in cui l'energia necessaria per la comunicazione Tag-Reader viene prelevata dal segnale di interrogazione del Reader o dall'ambiente circostante e l'informazione viene trasmessa attraverso segnali retro-diffusi (backscatter modulation).

Nei nuovi scenari imposti dall'Internet of Things, però, i sistemi industriali e dei consumatori avranno bisogno non solo di identificare un oggetto, ma anche di localizzarlo usando Tag energeticamente autonomi, economici e sostituibili. Per soddisfare questi requisiti l'adozione di sistemi passivi ultra-wideband (UWB) radio-frequency identification (RFID) sembra la soluzione promettente per superare i limiti dell'attuale standard RFID-Gen.2 [7].

Da una parte i sensori autonomi e non invasivi potrebbero essere usati

in ambito biomedicale, per monitorare i farmaci al fine di una gestione ospedaliera efficiente o nella catena alimentare per prevenire il rischio di contraffazione di cibo, dall'altra parte, la capacità di localizzazione in tempo reale con Tag ad alta precisione (pochi centimetri) risulterebbe utile in contesti di ricerca articoli logistici e commerciali. Si potrebbe immaginare in futuro un contesto in cui ogni oggetto sarà taggato per prendere parte ad applicazioni basate sulla realtà aumentata che consentirebbe alle immagini virtuali di sovrapporsi a quelle fisiche in tempo reale come illustrato in Fig. 2.2.

Queste prospettive evidenziano i principali limiti della reale tecnologia HF e UHF-RFID inizialmente concepita come già detto per sostituire i codici a barre e quindi solo per l'identificazione. Le limitazioni consistono nell'assenza di sensori sui Tag, mancata capacità di localizzazione, limitato campo operativo (inferiore ai 10 metri per Tag UHF, al di sotto di 1-2 metri per gli HF) e assenza di integrazione con gli standard di comunicazione mobile. Nel corso degli anni alcune di queste limitazioni sono state superate attraverso l'introduzione di miglioramenti degli standard come l'aggiunta di sensori sui Tag per monitorare parametri fisici, l'integrazione di Reader HF negli smartphone (standard NFC) e la disponibilità di alcune soluzioni che offrono una localizzazione approssimativa. Sfortunatamente questi miglioramenti hanno generato conseguenze non volute: ulteriore riduzione dell'intervallo operativo (ad esempio, gli smartphone NFC possono leggere Tag fino a 10-20 cm), Reader più complessi (array di antenne).

In conclusione è chiaro che i requisiti sopra menzionati non possono essere soddisfatti dall'attuale prima e seconda generazione RFID. Al giorno d'oggi la localizzazione ad alta precisione è fornita da tecnologie distinte come reti di sensori wireless (wireless sensor network - WSN) e sistemi di localizzazione in tempo reale (real-time locating system - RTLS), per cui al fine di soddisfare i requisiti c'è la necessità di unire i due mondi fondendo la tecnologia ultra-wideband, finora utilizzata nei sistemi RTLS, con il principio di modulazione a retro-diffusione, che è alla base dei sistemi UHF-RFID [7, 1].

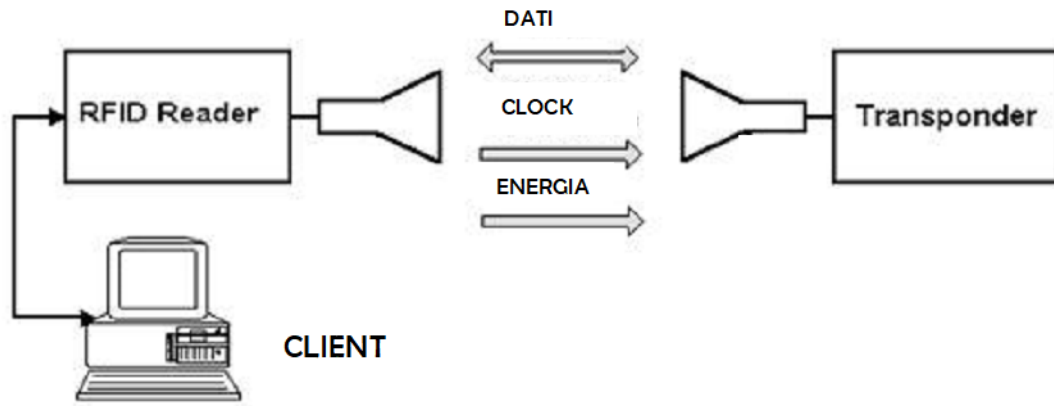


Figura 2.1: Architettura RFID

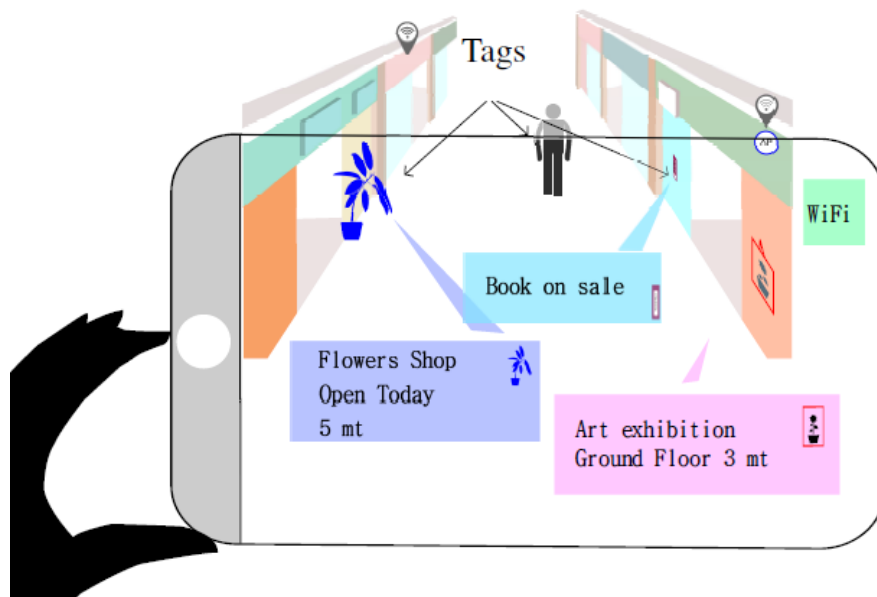


Figura 2.2: Possibile scenario futuro: un utente con il suo smartphone può localizzare e interagire con gli oggetti taggati presenti nell'ambiente [7]

## 2.1 Tecnologia UWB

In accordo con la definizione della Federal Communications Commission (FCC) ogni segnale caratterizzato da una larghezza di banda superiore ai 500 MHz è considerato UWB indipendentemente dal modo in cui viene generato. Il metodo classico e più semplice per ottenere segnali UWB è attraverso impulsi radio UWB (IR-UWB). In un sistema IR-UWB si genera una sequenza di brevi impulsi (tipicamente dalla durata di 1 ns) per bit di informazione al fine di garantire una risoluzione precisa del tempo di arrivo del segnale (TOA, time of arrival) che permette una maggiore accuratezza negli algoritmi di localizzazione basati sulla misura del tempo di propagazione dell'onda radio. La tecnologia UWB è attualmente usata come base di partenza nei sistemi RTLS a corto raggio ad alte prestazioni che utilizzano Tag attivi alimentati da batterie.

## 2.2 Soluzione UWB-RFID

Unire l'uso delle tecnologie RFID e UWB, come già detto, rappresenta una soluzione interessante per fondere i vantaggi dei sistemi RFID e RTLS. Alcuni progetti hanno analizzato queste opzioni specialmente considerando schemi UWB-RFID attivi, ma i rigidi vincoli sul consumo di energia richiedono soluzioni basate su Tag passivi con modulazione backscatter. La modulazione a retrodiffusione è basata sull'idea che le proprietà di riflessione dell'antenna del Tag dipendano dai dati da trasmettere. Questo principio è già ampiamente usato nei tradizionali Tag passivi UHF-RFID basati su segnali sinusoidali continui (continuous wave - CW) per mandare dati dal Tag al Reader. La stessa idea è stata adottata usando segnali di interrogazione UWB piuttosto che CW. Quando un treno di impulsi UWB investe l'antenna del Tag, viene in parte riflesso indietro (retro diffuso) in funzione della configurazione dell'antenna. La risposta retro diffusa è costituita da due modi, uno strutturale e uno dipendente dall'antenna. Il modo strutturale dipende dalla forma e dai materiali dell'antenna ed è indipendente dal carico della stessa, mentre l'altro modo è funzione del carico dell'antenna, per esempio quando l'antenna è cortocircuitata tale



impulso è caratterizzato da un'inversione di segno rispetto alla condizione di carico aperto. Questa proprietà può essere sfruttata per stabilire un collegamento tra Tag e Reader. Inoltre stimando il round-trip time (RTT) del segnale riflesso e misurando il suo TOA rispetto al tempo del segnale di interrogazione è possibile dedurre la distanza tra Reader e Tag con elevata precisione. Sfortunatamente i segnali diffusi dall'ambiente circostante e il modo strutturale dell'antenna prevalgono nel segnale ricevuto dal Reader, rendendo così il rilevamento del modo di antenna (che trasporta i dati) uno dei problemi principali negli UWB-RFID passivi. Nel corso della seguente tesi non ci si occuperà di queste problematiche se non solamente dal punto di vista teorico al fine di inquadrare il contesto nel quale si inserisce la vera attività di tesi già introdotta. Maggiore dettagli si possono trovare [9] riferimento bibliografico.



# Capitolo 3

## Sistema UWB-RFID

### 3.1 Architettura

L'architettura generale del sistema implementato nel progetto LOST è mostrata nella Fig. 3.1. Il sistema comprende un'unità centrale, dei nodi di riferimento (trasmettitori e ricevitori) e dei Tag. Ogni nodo comunica con l'unità centrale tramite collegamenti cablati o attraverso standard wireless per il rilevamento del trasferimento di informazioni e misure della distanza. Lo stesso link è usato dall'unità centrale per controllare e sincronizzare i vari nodi per scopi di calibrazione e sono proprio questi collegamenti l'oggetto della tesi.

Nel caso d'interesse, inoltre, si è usata una configurazione multi-statica consistente in un trasmettitore e più ricevitori, il che si traduce in una maggiore copertura. Si presume che non ci sia un riferimento temporale comune tra i ricevitori e il trasmettitore, quindi un tempo di arrivo TOA assoluto non è disponibile. Per questo si adopera come parametro negli algoritmi di localizzazione la differenza dei tempi di arrivo (time-difference-of-arrival -TDOA).

I Tag usati richiedono un interruttore RF e un controller, quindi non è una soluzione senza chip. Tuttavia, a causa del consumo energetico dei Tag estremamente basso, la potenza può essere fornita in modalità wireless dal Reader. Tutto ciò richiede un canale separato per esempio nella banda UHF, in quanto i livelli di trasmissione in UWB sono severamente limitati da

vincoli normativi. Inoltre la trasmissione nella banda UHF può essere usata per operazioni di programmazione dei Tag in real-time come per esempio l'indirizzamento. A tal proposito nella Fig. 3.2 è riportata una architettura più fedele del sistema dove vengono separati i due collegamenti UHF e UWB. Nel seguito viene spiegato per punti il funzionamento generale del sistema:

- Reader alimenta e indirizza il Tag utilizzando segnali nella banda UHF, 868 MHz in Europa secondo il protocollo Gen2;
- Reader deve avvertire l'oscilloscopio e il generatore di segnali che il Tag è pronto utilizzando un segnale di sincronismo, deve inoltre inviare all'oscilloscopio l'indirizzo del Tag energizzato per permettergli di capire a quale Tag si riferiscono i segnali che riceverà;
- Il trasmettitore inizia a trasmettere impulsi UWB attraverso un generatore di segnali;
- Il Tag attraverso uno switch riflette gli impulsi UWB ricevuti;
- Tali impulsi riflessi vengono captati dal ricevitore, elaborati e inviati all'oscilloscopio che attraverso algoritmi di localizzazione mostra a video la posizione del Tag.

Tutto questo deve avvenire nella finestra di attività del Tag, quando è in grado di mettere in atto la modulazione a backscattering andando a variare il carico dell'antenna grazie all'interruttore.

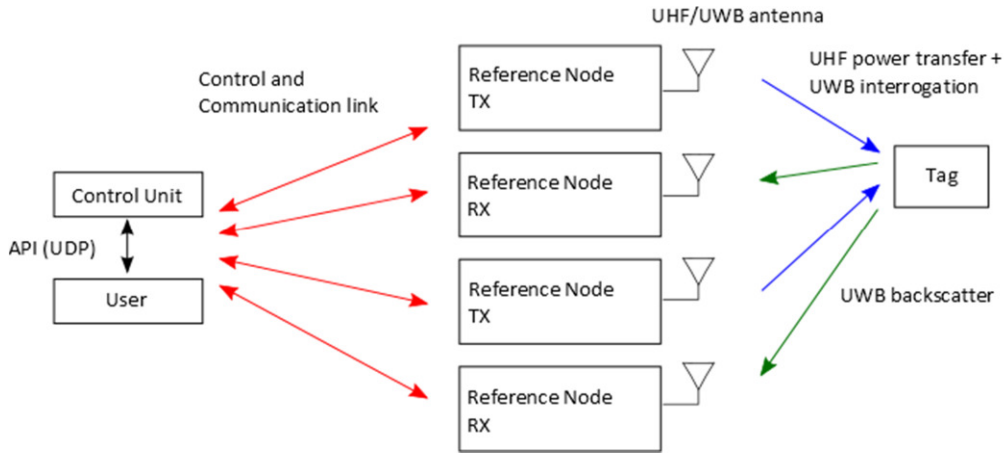


Figura 3.1: Architettura del sistema generalizzata e multi-statica

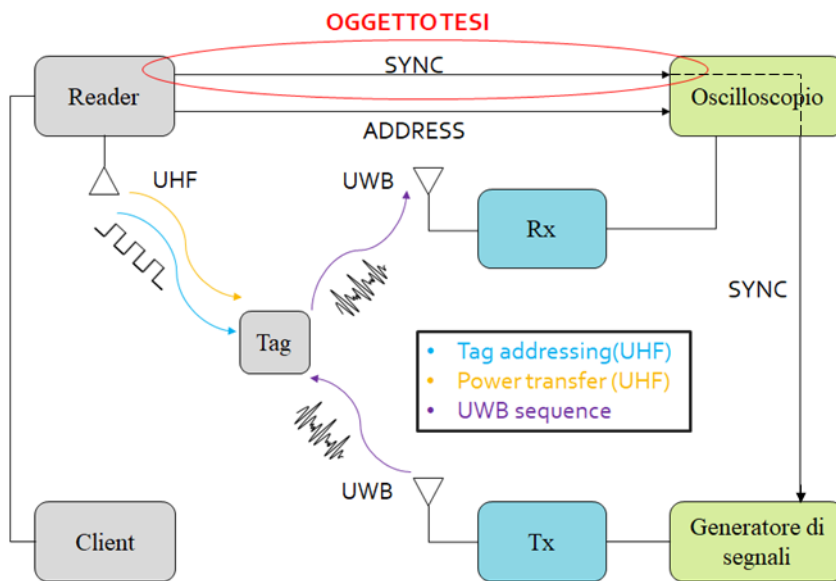


Figura 3.2: Architettura del sistema nel caso monostatico presa come riferimento in laboratorio

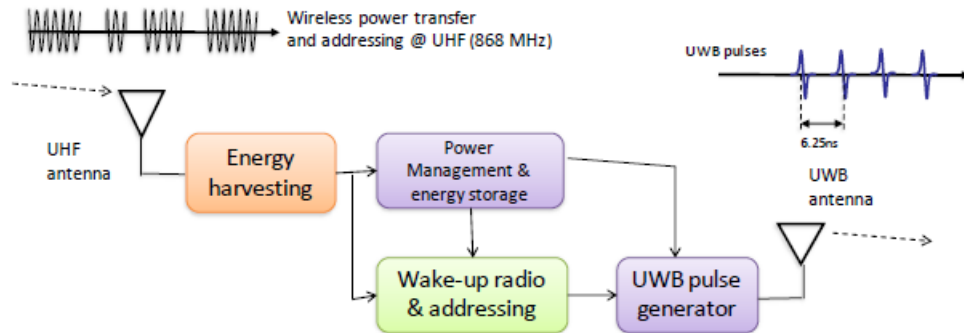


Figura 3.3: Schematico a blocchi del UWB/UHF RFID Tag [8]

## 3.2 Tag

Nel sistema in esame, come già detto c'è la necessità di avere Tag energeticamente autonomi ed economici cioè passivi, nella Fig. 3.3 viene mostrata l'architettura di riferimento dei Tag utilizzati nel progetto. L'etichetta elettronica è composta da un'antenna UHF abbinata ad un raddrizzatore a doppio percorso, seguito da un power management unit (PMU), wake-up receiver (WRU) e il generatore di impulsi UWB che controlla la rispettiva antenna UWB [8].

Un raddrizzatore UHF a doppio percorso è stato realizzato per riuscire a fornire energia al convertitore DC/DC, al generatore di impulsi e al WUR. Per alimentare i due sotto-sistemi utilizzando fonti di energia remota, la potenza ricevuta in RF viene subito divisa con un apposito circuito reattivo, evitando le perdite che ci sarebbero state se tale operazione fosse avvenuta in DC. Per sfruttare al meglio la potenza a disposizione si è optato per due separate antenne: una per l'energy-harvesting e una per la localizzazione UWB. Esistono anche soluzioni con un'unica antenna a doppia banda e un diplexer passivo per separare i due canali [15].

L'energia richiesta dal PMU varia a seconda che si operi in una condizione di *cold-start* o dopo precedenti cariche. Per la realizzazione del sotto-sistema RF a energy-harvesting si è fatto riferimento al caso peggiore di *cold-start*. Per raggiungere i 10m di distanza Tag-Reader è stata utilizzata la banda di

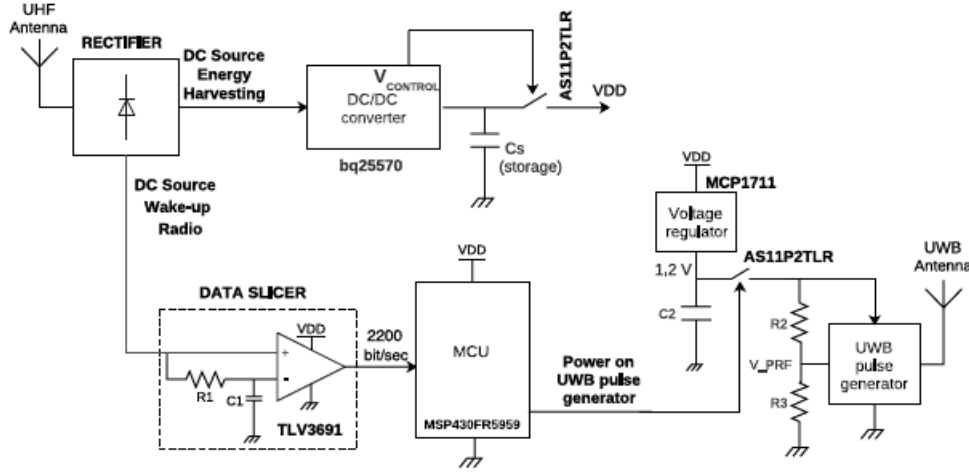


Figura 3.4: Circuito del Tag [8]

frequenze RFID GEN2 di 868MHz.

Tutti i componenti in Fig. 3.4 sono stati scelti per minimizzare sia il consumo di energia a riposo che quello complessivo.

Il convertitore DC/DC monitora la potenza fornita al microcontrollore in base all'energia raccolta, quest'ultima deve essere sufficiente per alimentare almeno una trasmissione UWB. Il segnale raddrizzato viene inviato al WUR composto da un comparatore, il quale esegue una modulazione On-Off Keying (OOK) sulla sorgente RF e decodifica l'indirizzo del Tag con l'aiuto di un microcontrollore. Un regolatore di tensione fornisce la potenza al generatore UWB attraverso uno switch controllato dal microcontrollore che può quindi, in base al livello di energia immagazzinato nella fase di energy-harvesting, connettere o disconnettere il generatore all'alimentazione.

Il generatore di impulsi UWB ha un consumo di energia pari a 1,5pJ per impulso. La durata dell'impulso è di circa 1,2ns corrispondente ad una larghezza di banda di 1,4GHz con una frequenza di ripetizione di 220MHz.

### 3.3 Sistema di localizzazione UWB

Nel seguito viene spiegata la tecnica di localizzazione basata su impulsi UWB facendo riferimento al caso di un solo trasmettitore e ricevitore [6, III]. Nel momento in cui il segnale di sincronizzazione è stato inviato dal Reader al sotto-sistema UWB a seguito di un'avvenuta identificazione di un Tag, il trasmettitore genera una serie di impulsi UWB separati da un intervallo di frame  $T_f=50\text{ns}$  che vengono riflessi dal Tag. La riflessione avviene tramite un interruttore che modula gli impulsi ricevuti in modo da generare una sequenza di impulsi riflessi positivi e negativi alternativamente.

Il ricevitore riceve subito gli impulsi generati dal trasmettitore e successivamente con un certo ritardo, dell'ordine del nanosecondo, riceve anche quelli riflessi dal Tag modulati diversamente.

Il primo passaggio consiste nel separare i vari impulsi ricevuti in base al  $T_f$  di 50ns per poi sommarli. Il vantaggio di sommare gli impulsi sta nell'aumentare l'ampiezza dell'impulso risultante di un fattore  $N$  corrispondente al numero degli impulsi ricevuti, di conseguenza la potenza utile aumenterà di un fattore  $N^2$ . Per quanto riguarda il rumore invece è indipendente tra i vari impulsi, si hanno tante variabili aleatorie gaussiane indipendenti. La varianza della somma di variabili gaussiane indipendenti è la somma delle singole varianze per cui la potenza del rumore cresce di un fattore  $N$ . Il risultato è un rapporto segnale-rumore SNR che aumenta di  $N$ . Considerando che il numero di impulsi trasmessi è 4000 allora SNR aumenta di un fattore 4000. Tutto ciò è necessario poiché per questioni normative l'impulso UWB è debole, non può avere potenza elevata.

L'operazione di somma viene fatta anche sugli impulsi ricevuti dalla riflessione del Tag, il problema però è che questi impulsi sono alternativamente positivi e negativi per cui la somma darebbe zero. Per questo motivo prima di eseguire la somma si ritaglia la finestra contenente questi impulsi e la si moltiplica per la stessa sequenza seguita dallo switch del Tag in modo tale che gli impulsi negativi diventino positivi. Una volta fatta la somma anche qui si ottiene lo stesso risultato: un SNR che aumenta di un fattore  $N$ .

Il motivo per cui gli impulsi riflessi dal Tag vengono modulati in maniera alternata positivi e negativi è quello di eliminare il Clutter. Quando viene trasmesso un impulso questo viene riflesso oltre che dal Tag anche



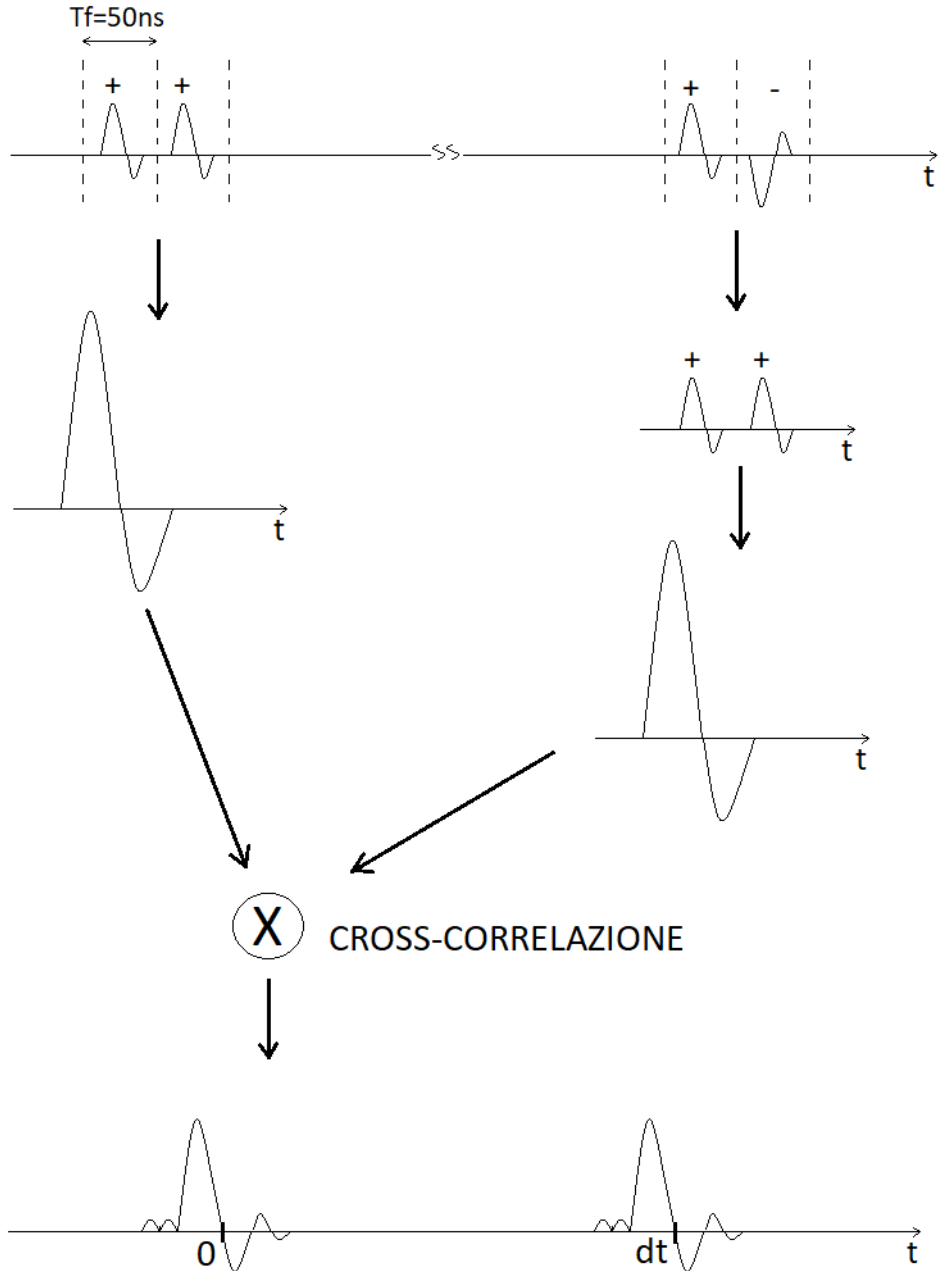


Figura 3.5: Sistema di localizzazione con impulsi UWB. L'impulso di sinistra si riferisce al segnale non modulato dalla sequenza ricevuto direttamente dal trasmettitore, mentre quello a destra si riferisce alla componente riflessa dal Tag e modulata dalla sequenza di codice.

dall'ambiente per cui il ricevitore registra oltre che gli impulsi riflessi dal Tag anche altri impulsi molto ampi dovuti alla riflessione dell'ambiente che sovrastano il segnale del Tag molto più debole. Il vantaggio è che questi impulsi chiamati Clutter sono tutti positivi mentre la parte utile nascosta nel Clutter è modulata con più e meno. Quando vado a moltiplicare la serie di impulsi con la stessa sequenza per poi sommarli, la parte utile come già detto viene amplificata di un fattore  $N$ , il Clutter, invece, si annulla perché da tutti impulsi positivi si genera una sequenza di impulsi positivi e negativi la cui somma da zero.

Il passaggio successivo consiste nella CROSS-CORRELAZIONE dei due impulsi ottenuti dalle precedenti operazioni. Se i due impulsi avessero lo stesso ritardo il risultato sarebbe un impulso centrato all'istante zero, se invece come in questo caso uno dei due è ritardato di un certo tempo  $dt$  rispetto all'altro allora avrò un ulteriore picco all'istante  $dt$ . Successivamente tramite un opportuno algoritmo si ricava l'istante  $dt$  in cui si ha il picco. Tale istante che è la differenza di tempo tra gli impulsi originali e quelli riflessi dal Tag è proporzionale alla differenza tra il percorso Tx-Rx e il percorso Tx-Tag-Rx. Siccome la distanza trasmettitore-ricevitore è nota posso ricavarmi la lunghezza del percorso Tx-Tag-Rx. Tale distanza definisce un'ellisse (luogo geometrico dei punti del piano per i quali è costante la somma delle distanze da due punti fissi detti fuochi) i cui fuochi sono Tx e Rx.

Con un solo trasmettitore e ricevitore si riesce quindi a sapere che il Tag sarà su un qualsiasi punto dell'ellisse avente come fuochi proprio il Tx e Rx, per cui se si vuole effettivamente localizzare il Tag è necessario avere almeno due ricevitori che generano due ellissi nelle quali una delle due intersezioni corrisponde alla posizione del Tag mentre l'altra solitamente corrisponde ad una posizione improbabile. L'ideale è utilizzare tre ricevitori essendo l'intersezione di tre ellissi un solo punto.

In realtà tutto il sistema spiegato è perfetto teoricamente, ma nel momento in cui lo si applica nella pratica non funziona. Il trasmettitore e ricevitore sono due oggetti diversi, per esempio nel nostro caso il primo è un generatore di funzioni il secondo è un oscilloscopio perciò hanno una visione del tempo diversa. Il generatore viene programmato per generare un impulso ogni 50ns e l'oscilloscopio per registrare il segnale ricevuto e tagliarlo ogni 50ns.

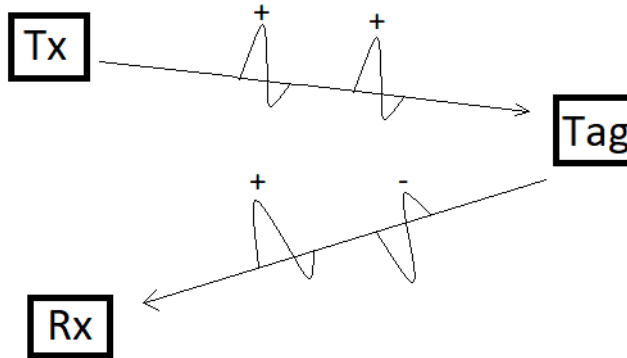


Figura 3.6: Architettura con un solo trasmettitore e ricevitore

Il problema è che i 50ns dell'oscilloscopio non corrispondono esattamente ai 50ns del generatore ci sono delle piccole differenze di qualche decina di picosecondo. All'inizio non ci sono problemi, ma nel tempo l'accumularsi di queste differenze porta l'oscilloscopio a tagliare gli impulsi nell'istante sbagliato. Per ovviare a queste complicazioni si utilizzano appositi algoritmi che cercano di capire quanto l'asse dei tempi è diluito o compresso andando per tentativi.

Tutta questa elaborazione viene fatta in Matlab ed è per questo che bisogna attendere circa un secondo prima di avere un risultato che visualizzi la posizione del Tag. Ovviamente in una possibile implementazione futura tutto ciò verrà eseguito su FPGA in tempo reale senza ritardi.

### 3.4 Reader RFID

Un software che legge/scrive i dati nei Tag richiede un Reader come interfaccia che possa permettere un accesso trasparente al supporto dati. Le operazioni di lettura e scrittura si basano sul principio MASTER-SLAVE: tutti i Reader sono inizializzati dall'applicazione per cui il software rappresenta il master mentre il lettore come slave è attivato quando i comandi di

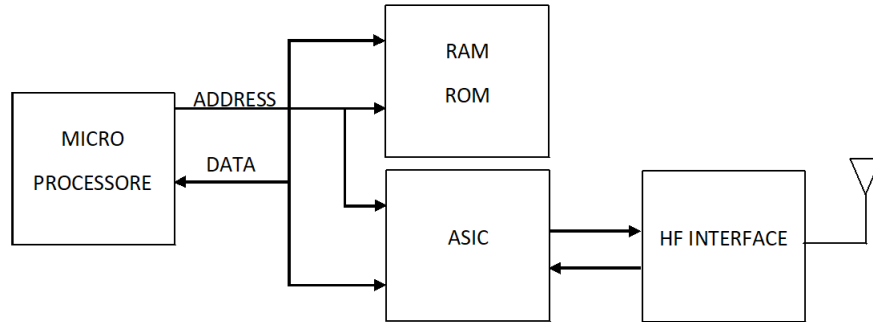


Figura 3.7: Architettura di un Reader

lettura/scrittura vengono ricevuti. Per eseguire il comando il Reader entra in comunicazione con il Tag per cui ora è il Reader a giocare il ruolo di master mentre il transponder risponde ai suoi comandi.

Nonostante diversi tipi di accoppiamenti (induttivo, elettromagnetico, capacitivo), di sequenze di comunicazione (FDX, HDX, SEQ), di procedure di trasmissione verso il Reader (load modulation, backscatter modulation, subharmonic) e di range di frequenza, tutti i Reader sono simili nei loro principi operativi base e quindi nel loro design. Possono essere ridotti in due fondamentali blocchi funzionali [1]: CONTROL SYSTEM e RF INTERFACE.

Interfaccia RF genera segnali ad alta frequenza per attivare il transponder e alimentarlo, modula il segnale trasmesso per inviare dati al Tag, demodula i segnali ricevuti da quest'ultimo. Alcuni sistemi impiegano un accoppiatore direzionale per separare i flussi da e verso il transponder.

CONTROL UNIT comunica con l'applicazione software ed esegue i comandi di controllo della comunicazione con il transponder, codifica e decodifica i segnali, esegue algoritmi di anticollisione. Solitamente si basa su un microprocessore con l'aggiunta di un modulo ASIC per procedure di crittografia e codifica di segnali.

# Capitolo 4

## EPCglobal Class1 Generation2

Per la comunicazione tra Tag, Reader e Client c'è la necessità di utilizzare protocolli standard, che permettano il funzionamento anche con componenti di produttori diversi:

- La comunicazione tra Tag e Reader è definita da quello che viene chiamato air protocol. Tra questi il più diffuso è l'EPC Class 1/Generation 2 [4];
- Per la comunicazione tra Reader e sistema informativo (Client) si adotta il protocollo standard Low Level Reader Protocol, LLRP [3].

Nel 1999 è stata avviata l'attività di ricerca e sviluppo per la definizione di un'architettura standard e aperta, con lo scopo di creare una rete globale di oggetti fisici definita un "Internet delle cose" (Internet of Things). Questo obiettivo è raggiungibile attraverso la tecnologia RFID che diventa il mezzo per collegare in modo trasparente gli oggetti alla rete globale: la EPC global Network. Quest'ultima è composta da 5 componenti:

1. l'Electronic Product Code (EPC) - un "codice a barre" elettronico che identifica il produttore, il prodotto, la versione ed il numero di serie di ogni singolo oggetto;
2. un sistema di identificazione o "system id" - composto da un'etichetta elettronica, o Tag, e da un lettore (Reader);

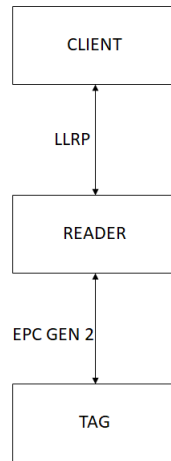


Figura 4.1: Protocolli di trasmissione in un sistema RFID

3. Epc Middleware - il software che consente lo scambio di informazioni tra il Reader ed il sistema informativo a cui si collega;
4. un "Discovery Service" - una rete di nodi di servizio hardware e software in grado di trovare i dati correlati con quel specifico codice EPC;
5. un "servizio informativo" EPC - che consente agli utilizzatori di scambiare dati basati sul codice EPC.

Le specifiche della Classe 1 / Generazione 2 - comunemente richiamate con il termine Gen 2 - sono state sviluppate per superare limitazioni delle specifiche precedenti. EPCglobal Class 1 Generation 2 UHF Air Interface Protocol stabilisce i requisiti fisici e logici per un sistema RFID operante in banda 860-960 MHz. Emesso da EPCglobal nel 2004, è successivamente stato recepito dall'ISO diventando uno standard in diversi settori. EPC Gen2 descrive le caratteristiche costruttive di Tag e Reader e definisce procedure, comandi, modulazioni e gestione delle collisioni con lo scopo di rendere il sistema RFID più efficiente e affidabile.

I Tag UHF EPC Gen2 sono passivi, pensati per essere economici, di piccole

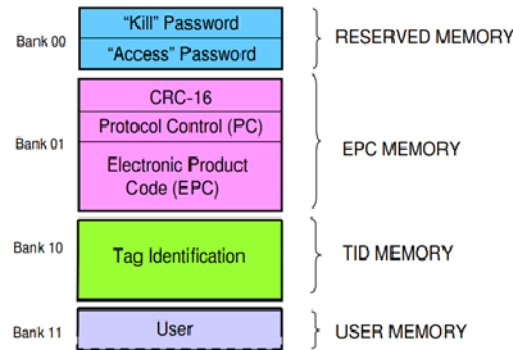


Figura 4.2: Mappa della memoria di un Tag RFID Gen2

dimensioni, e la loro memoria è ripartita in quattro parti:

- Memoria riservata: può contenere le password per le funzioni “kill”, che permette di rendere il Tag permanentemente inutilizzabile, e “access”, per accedere alla memoria del Tag;
- Electronic Product Code (EPC): informazione principale del Tag per identificare univocamente l’oggetto su cui è posto, formata da una stringa di almeno 96 bit riscrivibile;
- TID: memoria di tipo read-only, contiene le informazioni che identificano produttore e modello del Tag;
- Memoria destinata all’utente.

L’operazione fondamentale che compie il Reader è quella di inventory che consiste nel rilevare i Tag presenti nella Field Of View (FOV) delle antenne leggendone gli EPC. RFID Gen2 definisce 4 differenti sessioni per l’inventory. Per ogni sessione ogni Tag ha un flag di stato impostato su A (default state) o B (inventoried state). Normalmente durante l’inventory i Tag in stato A

vengono interrogati e passano allo stato B in cui rimangono per un certo intervallo di tempo smettendo di rispondere al Reader. Essendo gli stati vincolati alla sessione, diversi Reader che operano in differenti sessioni non interferiscono fra loro. Le sessioni sono così definite:

- S0: Richiede che il Tag torni allo stato A ogni volta che viene inventariato. Lavorare in questa sessione è ottimale quando vogliamo leggere più volte lo stesso Tag, al contrario è poco efficiente quando abbiamo una numerosa popolazione di Tag.
- S1: il Tag rimane nello stato B per massimo cinque secondi. Lo stesso Tag verrà perciò letto meno volte rispetto al caso precedente.
- S2 e S3: il Tag deve rimanere nello stato B per almeno 2 secondi. Queste sessioni sono ottimali in un ambiente con molti Tag.

## 4.1 Low Level Reader Protocol

L'architettura dell'EPCGlobal network richiede la necessità di controllare e coordinare la rete di Reader. Il protocollo LLRP ha esattamente questo scopo, permettendo al Client di dialogare con il Reader per coordinare le operazioni. Rappresenta un'interfaccia di basso livello comune a più Reader di produttori diversi. Le principali funzioni che LLRP fornisce sono:

- Possibilità di leggere il codice EPC ed altri dati contenuti nel Tag, scrivere dati nel Tag ed eseguire altri comandi, come quelli descritti da EPC Gen2;

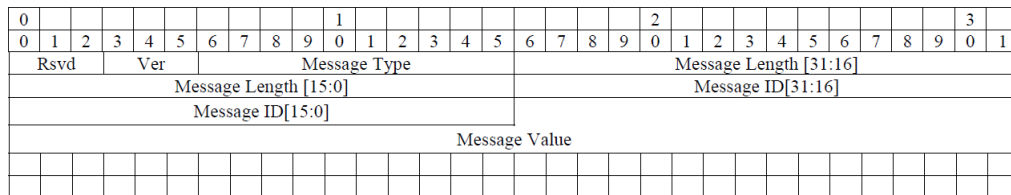


Figura 4.3: Struttura binaria di un messaggio LLRP



- Capacità di gestire gli errori che si possono generare durante l'accesso al Tag;
- Capacità di regolare la potenza RF e l'utilizzo dello spettro del Reader;
- Possibilità di aggiungere supporto per nuovi air protocols;
- Possibilità di conoscere le funzioni consentite dal Reader;
- Possibilità di definire estensioni al protocollo specifiche per i produttori, facendo in modo che non interferiscano con quelle di altri produttori.

Le due funzioni fondamentali di un Reader su una popolazione di Tag sono **INVENTORY** e **ACCESS**. Inventory è l'operazione di identificazione dei Tag, il Reader interroga un singolo Tag e richiede il suo EPC. Access consiste in operazioni di comunicazione (lettura e scrittura) col Tag, ma quest'ultimo deve essere stato univocamente identificato prima di accedervi. Volendo si può aggiungere una terza categoria chiamata **SELECT** che racchiude tutte quelle operazioni che il Reader esegue su un sottoinsieme di Tag, ciò risulta utile nel caso in cui si volessero fare degli inventari o degli accessi solo su un particolare gruppo di Tag e non su tutta la popolazione. LLRP definisce una serie di messaggi, parametri e campi per la comunicazione tra i due interlocutori. L'elemento fondamentale è il messaggio, LLRP ha una libreria di quaranta messaggi per la comunicazione tra Client e Reader costituiti da comandi, risposte, segnalazioni di eventi, abilitazione di un evento, messaggi custom dedicati a possibili estensioni da parte dei produttori del Reader. I dati inviati sono contenuti nel messaggio come parametri e campi, si veda la fig. 4.3 che mostra la struttura della codifica binaria di un messaggio LLRP. I messaggi dal Client al Reader includono il recupero e l'impostazione della configurazione del lettore, rilevamento della capacità del Reader, la gestione dell'inventario e le operazioni di accesso. I messaggi dal Reader al Client sono costituiti dalla segnalazione del proprio stato e da dei report per mostrare i risultati dell'inventario e degli accessi. Le operazioni LLRP possono essere classificate nelle due categorie seguenti che rispecchiano le funzionalità del Reader esposte precedentemente:

- Reader Operation (RO): definiscono le operazioni che le antenne del Reader devono eseguire, ovvero:

- Antenna Inventory (AI): L'interazione più elementare tra Reader e Tag nel FOV dell'antenna. Qui possono essere definiti i parametri riguardanti l'operazione di inventory come le antenne da usare, il trigger di stop per l'inventory, impostazioni per l'air protocol, ecc. Possono essere definite più AI con parametri diversi che verranno eseguite in successione.
- RF Survey: Si possono stabilire criteri come la frequenza iniziale e finale a cui viene fatta l'inventory.

Si deve definire una Reader Operation Parameter Specification (ROSpec) che definisce i parametri per la RO e comprende almeno una Spec (parameter specification for an operation), che può essere di tipo AISpec o RFSurveySpec. Nella ROSpec sono contenute anche le istruzioni per generare i report, ovvero in che modo o a seguito di quale evento il Reader deve comunicare al Client le informazioni sui Tag raccolte, a seguito di operazioni di inventory o survey.

- Access Operations: Sono le operazioni di accesso ad un singolo Tag, come scrivere o leggere particolari informazioni diverse dall'EPC. Analogamente al caso precedente sono descritte da AccessSpec che deve indicare il Tag su cui si vuole svolgere l'operazione (TagSpec), il tipo di operazione (OpSpec) e può contenere altre informazioni come quelle riguardanti i report.

#### 4.1.1 Messaggi LLRP

Nel seguito verrà riportato un elenco dei principali messaggi scambiati tra Reader e Tag per aprire una sessione d'inventario poiché per l'attività di tesi si è fatto riferimento solo a questo tipo di operazione essendo l'unica che permette di conoscere gli EPC della popolazione di Tag.

Si useranno le seguenti abbreviazioni:

*CtR*: Client to Reader, ovvero pacchetti creati dal software e spediti al Reader.

*RtC*: Reader to Client, ovvero pacchetti ricevuti dal Reader inoltrati al software.

Prima di qualunque messaggio viene creata la connessione tramite indirizzo IP del Reader e porta di default 5084. Il Client resta in attesa di un messaggio di avvenuta connessione.

1. RtC: `READER_EVENT_NOTIFICATION` è il messaggio che notifica l'avvenuta connessione, restituisce l'evento `Success` se l'operazione avviene con successo altrimenti `Failed`.
2. CtR: `GET_READER_CAPABILITIES` è una richiesta inoltrata al Reader per conoscere le sue caratteristiche.
3. RtC: `GET_READER_CAPABILITIES_RESPONSE` indica quali sono le funzioni che può eseguire il Reader, che saranno diverse a seconda di marca e modello. Qui si può controllare se il Reader è di marca Impinj e in caso affermativo successivamente si possono abilitare le estensioni custom.
4. CtR: `SET_READER_CONFIG` permette di settare le impostazioni di base del Reader.
5. RtC: `SET_READER_CONFIG_RESPONSE` indica l'esito positivo alla richiesta.
6. CtR: `CUSTOM_MESSAGE:IMPINJ_ENABLE_EXTENSIONS` permette di abilitare le funzionalità custom del Reader Impinj.
7. RtC: `CUSTOM_MESSAGE:IMPINJ_ENABLE_EXTENSIONS_RESPONSE` ritorna lo stato di successo/insuccesso.
8. CtR: `ADD_ROSPEC` aggiunge una ROSpec al Reader contenente diversi parametri tra cui:
  - `ROSpecState:Disabled`, il ROSpec deve essere inizialmente disabilitato secondo lo standard;
  - `ROSpecID` identificativo del ROSpec;

- ROSpecStartTriggerType:Null, il ROSpec viene avviato attraverso un opportuno messaggio di START\_ROSPEC da parte del Client;
  - ROSpecStopTriggerType:Null, si ferma al termine di tutte le altre Spec o col comando STOP\_ROSPEC da parte del Client;
  - AISpec che definisce le operazioni di inventario sulle antenne, a sua volta contiene una serie di parametri per definire questo inventario: AISpecStopTrigger, AntennaIDs, ecc...
9. RtC: ADD\_ROSPEC\_RESPONSE segnala l'esito positivo (o negativo) della richiesta precedente.
  10. CtR: ENABLE\_ROSPEC è la richiesta di abilitare il ROSpec creato.
  11. RtC: ENABLE\_ROSPEC\_RESPONSE ritorna lo stato di successo/insuccesso.
  12. CtR: START\_ROSPEC contiene come parametro il ROSpecID del ROSpec da avviare.
  13. RtC: START\_ROSPEC\_RESPONSE risposta al messaggio precedente, se c'è un ROSpec abilitato corrispondente al ROSpecID in questione allora restituisce l'evento Success.
  14. RtC: RO\_ACCESS\_REPORT fornisce le informazioni sui Tag rilevati. In questo caso avremo un messaggio di questo tipo per ogni Tag letto e conterrà:
    - EPC
    - AntennaID
    - FirstSeenTimestamp, istante di tempo in millisecondi in cui è stato rilevato il Tag per la prima volta secondo lo standard UTC.
    - PeakRSSI, potenza del segnale radio ricevuto.

15. CtR: `DISABLE_ROSPEC` interrompe l'invio di `RO_ACCESS_REPORT`. Questo messaggio viene generato nel momento in cui l'utente interrompe l'inventario tramite interfaccia grafica. Il ROSpec passa allo stato disabilitato.
16. RtC: `DISABLE_ROSPEC_RESPONSE` ritorna lo stato di successo/insuccesso.
17. CtR: `STOP_ROSPEC` sposta il ROSpec in uno stato inattivo.
18. RtC: `STOP_ROSPEC_RESPONSE` risposta al messaggio precedente.
19. CtR: `DELETE_ROSPEC` elimina il ROSpec
20. RtC: `DELETE_ROSPEC_RESPONSE` ritorna lo stato di successo/insuccesso.
21. CtR: `CLOSE_CONNECTION` è la richiesta di poter chiudere la connessione LLRP. Il Client resta in attesa della risposta prima di effettuare la chiusura.
22. RtC: `CLOSE_CONNECTION_RESPONSE` ritorna lo stato di successo/insuccesso e chiude la connessione.

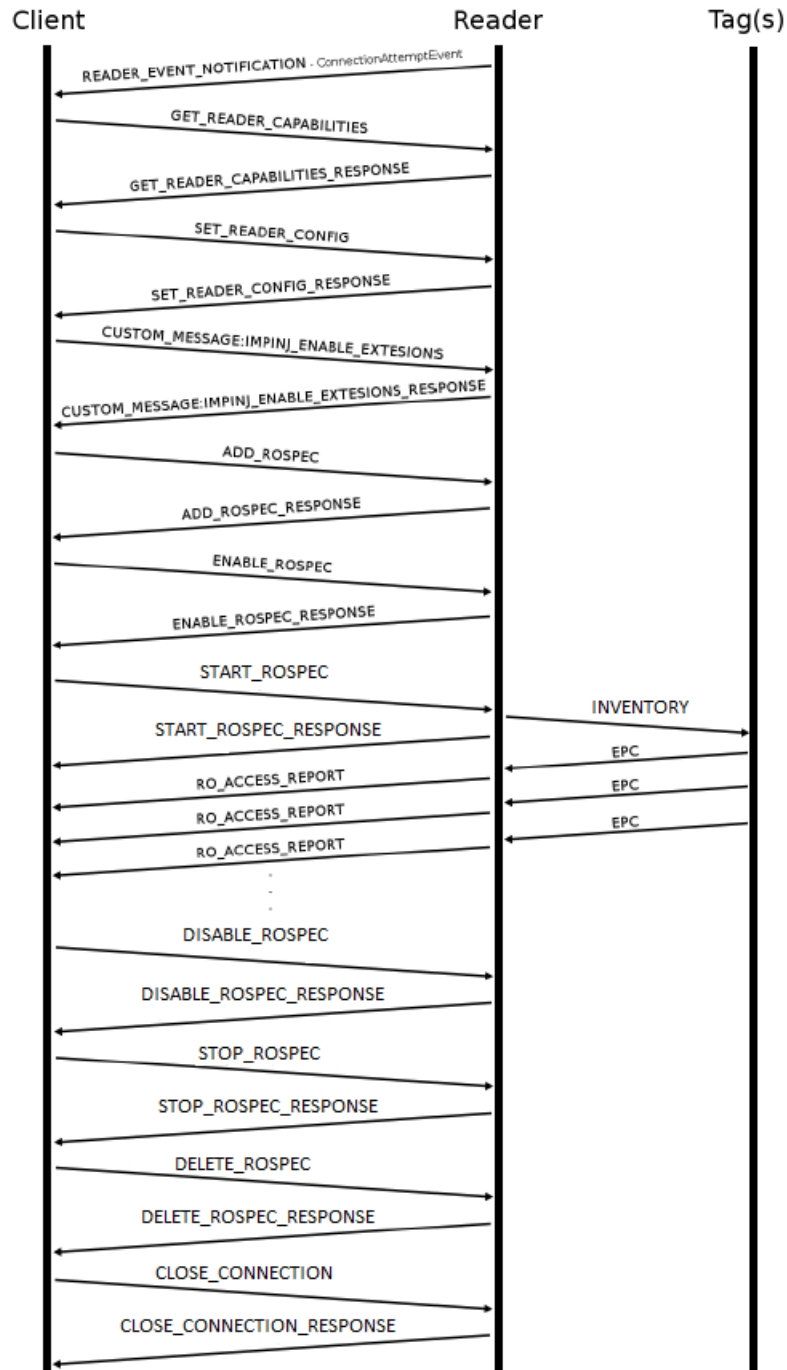


Figura 4.4: Messaggi LLRP scambiati tra Client e Reader durante una sessione d'inventario [13]

# Capitolo 5

## Sincronizzazione

Nell'architettura usata in laboratorio, Fig.3.2, uno dei problemi da risolvere è avvertire l'oscilloscopio quando aprire la finestra di campionamento del segnale ricevuto da Rx. Tale finestra va aperta quando il Tag è ancora attivo e in grado di riflettere gli impulsi UWB ricevuti da Tx. A gestire questa sincronizzazione se ne occupa il Reader poiché essendo lui ad alimentare il Tag durante l'inventario è lui a sapere quando è attivo o meno.

In Fig. 5.1 sono mostrati i segnali di cui si è tenuto conto nel generare il trigger: alimentazione dell'etichetta, EPC, impulsi UWB trasmessi dal Tag. Quest'ultimo segnale è un clock con frequenza di 1MHz. Il trigger deve essere inviato dopo l'EPC durante la finestra di alimentazione e prima che finisca la sequenza di backscatter.

Durante l'inventario ci sono una serie di messaggi scambiati tra Reader e Tag mostrati in Fig. 5.2. Il primo comando eseguito è il *Select* che seleziona la popolazione di Tag su cui fare l'inventario, segue un comando di *Query* che dà inizio a quello che viene chiamato un'inventario round. La Query contiene un parametro Q il cui valore viene settato in funzione della dimensione della popolazione di Tag. Ogni Tag che partecipa all'inventario ricevendo il comando di Query attiva un contatore che va a decrementare un numero casualmente selezionato nell'intervallo  $[0, 2^Q - 1]$  e caricato in memoria. Quando questo contatore arriva a zero il Tag trasmette al Reader un messaggio contenente l'RN16. RN16 è un numero pseudo-casuale di 16 bit generato dal Tag e salvato in memoria. Il Reader ingloba RN16 ricevuto

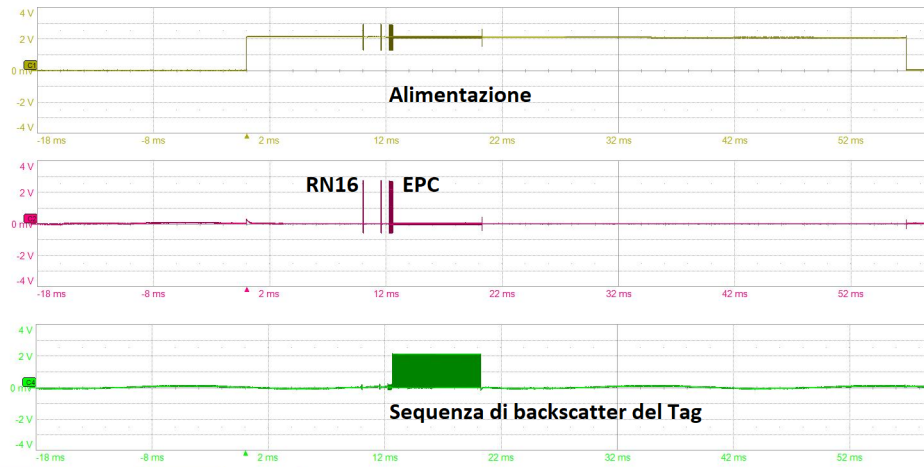


Figura 5.1: Segnali UHF e UWB

all'interno di un messaggio di *Ack* da trasmettere. Solo il Tag con lo stesso numero contenuto nell'*Ack* risponde inviando l'EPC [4].

La prima parte della comunicazione esegue un accesso random ai Tag con lo scopo di minimizzare le probabilità di collisione grazie alla generazione di numeri casuali in un intervallo la cui dimensione dipende dalla popolazione di Tag. Per cui più Tag ci sono maggiore sarà la finestra in cui quest'ultimi dovranno selezionare un numero, cioè minore sarà la probabilità che sceglieranno lo stesso numero. La seconda parte della comunicazione inizia quando il Tag con il contatore a zero risponde al Reader, da quel punto in poi il Tag è stato selezionato per cui si esegue un accesso deterministico attraverso l'RN16. Ovviamente durante tutta la comunicazione negli intervalli in cui non c'è scambio di messaggi viene trasmessa dal Reader la continua CW per mantenere attivo il Tag.

Ricapitolando, nel momento in cui tramite protocollo LLRP il Client inizia una Reader Operation (RO), usando opportuni messaggi e parametri (ROSpec, AISpec, RFSurveySpec), si avvia una sessione d'inventario in cui avviene lo scambio di messaggi tra Reader e Tag appena spiegato, tale sessione termina con l'invio dell'EPC al Reader e successivamente al Client come report d'inventario.



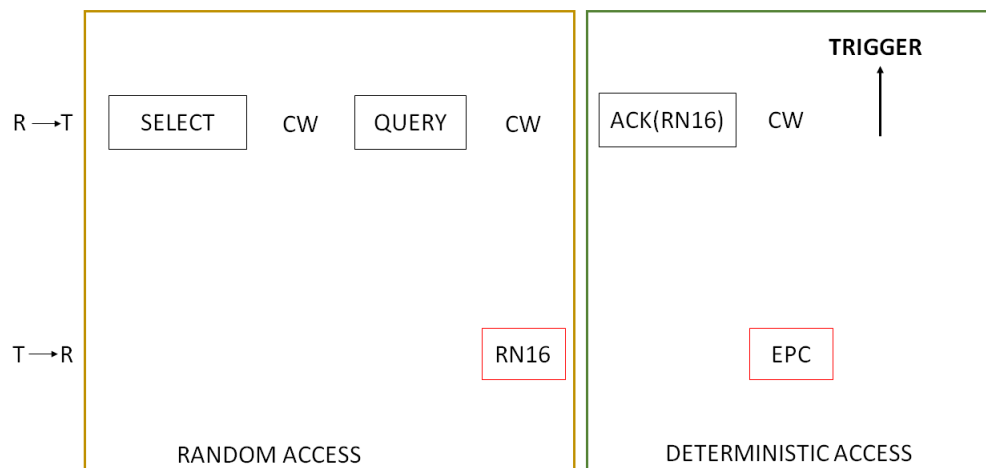


Figura 5.2: Comunicazione d'inventario tra Reader e Tag

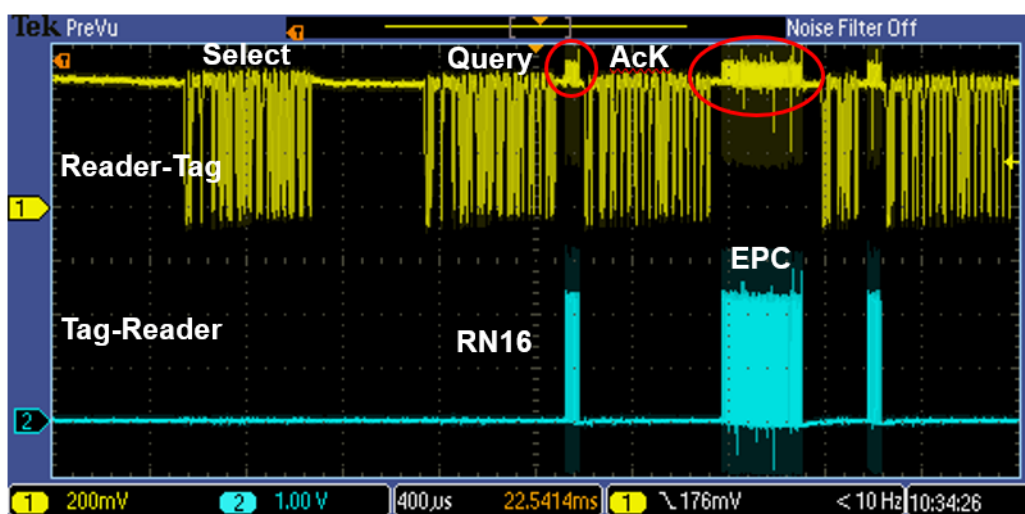


Figura 5.3: Inventario visto all'oscilloscopio

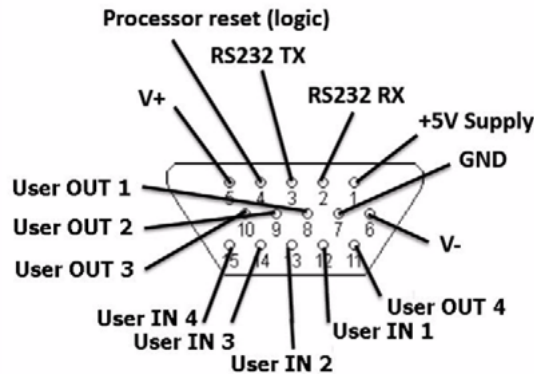


Figura 5.4: Porta GPIO Reader Impinj

## 5.1 Generazione del trigger tramite LLRP

Una prima possibile soluzione per sincronizzare il sistema consiste nell'usare la porta GPIO e il protocollo LLRP. Tramite opportuni messaggi LLRP si può generare un trigger in uno dei quattro pin di uscita GPO da trasmettere all'oscilloscopio che a sua volta presenta una connessione GPIO. In questa tesi si è usato un ambiente di sviluppo Java con l'uso di librerie LLRP che permettano di definire i vari messaggi. In particolar modo a noi interessa il messaggio di SET\_READER\_CONFIG inviato dal Client al Reader per configurare quest'ultimo attraverso i parametri specificati nel comando. Tale messaggio rimane valido per tutta la durata della connessione LLRP o fino ad un nuovo messaggio di configurazione. Tra i parametri di SET\_READER\_CONFIG compare GPOWriteData che permette di scrivere su una porta di uscita GPO specificando il pin di uscita utilizzato (GPO Port Number) e il dato da scrivere (GPO Data) che nel nostro caso sarà il bit 1 [3]. Oltre a tale messaggio in realtà è necessario usare un ulteriore parametro custom del Reader Impinj: ImpinjAdvancedGPOConfiguration [5]. Tale parametro aggiunge dei controlli avanzati sulla GPO del Reader Speedway, in nostra dotazione, attraverso i seguenti campi:

- GPOPortNum, intero senza segno che determina la porta di uscita utilizzata;

- GPOMode, intero senza segno che determina la modalità con la quale si configura la porta GPO:
  - 0, modalità Normal (di default) dove la GPO è configurata tramite il messaggio LLRP SET\_READER\_CONFIG.
  - 1, modalità Pulsed dove la GPO cambia stato in base ai parametri settati in SET\_READER\_CONFIGURATION, ma tornerà nel suo stato dopo un tempo in millisecondi pari al valore contenuto nel campo GPOPulsedDurationMSec.
  - 2, Reader\_Operational\_status.
  - 3, LLRP\_Connection\_Status.
  - 4, Reader\_Inventory\_status.
  - 5, Network\_Connection\_Status.
  - 6, Reader\_Inventory\_Tags\_status.

Nelle ultime cinque modalità lo stato della GPO è di tipo booleano, per cui per esempio se c'è una connessione LLRP attiva con GPOMode=3 allora in uscita alla GPO avremo uno (true) altrimenti zero (false) se la connessione è chiusa, analogo discorso per gli altri stati.

Utilizzando questi parametri quello che è stato fatto è di configurare tramite `ImpinjAdvancedGPOConfiguration` il pin GPO 1 di uscita in modalità Pulsed con durata dell'impulso di un millisecondo, poi si è creata una funzione apposita che tramite il parametro `GPOWriteData` scrivesse il bit 1 nel pin. Nella Fig.5.5 è stato riportato il codice della funzione che genera il trigger. Tale funzione viene invocata quando il Client riceve il report dal Reader ed è inserita ovviamente in un contesto più ampio riguardante un progetto Java per la realizzazione di un'interfaccia grafica, Fig. 5.6, realizzata dallo studente Iban Guibert, che permetta di eseguire sul Reader operazioni di accesso, inventario e configurazione dei parametri tra cui anche ben appunto il settaggio della porta GPO per l'invio del trigger.

Tramite interfaccia grafica è possibile determinare la durata dell'inventario e durante tale intervallo tra Reader e Tag si ripete la sequenza di messaggi in Fig. 5.2 con un'unica variazione: la Query viene inviata solamente la prima volta in apertura di comunicazione determinando un Q univoco. Durante

```

public int sendGPOTrigger()
{
    GPOWriteData gPOWriteData = new GPOWriteData();
    gPOWriteData.setGPOData(new Bit(1));
    gPOWriteData.setGPOPortNumber(new UnsignedShort(1));

    SET_READER_CONFIG gpoConfig = new SET_READER_CONFIG();
    gpoConfig.addToGPOWriteDataList(gPOWriteData);

    gpoConfig.setResetToFactoryDefault(new Bit(0));
    gpoConfig.setMessageID(new UnsignedInteger(m_MessageID));
    m_MessageID++;

    sendMessage(gpoConfig);

    return 0;
}

```

Figura 5.5: Funzione `sendGPOTrigger` che genera e trasmette il trigger

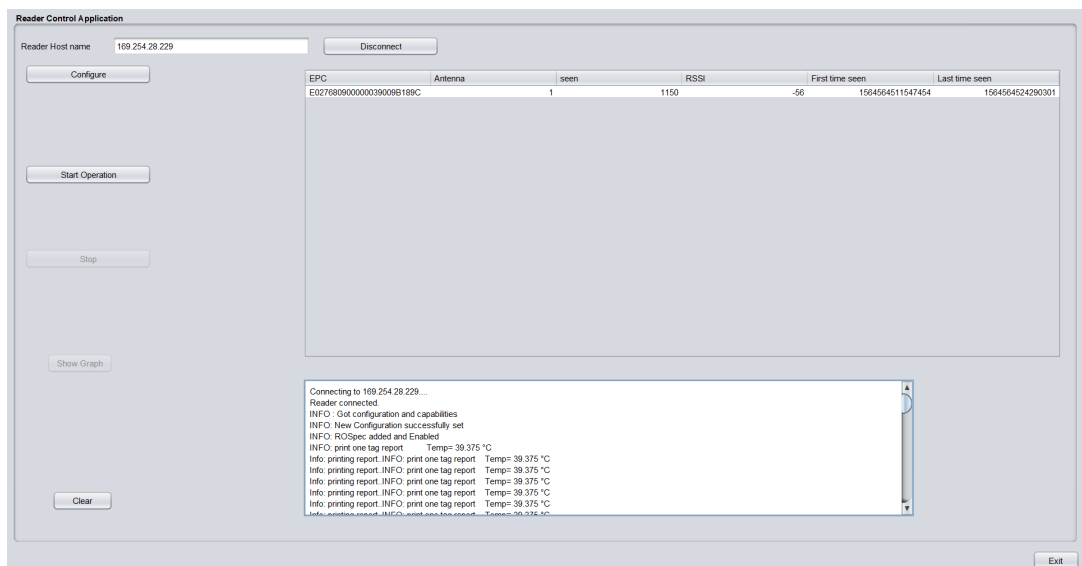


Figura 5.6: Interfaccia grafica per il controllo del Reader

la comunicazione, invece, vengono inviate dal Reader delle QueryAdjust e QueryRep. QueryAdjust in base al numero di Tag che rispondono all'inventario se necessario incrementa o decrementa il  $Q$ , si tenga conto che inizialmente non è detto si conosca esattamente la popolazione di Tag per cui per determinare un valore iniziale di  $Q$  si sceglie un numero più possibile vicino all'esatta popolazione, poi ci penserà il QueryAdjust ad aggiustare  $Q$ . QueryRep, invece, se necessario cambia la sessione del Reader.

Durante tutta la durata dell'inventario ogni volta che il contatore del Tag arriva a zero si trasmette l'RN16 e poi l'EPC, per cui se per esempio un Tag ha un contatore impostato a  $N$  una volta raggiunto lo zero risponde al Reader e riavvia il contatore ripartendo da  $N$ . Questo vuol dire che durante una sessione d'inventario il Reader riceve dallo stesso Tag più volte il suo EPC, ma il trigger non deve essere generato ogni volta basta trasmetterlo la prima volta per avvertire l'oscilloscopio che il Tag è in funzione.

### 5.1.1 Risultati e Problemi

Questa prima soluzione in realtà è risultata dai test effettuati in laboratorio fallimentare. Per capirne i motivi è necessario scendere nei particolari dei test effettuati.

Innanzitutto in laboratorio i vari esperimenti sono stati effettuati ad una distanza di due metri tra Tag e Reader, con una popolazione di Tag pari a uno. Questo implica che la  $Q$  venga settata dal Reader a zero cioè l'intervallo di numeri disponibili converge ad un unico valore: zero. Il Tag quindi non deve avviare nessun contatore, ma è subito pronto a rispondere e per tutta la durata dell'inventario il Reader riceve unicamente i suoi EPC.

Il Tag utilizzato è differente da quelli commerciali per due motivi:

- Ha in aggiunta tutta l'architettura necessaria al backscattering degli impulsi UWB per la localizzazione del Tag vedi Fig. 3.4,
- possiede un sensore di temperatura il cui dato viene trasmesso all'interno dell'EPC. EPC è una stringa di 96 bit, per identificare i Tag non è necessario utilizzarli tutti per cui si può inserire il valore di temperatura direttamente al suo interno con il vantaggio di trasmettere un solo dato piuttosto che due.

Tutto ciò si rispecchia in un Tag a componenti discrete con microcontrollore mentre quelli commerciali presentano un unico chip collegato ad un'antenna il tutto estremamente ottimizzato dal punto di vista energetico. La conseguenza è una maggiore lentezza nell'accendersi avendo bisogno di più energia. Se un Tag commerciale si attiva nei primi 4/5ms in cui il Reader invia la CW prima della Query, il nostro Tag, a due/tre metri di distanza dal Reader, richiede 300ms per caricarsi avendo una capacità più grande che permette di fornire energia a tutte le componenti discrete. Quindi mentre il Tag commerciale risponde ad ogni slot d'inventario il nostro Tag per i primi 300ms non risponde, anche se il Reader sta trasmettendo l'inventario non riceve risposta, una volta carico il Tag risponde trasmettendo l'EPC poi però durante la risposta si scarica per cui necessita di ulteriori 300ms per ricaricarsi, in realtà richiede un po' meno tempo perché la capacità sta volta non è completamente scarica. Il carico viene disabilitato quando l'alimentazione sulla capacità di storage scende sotto 1.8V ed è necessaria una tensione di 2.3V per riuscire ad alimentare il Tag. In conclusione in un secondo il Tag risponde mediamente tre volte.

Ritornando ai test in laboratorio, quando si attiva il sistema il Reader dà inizio alla sequenza d'inventario inviando una CW per 4-5ms seguita da una query a cui si aspetta subito una risposta sapendo che il Q è stato settato a zero, in realtà nelle prime sessioni non riceve niente dovendo il Tag ricaricarsi. Dopo 300ms si ha la prima risposta che si conclude con l'invio non di un solo EPC, ma per come è stato programmato il firmware del Tag con l'invio di quindici EPC ai quali seguono i rispettivi trigger. Il primo problema sta nel fatto che la funzione `sendGPOTrigger` in Fig. 5.5, all'interno del programma Java viene invocata ogni volta che il Reader riceve un EPC ci si aspetterebbe che per ogni EPC subito dopo venga trasmesso il corrispondente trigger in realtà in Fig. 5.7 si osserva come il treno di trigger parta solo dopo la trasmissione di tutti gli EPC. Il motivo è dovuto principalmente ad una maggiore priorità assegnata all'inventario. Analizzando i file di log durante l'esecuzione di un inventario si è visto che finché è in corso l'inventario il Reader non esegue altro per cui ogni richiesta di attivazione della GPO per la trasmissione del trigger viene rimandata a fine sequenza d'inventario una volta arrivati tutti gli EPC. Questa complicazione verrà successivamente risolta andando a modificare il

firmware del Tag per poter inviare un solo EPC.

Un secondo problema è il notevole ritardo tra la ricezione dell'EPC e l'invio del trigger dovuto a diversi fattori, tra cui la lentezza intrinseca nel protocollo TCP/IP usato nel collegamento ethernet tra Client e Reader. Inoltre l'uso del protocollo LLRP richiede per ogni comando una risposta da parte del Reader, SET\_READER\_CONFIG deve essere seguito da un messaggio di SET\_READER\_CONFIG\_RESPONSE per confermare l'avvenuta ricezione del messaggio e questo causa ritardi nell'attivazione per esempio della porta GPO. In realtà il ritardo non è detto sia un problema poiché il trigger serve ad attivare la finestra di ricezione dell'oscilloscopio e questo deve avvenire mentre il Tag è attivo e non necessariamente subito dopo la trasmissione degli EPC. Diventa un problema quindi se il trigger viene generato quando ormai il Tag è spento e questo è possibile nel Tag di laboratorio perché come già detto dopo ogni sequenza d'inventario rimane spento per i successivi 300ms. Quello commerciale, invece, essendo sempre acceso non ha queste complicazioni, il ritardo non è rilevante. In verità un notevole ritardo nella generazione del trigger potrebbe in ogni caso provocare complicazioni soprattutto se la popolazione di Tag è elevata. In questo caso durante un inventory round il Reader riceve gli EPC da più Tag e se per esempio il trigger n-esimo viene trasmesso con elevato ritardo dalla ricezione dell'n-esimo EPC potrebbe accadere che in tale intervallo di tempo il Reader ricevi l'EPC di un altro Tag, a questo punto l'oscilloscopio non capirebbe a quale indirizzo fa riferimento il trigger ricevuto. Le latenze misurate in laboratorio raggiungono i 20/30ms come si può osservare nella immagine di destra in Fig. 5.8 dove si deve tenere in conto che il trigger raffigurato fa riferimento al primo EPC.

Terzo problema e sicuramente il più critico è la presenza di Jitter sulla durata del ritardo, la variazione dell'intervallo di tempo tra EPC e trigger è elevata, si vedano le due immagini in Fig. 5.8. Se il ritardo in sé come detto prima potrebbe non causare difficoltà, il Jitter sulla sua durata, invece, è estremamente dannoso per il funzionamento del sistema perché genera incertezza. A volte quando il ritardo sarà piccolo l'oscilloscopio riuscirà ad aprire la finestra con il Tag ancora attivo, al contrario altre volte sarà già spento a causa di una latenza maggiore. La criticità del Jitter sta proprio nell'incertezza che genera, se ci fosse un ritardo costante una volta noto si

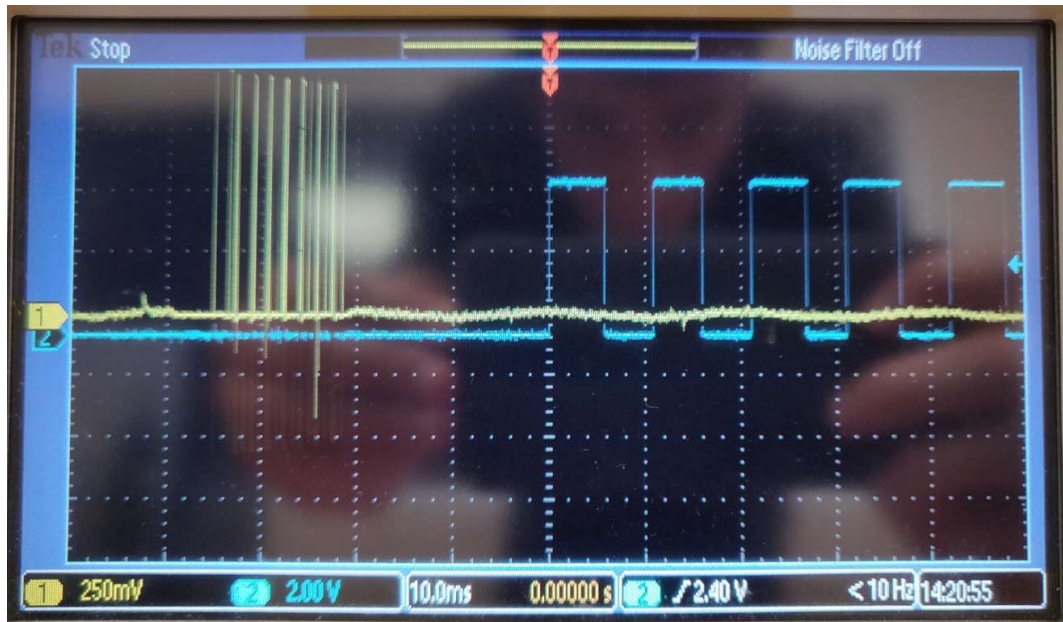


Figura 5.7: In giallo gli EPC, in blu i trigger

andrebbe a programmare l'oscilloscopio tenendo conto di tale tempo, per esempio l'oscilloscopio in dotazione in laboratorio permette di aprire la finestra qualche microsecondo prima dell'arrivo del Tag grazie ad un buffer che memorizza comunque i campioni ricevuti. Se invece è presente del Jitter non è più possibile prendere precauzioni perché la durata del ritardo cambia continuamente.

## 5.2 Generazione del trigger tramite seriale

Come scritto nel paragrafo precedente generare il trigger usando opportuni messaggi LLRP con cui comandare il Reader non è una soluzione efficace. Per questo motivo il secondo approccio utilizzato consiste nel generare il segnale tramite porta seriale per verificare che magari questa generi un jitter ridotto rispetto alla GPIO. Inoltre per eliminare il più possibile latenze nella comunicazione Client-Reader, stavolta il programma per la gestione



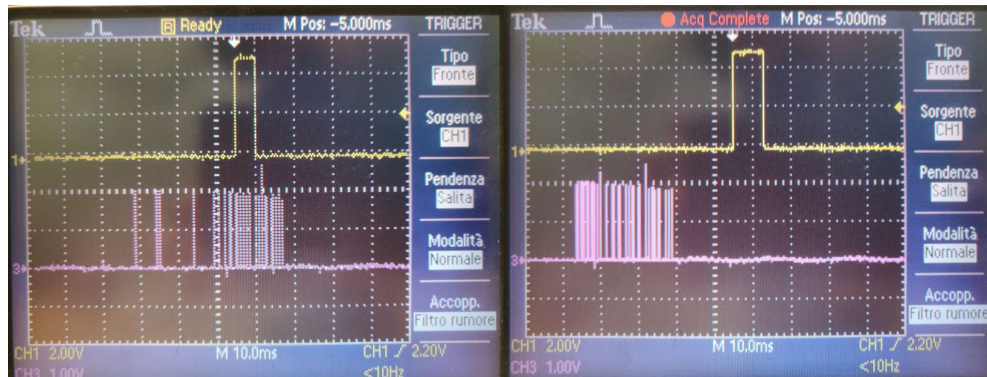


Figura 5.8: In rosa gli EPC, in giallo il trigger

dell'inventory e la trasmissione dei trigger viene caricato direttamente nella memoria del Reader. Il Client è sempre collegato tramite cavo ethernet al Reader solo che in questa configurazione deve solamente aprire la connessione e avviare l'inventory dopo di che non c'è più uno scambio di messaggi LLRP tra i due dispositivi, ma il Reader avendo in memoria il programma sa cosa fare. Il Client riceve solamente i RO\_ACCESS\_REPORT contenente gli EPC stampati successivamente su linea di comando così facendo si limita l'uso dei protocolli TCP/IP e LLRP tra Client e Reader che come già detto implicano notevoli ritardi.

A differenza del primo approccio qui non si ha un'interfaccia grafica per cui non si può scegliere a priori la durata dell'inventory ma l'utente lato Client può solo decidere quando iniziare l'inventory e quando fermarlo.

Nella gestione della seriale per la generazione del trigger non è stato usato il protocollo LLRP poiché dai primi test sulla GPIO si è visto che è implicitamente lento, ma sono state utilizzate librerie appositamente rilasciate per la configurazione della porta seriale del Reader Impinj.

Dal punto di vista della struttura del codice non ci sono differenze, la gestione del trigger avviene tramite un'apposita funzione che viene invocata nel programma principale (scritto in C++ non più in Java) ogni qualvolta il Reader riceve un EPC. In tale funzione si vanno a settare tutti i parametri per la comunicazione seriale, in particolare nel caso di interesse si ha una bit-rate di 115.220bps, 8 bit di dato, nessun bit di parità e un solo bit di

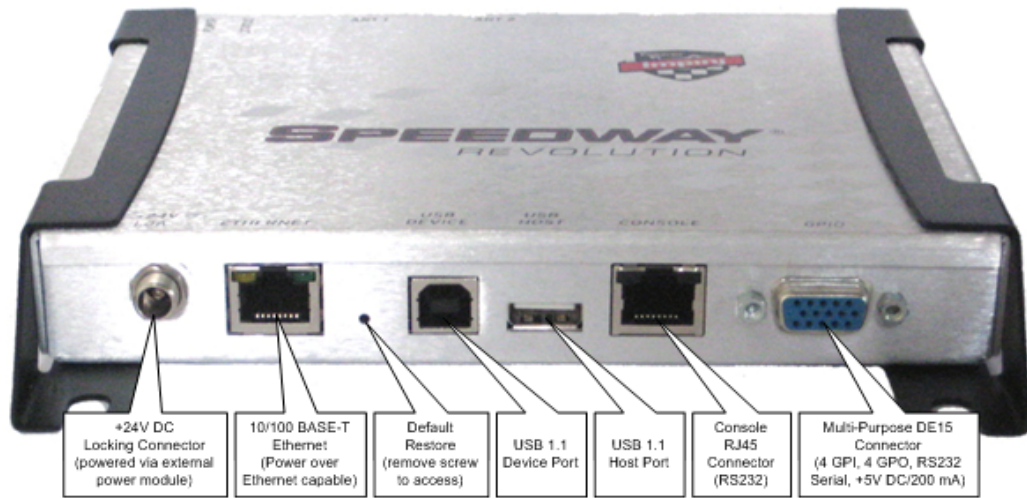


Figura 5.9: Pannello posteriore Reader Impinj Speedway Revolution

stop.

Per quanto riguarda la parte hardware il Reader Impinj Speedway Revolution usato in laboratorio mette a disposizione due connessioni seriali. La prima accessibile attraverso la porta GPIO usando i pin 2 (RS232 RX) e 3 (RS232 TX), vedi Fig.5.4. La seconda etichettata come Console nel pannello posteriore del Reader e accessibile attraverso un jack RJ-45, vedi Fig. 5.9, ed è quella utilizzata nei test in laboratorio. Quest'ultimo collegamento in realtà è un po' più complicato perché la comunicazione seriale richiede di essere attivata lato ricevitore, ora nella simulazione finale con il sistema di localizzazione il ricevitore come detto più volte è un oscilloscopio dotato di sistema operativo Windows, per cui comportandosi come un computer è possibile installare sopra programmi per la configurazione e attivazione di una comunicazione seriale. Per i test preliminari, invece, il ricevitore usato è un normale oscilloscopio da laboratorio quindi è stato necessario prendere un accorgimento. Semplicemente si è usato come ricevitore del trigger un computer in grado di attivare la comunicazione e si è sniffato il segnale ricevuto tramite la scheda hardware appositamente realizzata in Fig. 5.10 interposta nel collegamento Reader-Ricevitore.

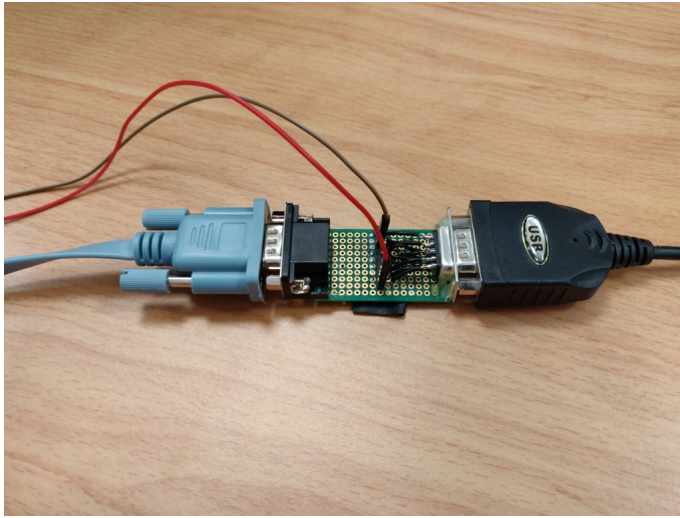


Figura 5.10: Scheda interposta nel collegamento Reader-Ricevitore

Tornando un attimo indietro al caricamento del programma nella memoria del Reader è doveroso scendere nei dettagli di tale operazione [2]. La memoria flash del Reader Impinj Speedway Revolution è rappresentata in Fig. 5.11. L'organizzazione della memoria è logicamente partizionata in due immagini per avere maggiore robustezza in caso di errori negli aggiornamenti del firmware. Ogni immagine è partizionata come mostrato in Fig. 5.11. SOP sta per System Operating Partition ed è il primo sistema di partizione del Reader Speedway, non è modificabile dallo sviluppatore e contiene software importanti come il kernel Linux, il firmware del FPGA, ecc. SSP sta per System Persistent Partition, non è modificabile dallo sviluppatore e i file memorizzati in questa partizione sono generati automaticamente durante l'esecuzione dei software sul Reader. Il contenuto del SSP sono file di configurazione, file di log e crash. CAP sta per Custom Application Partition ed è la partizione della memoria dedicata alle applicazioni custom per cui è modificabile dallo sviluppatore. Il programma per la gestione dell'inventario e la generazione del trigger viene caricato su questa partizione usando la pagina web del Reader raggiungibile

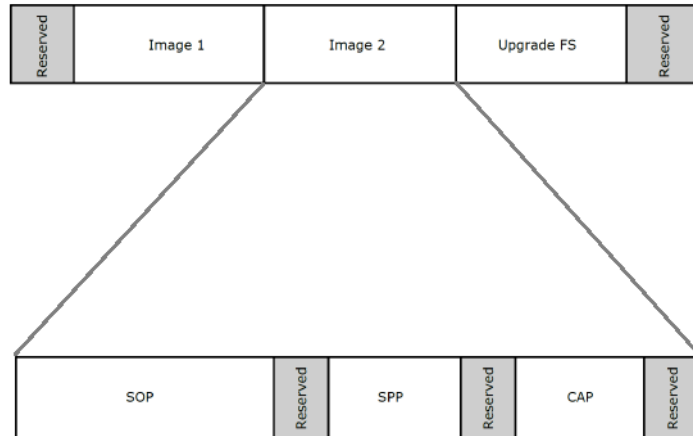




Figura 5.11: Organizzazione memoria flash Reader Impinj

tramite indirizzo IP del device, vedi Fig.5.12.

Infine ultimi due aspetti degni di nota, il primo è che si è messo mano al firmware del Tag per far sì che invii un solo EPC ad ogni sequenza d'inventario in modo da avere un solo trigger e il più possibile vicino alla ricezione dell'EPC. Il secondo è l'utilizzo della modalità *Low\_Latency* nell'invio dei *RO\_ACCESS\_REPORT*. *ImpinjReportBufferConfiguration* è un parametro custom di Impinj Octane LLRP per configurare la frequenza di invio dei report salvati nel buffer del Reader [5]. Nella modalità di default "Normal" il Reader memorizza i report internamente prima della trasmissione mentre nella modalità "Low\_Latency" i report vengono inviati immediatamente. Nel sistema in esame è preferibile quest'ultima modalità perché ad ogni *RO\_ACCESS\_REPORT* contenente un EPC corrisponde un trigger per cui se il Reader colleziona i report si rischia di avere un serie di report seguita da una sequenza di trigger senza una precisa corrispondenza.





READER INFORMATION		READER UPGRADE	
<b>Reader Name</b>	SpeedwayR-11-1F-C0	<b>Upgrade Status</b>	Ready
<b>Uptime</b>	0 Days, 0 hours, 27 minutes, 4 seconds	<b>Last Operation Status</b>	N/A
<b>System Time</b>	Wed Sep 11 09:34:12 UTC 2019	<b>Select Upgrade File</b>	<input type="button" value="Browse..."/> No file selected.
<b>LLRP Status</b>	Disconnected	<b>Upgrade Now</b>	<input type="button" value="Upgrade"/>
<b>RFID Status</b>	Idle		
<b>STP Status</b>	Not Activated		

DETAILED INFORMATION		READER REBOOT	
<b>Model Name</b>	Speedway R420	<b>Reboot Status</b>	Ready To Reboot
<b>Regulatory Region</b>	ETSI 302-208 v1.2.1	<b>Press to Reboot</b>	<input type="button" value="Reboot"/>
<b>MAC Address</b>	00:16:25:11:1F:C0		
<b>Software Version</b>	4.8.3.240		
<b>Hardware Version</b>	270-004-000		
<b>Application SW Version</b>	1.2.3.7		
<b>Serial Number</b>	370-14-35-0206		

NETWORK INFORMATION		QUICK LINKS	
<b>IP Address</b>	169.254.28.229	<a href="http://www.impinj.com">www.impinj.com</a>	
<b>Network Mask</b>	255.255.0.0	<a href="http://support.impinj.com">support.impinj.com</a>	
<b>Default Route</b>	169.254.0.1	<a href="#">Speedway Installation &amp; Operations Guide</a>	
<b>Broadcast Address</b>	169.254.255.255		

© 2010,2011 Impinj, Inc., All rights reserved

Figura 5.12: Pagina Web del Reader raggiungibile con indirizzo IP

Tabella 5.1: Distanza tra EPC e trigger

2,33ms	2,38ms	2,27ms	2,34ms	2,30ms	2,31ms
2,34ms	2,42ms	2,35ms	2,37ms	2,34ms	2,30ms
2,32ms	2,34ms	2,34ms	2,33ms	2,35ms	2,29ms
2,34ms	2,30ms	2,30ms	2,33ms	2,30ms	2,33ms
2,33ms	2,30ms	2,33ms	2,34ms	2,37ms	2,37ms
2,30ms	2,33ms	2,34ms	2,42ms	2,30ms	2,34ms
2,30ms	2,29ms	2,79ms	2,34ms	2,32ms	2,40ms
2,30ms	2,34ms	2,63ms	2,32ms	2,29ms	2,39ms

### 5.2.1 Risultati e Problemi

Avendo modificato il Tag per inviare un solo EPC ci si aspetterebbe un avvicinamento temporale tra EPC e trigger. In realtà in prima istanza non cambia niente rispetto alla situazione precedente con quindici EPC, questo perché è necessario andare a ridurre anche la trasmissione degli RN16, altrimenti è vero che si trasmette un solo EPC ma si continuano a trasmettere tutti gli RN16 per cui il trigger verrà generato solo alla fine di tale trasmissione. Una volta modificato il firmware alla fine della sequenza d'inventario si avrà un RN16 accompagnato dal suo corrispondente EPC al quale seguirà il corrispettivo trigger che ora è più vicino nel tempo, vedi Fig. 5.13.

Per quanto riguarda il ritardo nella generazione del trigger si assesta intorno a 2ms come si osserva in Fig. 5.13 per cui c'è un miglioramento come ci si aspettava dal momento in cui è stato limitato l'uso di protocolli lenti come LLRP e sono state utilizzate funzioni low-latency.

Il Jitter, che come già è stato detto è la criticità più importante da eliminare, in questa soluzione è notevolmente ridotto. La tabella 5.1 mostra una serie di tempi misurati all'oscilloscopio come distanza tra l'EPC e lo start bit del trigger al fine di valutare come varia tale intervallo. Dai dati emerge un Jitter mediamente intorno alle decine di  $\mu s$ , un valore più che accettabile per il funzionamento del sistema. Una volta risolto il problema del Jitter, per verificare che il ritardo non sia eccessivo, l'ultimo aspetto da controllare

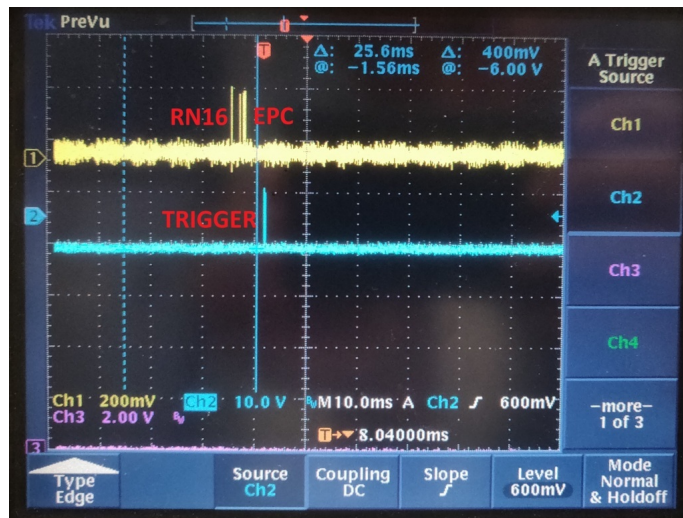


Figura 5.13: In giallo EPC, in blu trigger

è accertarsi che il trigger venga generato quando ancora il Tag è attivo. Come si vede dalla Fig. 5.14 il trigger è trasmesso perfettamente all'interno della finestra di alimentazione.

C'è un accorgimento che riguarda la forma del trigger degno di nota. Mentre con opportuni messaggi LLRP eravamo in grado di generare un vero e proprio trigger, si veda Fig. 5.8, un impulso rettangolare rappresentabile come un 1 logico. Nella comunicazione seriale, invece, c'è un protocollo di comunicazione che prevede uno start bit, un certo numero di bit di dato e almeno un bit di stop (il bit di parità può essere omesso) per questo motivo non si ha un impulso rettangolare ma una parola di qualche bit. Nel particolare si veda la Fig. 5.16 in cui viene mostrato il trigger utilizzato. Tutto ciò non è un problema per il funzionamento del sistema perché i due fronti di salita sono ad una distanza di qualche decina di microsecondi e da una parte l'oscilloscopio richiede circa un secondo per eseguire il codice Matlab mentre il generatore di segnali è sì più veloce, ma dovendo generare una sequenza di 4000 impulsi equi-spaziati di 50ns impiega  $200\mu\text{s}$  prima di essere pronto a rilevare un nuovo trigger.

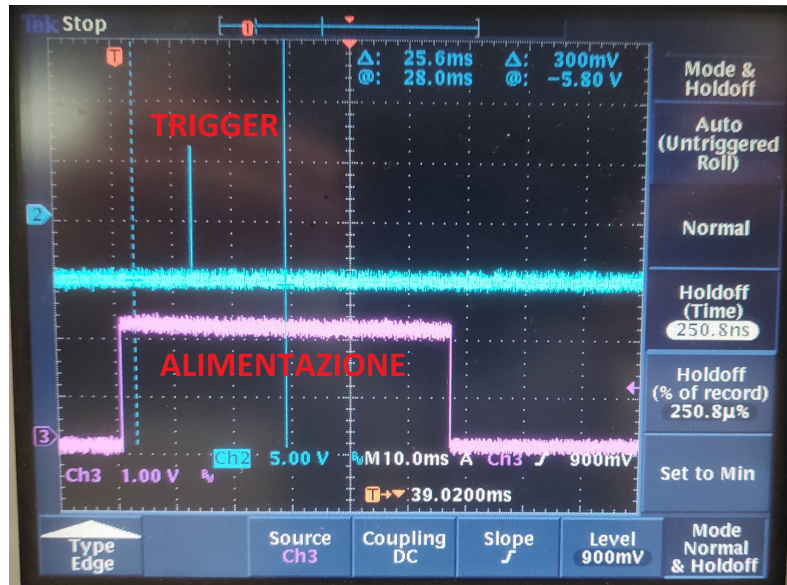


Figura 5.14: In blu il trigger, in rosa l'alimentazione del Tag



Figura 5.15: In giallo EPC, in blu trigger, in rosa l'alimentazione



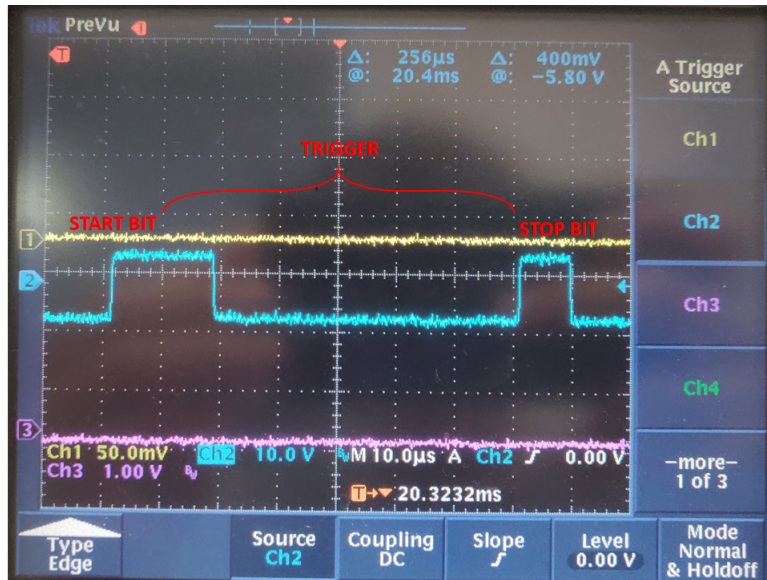


Figura 5.16: Forma del trigger

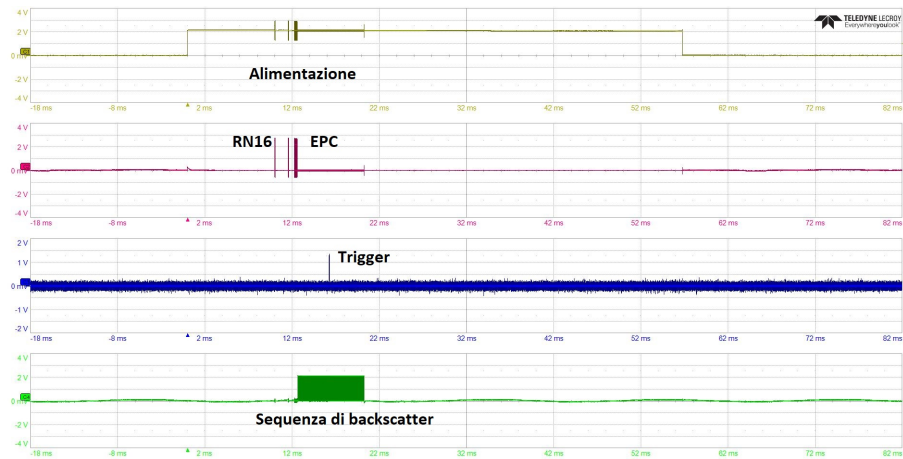


Figura 5.17: Relazioni temporali tra i vari segnali



# Capitolo 6

## Test finale

Fino ad ora i test sono stati effettuati usando il setup mostrato in Fig. 6.1:

- Reader Impinj Speedway R420, Fig. 6.2,
- Antenna UHF collegata al Reader,
- Tag custom, Fig. 6.2, appositamente realizzato per implementare il protocollo RFID,
- Oscilloscopio Tektronix TDS3034B per analizzare le forme d'onda trasmesse come EPC e trigger.

Tutte le prove sono state realizzate senza una vera e propria sincronizzazione di sistema mancando tutta la parte di trasmissione di impulsi UWB.

Una volta verificato la corretta trasmissione del trigger, l'ultima fase consiste nel mettere in piedi il sistema vero e proprio contenente sia la parte UHF di identificazione che quella UWB per la localizzazione e constatare che il trigger riesca effettivamente a interfacciare i due mondi.

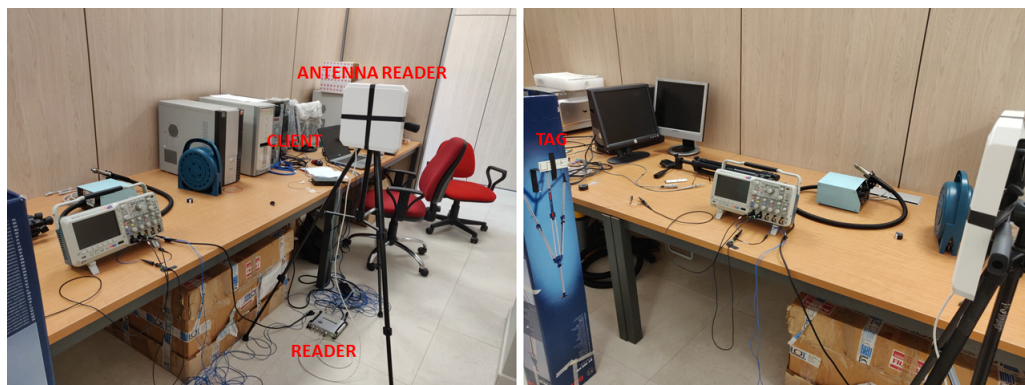


Figura 6.1: Setup utilizzato per testare il trigger



Figura 6.2: A sinistra Tag custom. A destra Reader Impinj Speedway R420

In Fig.6.3 è presente il sistema nel suo complesso con tutte le apparecchiature e dispositivi d'interesse. Di seguito viene spiegato per punti il funzionamento:

- Il Client dà inizio all'inventario attraverso una connessione SSH con il Reader. Sempre il Client fa anche da ricevitore del trigger quindi attiva la comunicazione seriale con il Reader;
- Il Reader inizia la sessione d'inventario interrogando il Tag;
- Il Tag grazie alla CW inviata dal Reader si attiva e inizia a trasmettere l'EPC e la sequenza di backscatter per la localizzazione;
- Quando il Reader riceve l'EPC trasmette sulla seriale il trigger che verrà ricevuto dal Client;
- L'oscilloscopio e il generatore di segnali prelevano il trigger inserendosi nella comunicazione Reader-Client;
- Il generatore ricevendo il trigger inizia a trasmettere impulsi UWB attraverso l'antenna a lui collegata;
- Il Tag ancora attivo è in grado di modulare la retro-diffusione degli impulsi UWB ricevuti;
- Nello stesso istante l'oscilloscopio una volta ricevuto il trigger apre la finestra di campionamento del segnale ricevuto sull'antenna ad esso collegata (tra l'antenna e l'oscilloscopio è presente un filtro passa-banda che seleziona le frequenze UWB a 4GHZ ed elimina la parte UHF).
- L'oscilloscopio con l'apposito algoritmo di localizzazione nell'arco di un secondo mostra la posizione del Tag in un'apposita finestra Matlab. In realtà come già detto nel caso di configurazione monostatica non verrà mostrata la posizione esatta ma solamente un'ellisse su cui si trova il Tag.

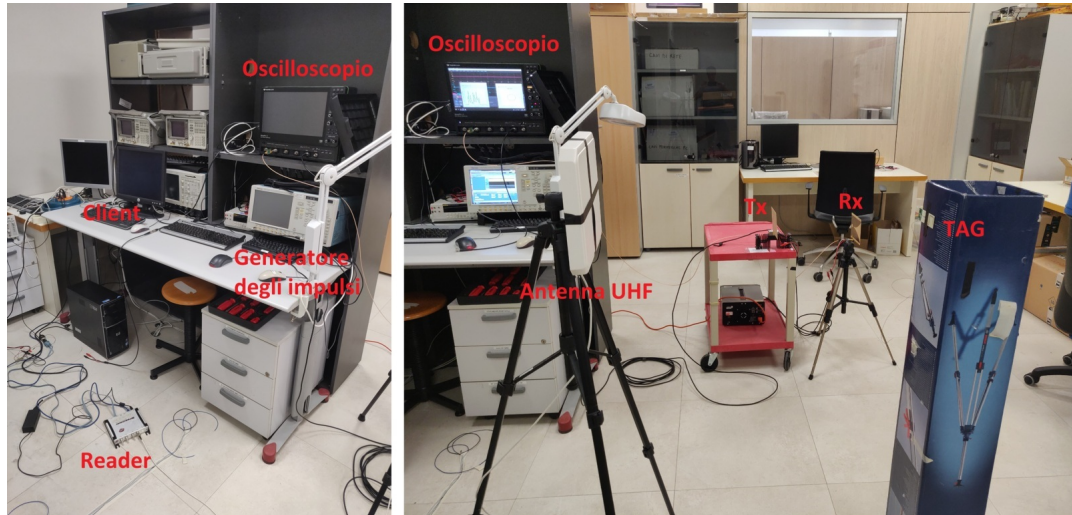


Figura 6.3: Sistema reale con parte RFID e UWB



Figura 6.4: A sinistra trasmettitore e ricevitore UWB. A destra Oscilloscopio TELEDYNE sopra, generatore Tektronix di impulsi UWB sotto.

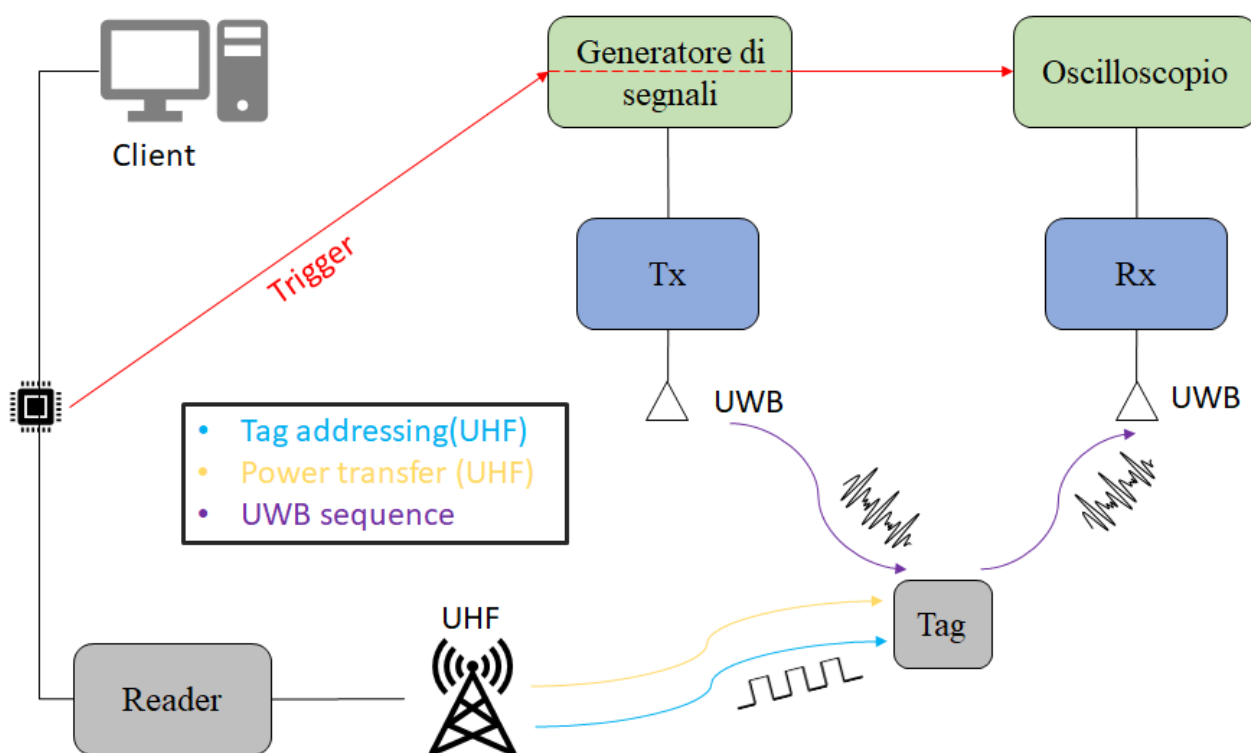


Figura 6.5: Schematico dell'architettura in Fig. 6.3

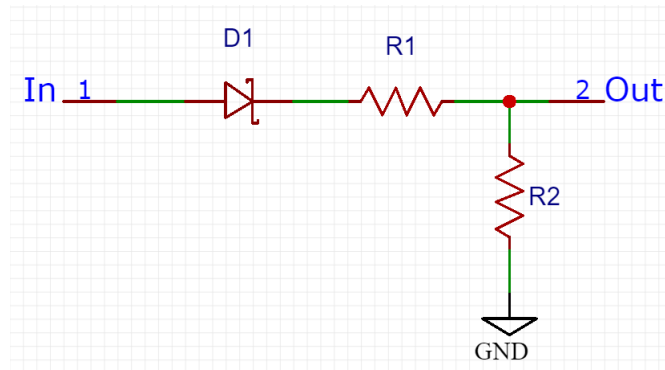


Figura 6.6: Circuito

## 6.1 Accorgimenti

Per il corretto funzionamento è stato necessario ridurre l'ampiezza del trigger che era ben oltre i 5V di valore efficace richiesti in ingresso all'oscilloscopio. Si è utilizzato un circuito appositamente realizzato in Fig. 6.6, composto da un diodo Schottky seguito da un partitore resistivo. Il diodo va a tagliare le tensioni negative mentre il partitore in base al valore delle resistenze attenua l'ampiezza del segnale.

Quando in ingresso si ha una tensione negativa, l'anodo risulta avere una tensione maggiore del catodo per cui il diodo è spento, non scorre corrente e l'uscita va a zero. Se l'ingresso è una tensione positiva, il diodo è acceso, scorre corrente e in uscita si avrà una tensione pari a  $\frac{I_n \cdot R_2}{R_1 + R_2}$  a meno della caduta di tensione sul diodo. Scegliendo opportunamente le resistenze si può limitare l'uscita a valori accettabili per l'oscilloscopio.

Il diodo Schottky, inoltre, garantisce bassa tensione di soglia e alta velocità di commutazione, quest'ultima importante perché bisogna garantire che in uscita il trigger rimanga tale, con ripidi fronti di salita e senza ritardi che vadano a smussare la forma del segnale.

In Fig.6.7 è presente una simulazione con Spice del circuito con valore delle resistenze di 50Ω. In ingresso si ha un'onda quadra con valore picco-picco di 10V e in uscita si vede un segnale con ampiezza ridotta e senza valori negativi.



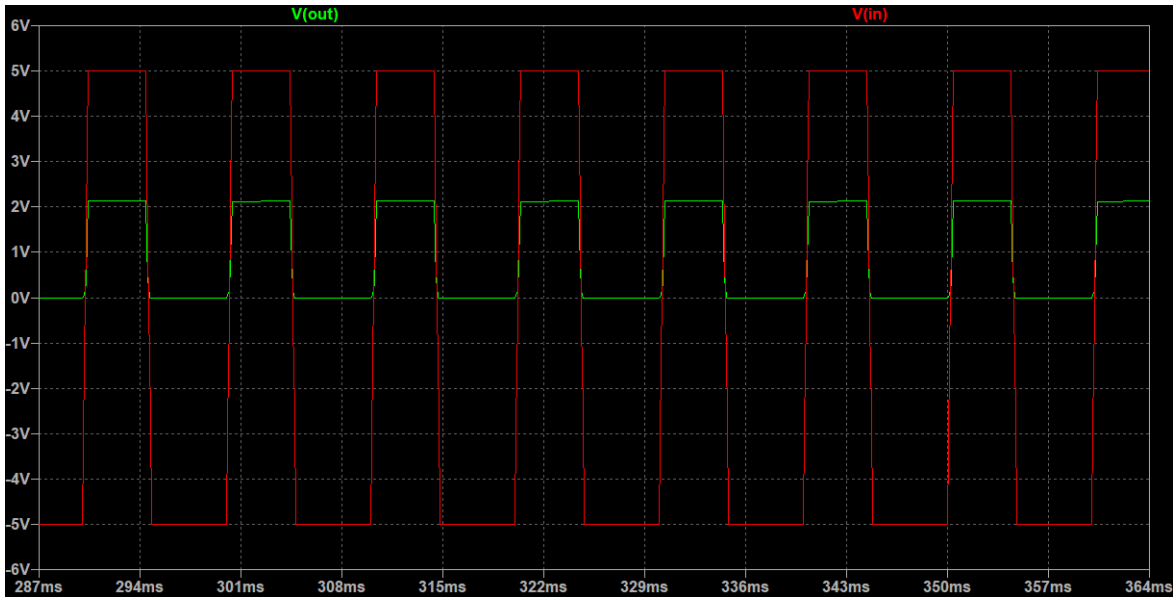


Figura 6.7: Simulazione Spice

## 6.2 Risultati sperimentali

Nel seguito vengono mostrati i risultati dei test effettuati sul sistema in Fig. 6.3.

In Fig. 6.8 nelle due finestre a sinistra vengono mostrati rispettivamente in alto il segnale del trasmettitore UWB ricevuto da Rx, mentre in basso il segnale di backscattering già pulito dal Clutter. La distanza tra i due indicatori rossi è esattamente la differenza di tempo tra gli impulsi originali e quelli riflessi dal Tag.

Nella finestra di destra, invece, viene mostrata la posizione dell'ellisse rispetto alle due antenne UWB, i due punti nel piano corrispondono al trasmettitore e ricevitore UWB, la curva tratteggiata mostra una distanza di riferimento per capire se l'ellisse arancione che dovrebbe rappresentare il Tag rispecchia effettivamente la reale posizione o è dovuta a fenomeni di Clutter.

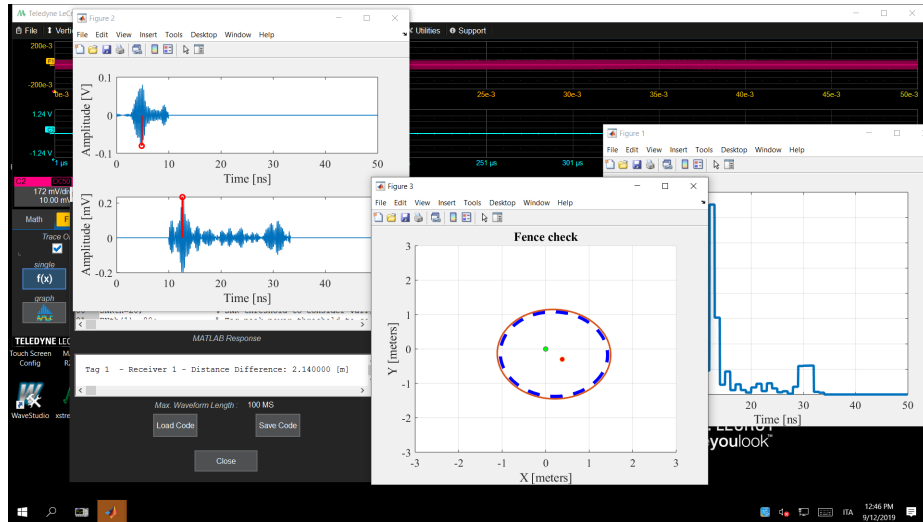


Figura 6.8: Tag posizionato vicino alla circonferenza di riferimento

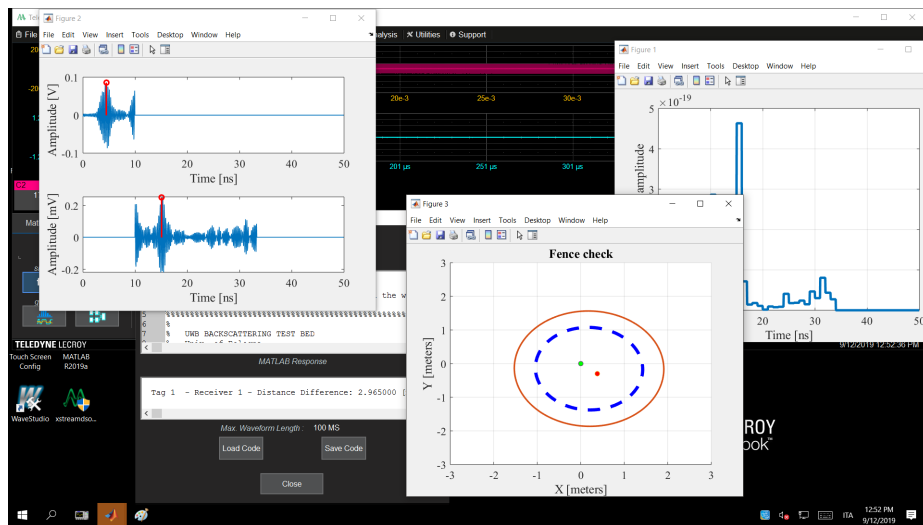


Figura 6.9: Tag posizionato all'esterno della circonferenza di riferimento

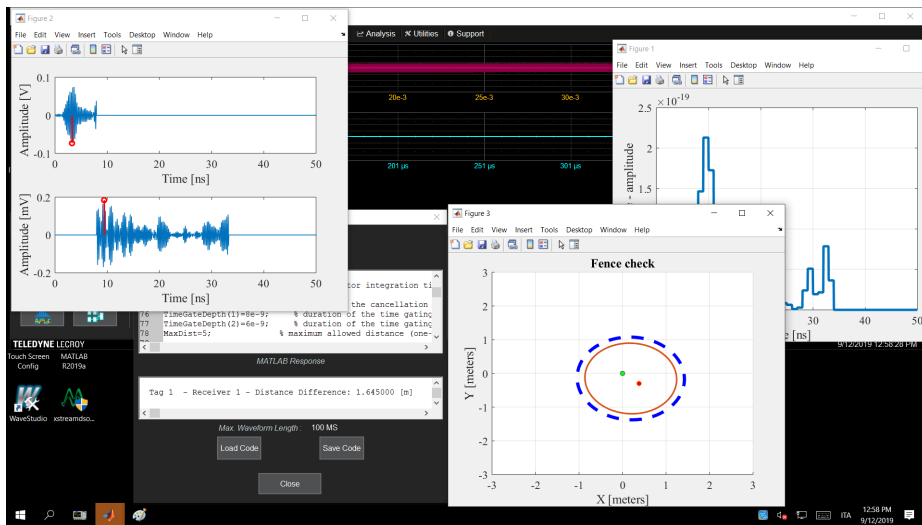


Figura 6.10: Tag posizionato all'interno della circonferenza di riferimento

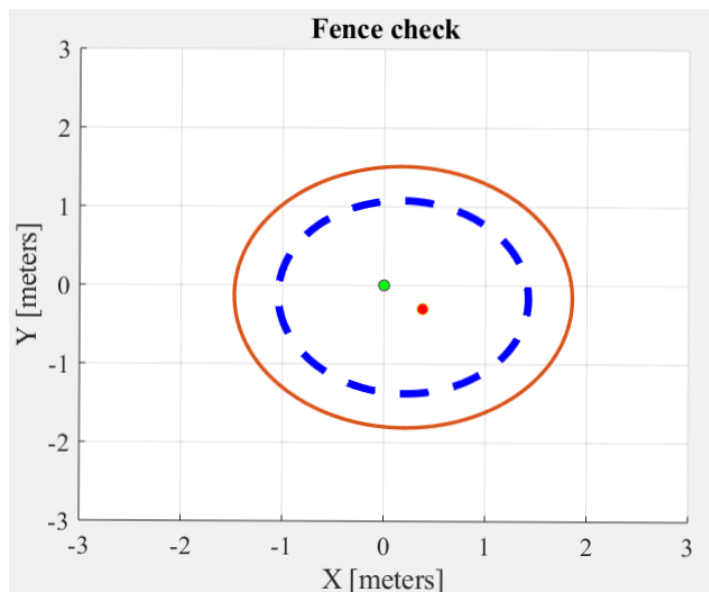


Figura 6.11: Piano in cui viene visualizzata la posizione del Tag



# Capitolo 7

## Conclusioni

Questa tesi ha presentato uno studio iniziale sull'interfacciamento di due sotto-sistemi per l'identificazione e la localizzazione di etichette elettroniche senza batterie. A parte una prima fase di studio dell'architettura del sistema e dei protocolli applicati ad esso, il lavoro è stato portato avanti per lo più in laboratorio cercando di sfruttare le porte di output in dotazione al Reader per generare un segnale di trigger verso oscilloscopio e generatore di segnali. La prima scelta ricaduta sulla programmazione della GPIO attraverso messaggi LLRP è risultata inadeguata mentre l'utilizzo della seriale con opportuni accorgimenti a livello di codice ha permesso la trasmissione del trigger nei tempi giusti. Per tale motivo l'obiettivo della tesi è stato raggiunto in quanto il sistema risulta sincronizzato e funzionante, ma lo studio della sincronizzazione non è concluso. Il ritardo di 2ms generato dalla seriale non è preoccupante fintanto che si ha a che fare con una decina di Tag, ma in un ipotetico impiego del sistema in un contesto con centinaia di Tag (si pensi ad un magazzino con migliaia di articoli da identificare), i 2ms di ritardo sommati non sarebbero più trascurabili e l'architettura risulterebbe troppo lenta nel localizzare e identificare le etichette. Questo problema non è stato possibile risolverlo in quanto il Reader non consentiva di operare a livello di firmware. Si è anche contattato il produttore del device chiedendo appunto se fosse possibile migliorare ulteriormente le latenze. La risposta ricevuta includeva come unica possibilità una modifica hardware al Reader fattibile, però, solo su grandi lotti di lettori e non su singoli dispositivi da

laboratorio. Tale risposta ha confermato l'impossibilità con il dispositivo a disposizione di ottenere risultati migliori.

Ovviamente il mercato dei Reader è ampio per cui è possibile rivolgersi ad altri produttori oltre Impinj, con la possibilità che esistano dispositivi che permettano di scendere maggiormente a basso livello controllando in modo più efficiente operazioni d'inventario e gestione delle porte di input/output. Esistono, per esempio Reader in grado di sfruttare il Bluetooth e Wi-Fi, in alternativa alla seriale, come il device VANCH in Fig.7.1 [11]. Altri dispositivi invece hanno come sistema operativo Android, Fig.7.2 [12], il che fa pensare ad una maggiore libertà di programmazione del Reader.

L'Auto-ID Labs del Politecnico federale di Zurigo ha creato Accada, una piattaforma open-source progettata per consentire agli utenti finali di sperimentare protocolli di rete EPCGlobal per sviluppare nuove applicazioni. La piattaforma include un modulo software di lettura che va eseguito su un apposito interrogatore EPC e implementa uno standard che permette ai Reader di comunicare con la piattaforma. Inoltre è disponibile un software, EPC Information Service, che riceve i dati RFID dalla piattaforma Accada li traduce in eventi resi disponibili al software di elaborazione. Tutto ciò permetterebbe di ricevere dati dal Reader senza utilizzo di seriale o GPIO ma sfruttando la comunicazione tra Reader e piattaforma. Per maggiori informazione si faccia riferimento a [14] in bibliografia

In conclusione va ricordato che il passo successivo alla sincronizzazione riguarda l'identificazione dei segnali retro-diffusi ricevuti dall'oscilloscopio. In un contesto con più Tag, l'oscilloscopio non solo deve ricevere il trigger dal Reader che lo avverte dell'attivazione del Tag, ma deve anche sapere di quale Tag si tratta per cui è necessario inviare oltre al trigger anche l'EPC che identifica univocamente l'etichetta. Come è visibile in Fig.3.2 in parallelo alla sincronizzazione c'è un altro collegamento tra Reader e oscilloscopio, ma in realtà la comunicazione dell'EPC può avvenire benissimo utilizzando la stessa seriale con cui si trasmette il trigger. Nella comunicazione seriale al massimo si possono trasmettere 8 bit di dato mentre l'EPC ha una lunghezza di 96 bit ciò non è un problema perché non tutti i 96 bit vengono usati per l'identificazione, nel sistema di interesse per esempio all'interno dell'EPC è contenuto anche il valore di temperatura rilevato dal sensore sul Tag. Quindi se la popolazione è inferiore a 256 (numero

di etichette univocamente identificabili con 8 bit) è sufficiente trasmettere solamente 8 dei 96 bit. Inoltre tale comunicazione viene a meno nel caso si riuscisse ad accedere direttamente al firmware del Reader. Ovviamente le soluzioni teoriche sono molteplici, bisognerà poi andare a vedere a livello pratico se vincoli di latenza e Jitter vengono rispettati; comunque sia tutto ciò viene lasciato come sviluppo futuro a questo lavoro di tesi.



Figura 7.1: Reader VI-83T/S Linux UHF RFID della VANCH [11]

Tabella 7.1: Specifiche VI-83T/S

<b>Dimensioni</b>	217(L)x217(W)x70(H) mm
<b>Peso</b>	2,5Kg
<b>Frequenze</b>	865-868MHz/902-920MHz
<b>Protocollo</b>	EPC C1 GEN2/ISO18000-6C
<b>RF Power</b>	30dBm
<b>Range</b>	Read range 3-5m, write range over 50cm
<b>Read rate</b>	oltre 50 tags/s
<b>CPU</b>	Adopt ATMEL AT91SAM9260 CPU
<b>Memoria</b>	16MB SDRAM, FLASH storage, 128M*8bits NandFlash
<b>OS</b>	Linux2.6.30
<b>Interfacce standard</b>	RS232, RS485, TCP/IP
<b>Interfacce opzionali</b>	Wiegand, CAN bus, WiFi, Bluetooth, 2G/3G/4G
<b>Rate porta seriale</b>	9600-115200bps
<b>RJ45</b>	10Mbps
<b>Interfaccia GPIO</b>	4 optically isolated inputs, 4 optically isolated outputs





Figura 7.2: Reader VF-S08 Android UHF RFID della VANCH [12]

Tabella 7.2: Specifiche VF-S08

<b>Dimensioni</b>	215(L)x185(W)x34(H) mm
<b>Peso</b>	1,5Kg
<b>Frequenze</b>	865-868MHz/902-920MHz
<b>Protocollo</b>	EPC C1 GEN2/ISO18000-6C/ISO 18000-6B
<b>RF Power</b>	20-30dBm±1dBm
<b>Range</b>	Read range 25m, write range 10m
<b>CPU</b>	RK3288 Cortex-A17 4-core processor, 1.8GHz
<b>RAM+ROM</b>	LPDDR2+eMMC 2GB/4GB+8GB/16GB/32GB
<b>Memoria</b>	Supporta SD card, 128GB massimo
<b>OS</b>	Android 5.1
<b>Interfacce standard</b>	RS232, RS485, TCP/IP
<b>Interfacce opzionali</b>	Wiegand, CAN bus, WiFi, Bluetooth, 2G/3G/4G
<b>Interfaccia GPIO</b>	4 optically isolated inputs, 4 optically isolated outputs
<b>Online upgrade</b>	Supporta sistema di upgrade remoto



Figura 7.3: Reader Impinj Speedway Revolution R420 usato in laboratorio [10]

Tabella 7.3: Specifiche R420

<b>Dimensioni</b>	190(L)x175(W)x30(H) mm
<b>Peso</b>	0,7Kg
<b>Frequenze</b>	865-868MHz/902-920MHz
<b>Protocollo</b>	EPC C1 GEN2/ISO18000-6C
<b>RF Power</b>	10-30dBm, 10-32dBm con alimentatore esterno
<b>Read Distance</b>	non pubblicata
<b>Read Rate</b>	fino a 1100 tag al secondo
<b>Interfacce standard</b>	RS232, ethernet
<b>Interfaccia GPIO</b>	4 optically isolated inputs, 4 optically isolated outputs
<b>Online upgrade</b>	Supporta sistema di upgrade remoto tramite IP

# Elenco delle figure

2.1	Architettura RFID . . . . .	5
2.2	Possibile scenario futuro: un utente con il suo smartphone può localizzare e interagire con gli oggetti taggati presenti nell'ambiente [7] . . . . .	5
3.1	Architettura del sistema generalizzata e multi-statica . . . . .	11
3.2	Architettura del sistema nel caso monostatico presa come riferimento in laboratorio . . . . .	11
3.3	Schematico a blocchi del UWB/UHF RFID Tag [8] . . . . .	12
3.4	Circuito del Tag [8] . . . . .	13
3.5	Sistema di localizzazione con impulsi UWB. L'impulso di sinistra si riferisce al segnale non modulato dalla sequenza ricevuto direttamente dal trasmettitore, mentre quello a destra si riferisce alla componente riflessa dal Tag e modulata dalla sequenza di codice. . . . .	15
3.6	Architettura con un solo trasmettitore e ricevitore . . . . .	17
3.7	Architettura di un Reader . . . . .	18
4.1	Protocolli di trasmissione in un sistema RFID . . . . .	20
4.2	Mappa della memoria di un Tag RFID Gen2 . . . . .	21
4.3	Struttura binaria di un messaggio LLRP . . . . .	22
4.4	Messaggi LLRP scambiati tra Client e Reader durante una sessione d'inventario [13] . . . . .	28
5.1	Segnali UHF e UWB . . . . .	30
5.2	Comunicazione d'inventario tra Reader e Tag . . . . .	31

5.3	Inventario visto all'oscilloscopio . . . . .	31
5.4	Porta GPIO Reader Impinj . . . . .	32
5.5	Funzione <b>sendGPOTrigger</b> che genera e trasmette il trigger . . . . .	34
5.6	Interfaccia grafica per il controllo del Reader . . . . .	34
5.7	In giallo gli EPC, in blu i trigger . . . . .	38
5.8	In rosa gli EPC, in giallo il trigger . . . . .	39
5.9	Pannello posteriore Reader Impinj Speedway Revolution . . . . .	40
5.10	Scheda interposta nel collegamento Reader-Ricevitore . . . . .	41
5.11	Organizzazione memoria flash Reader Impinj . . . . .	42
5.12	Pagina Web del Reader raggiungibile con indirizzo IP . . . . .	43
5.13	In giallo EPC, in blu trigger . . . . .	45
5.14	In blu il trigger, in rosa l'alimentazione del Tag . . . . .	46
5.15	In giallo EPC, in blu trigger, in rosa l'alimentazione . . . . .	46
5.16	Forma del trigger . . . . .	47
5.17	Relazioni temporali tra i vari segnali . . . . .	47
6.1	Setup utilizzato per testare il trigger . . . . .	50
6.2	A sinistra Tag custom. A destra Reader Impinj Speedway R420 . . . . .	50
6.3	Sistema reale con parte RFID e UWB . . . . .	52
6.4	A sinistra trasmettitore e ricevitore UWB. A destra Oscilloscopio TELEDYNE sopra, generatore Tektronix di impulsi UWB sotto. . . . .	52
6.5	Schematico dell'architettura in Fig. 6.3 . . . . .	53
6.6	Circuito . . . . .	54
6.7	Simulazione Spice . . . . .	55
6.8	Tag posizionato vicino alla circonferenza di riferimento . . . . .	56
6.9	Tag posizionato all'esterno della circonferenza di riferimento . . . . .	56
6.10	Tag posizionato all'interno della circonferenza di riferimento . . . . .	57
6.11	Piano in cui viene visualizzata la posizione del Tag . . . . .	57
7.1	Reader VI-83T/S Linux UHF RFID della VANCH [11] . . . . .	62
7.2	Reader VF-S08 Android UHF RFID della VANCH [12] . . . . .	63
7.3	Reader Impinj Speedway Revolution R420 usato in laboratorio [10] . . . . .	64

# Elenco delle tabelle

5.1	Distanza tra EPC e trigger . . . . .	44
7.1	Specifiche VI-83T/S . . . . .	62
7.2	Specifiche VF-S08 . . . . .	63
7.3	Specifiche R420 . . . . .	64



# Bibliografia

- [1] Klaus Finkenzeller, "*RFID HANDBOOK*", Third edition, WILEY, 2010.
- [2] Impinj, "*Reader and Gateway Embedded Developers Guide*", version 5.4.0, 2012. [Online]. Available: <https://support.impinj.com/hc/en-us/articles/202755298-Reader-Documentation>, voice *Download Reader and Gateway Embedded Developers Guide*.
- [3] EPCglobal, "*Low Level Reader Protocol (LLRP)*", version 1.0.1, 2007.
- [4] GS1, "*EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Standard*", 2018.
- [5] Impinj, "*Octane LRP*", version 6.0.0, 2012. [Online]. Available: <https://support.impinj.com/hc/en-us/articles/202755298-Reader-Documentation>, voice *Download Octane LLRP Guide*.
- [6] Anna Guerra, Davide Dardari, Nicolò Decarli, Jurgis Aleksandravicius, "*High-Accuracy Localization of Backscattering UWB Tags: Implementation and Experimental Results*", article, 2017 IEEE International Conference on RFID Technology Application (RFID-TA), Sept 2017.
- [7] Anna Guerra, Davide Dardari, Francesco Guidi, Nicolò Decarli, "*The Future of Ultra-Wideband Localization in RFID*", article, 2016 IEEE International Conference on RFID (IEEE RFID 2016), Orlando, USA, May 2016.

- [8] Aldo Romani, Alessandra Costanzo, Anna Guerra, Davide Dardari, Davide Fabbri, Diego Masotti, Marco Fantuzzi, Nicolò Decarli, "*An Ultra-wideband Battery-less Positioning System for Space Applications*", article, RFID-TA 2019 Conference, Pisa, Italy, Sept 2019.
- [9] Aldo Romani, Alessandra Costanzo, Anna Guerra, Davide Dardari, Davide Fabbri, Marco Fantuzzi, M. Del Prete, Matteo Pizzotti, Nicolò Decarli, Jurgis Aleksandravicius, "*Energy Autonomous UWB Localization*", article, IEEE Journal of Radio Frequency Identification, vol. 1, no. 3, pp. 228–244, Sept 2017.
- [10] —, "<https://www.atlasrfidstore.com/impinj-speedway-revolution-r420-uhf-rfid-reader-4-port/>".
- [11] —, "<http://www.vanch.net/product/detail/263.html>".
- [12] —, "<http://www.vanch.net/product/detail/259.html>".
- [13] Beatrice Borghetti, "*Sistemi per applicazioni ciclistiche*", thesis, University of Bologna, Cesena, 2016.
- [14] Mark Roberti, "*Auto-ID Lab Releases Accada RFID Prototyping Platform*", article, RFID Journal.
- [15] A. Costanzo, D. Masotti, M. Fantuzzi, M. Del Prete, "*Co-design strategies for energy-efficient UWB and UHF wireless systems*", article, IEEE Transactions on Microwave Theory and Techniques, vol. 65, no. 5, pp.1852–1863, May 2017.
- [16] D. Dardari, R. D’Errico, C. Roblin, A. Sibille, and M. Z. Win, "*Ultra-wide bandwidth RFID: The next generation?*" Proc. IEEE, vol. 98, no. 9, pp. 1570-1582, Sep 2010, special Issue on RFID - A Unique Radio Innovation for the 21st Century.
- [17] F. Guidi, N. Decarli, D. Dardari, F. Natali, E. Savio-li, and M. Bottazzi, "*A low complexity scheme for passive UWB-RFID: Proof of concept*", IEEE Communications Letters,



- vol. 20, no. 4, pp. 676-679, April 2016. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/7407304/>.
- [18] N. Decarli, F. Guidi, and D. Dardari, "*Passive UWB RFID for tag localization: Architectures and design*", IEEE Sensors J., vol. 16, no. 5, pp. 1385-1397, March 2016.
- [19] F. Guidi, C. Roblin, A. Sibille, and D. Dardari, "*Analysis of UWB tag backscattering and its impact on the detection coverage*", IEEE Trans. Antennas Propag., vol. 62, no. 8, pp.4292-4303, Aug. 2014.
- [20] M. Z. Win, D. Dardari, A. F. Molisch, W. Wiesbeck, and Z. Jinyun, "*History and applications of UWB [scanning the issue]*", Proc. IEEE, vol. 97, no. 2, pp. 198-204, Feb. 2009, special Issue on UWB Technology & Emerging Applications.
- [21] D. Dardari, F. Guidi, C. Roblin, and A. Sibille, "*Ultra-wide bandwidth backscatter modulation: Processing schemes and performance*", EURASIP Journal on Wireless Communications and Networking, vol. 2011:47, no. 1, pp. 1-15, Jul. 2011. [Online]. Available: <http://dx.doi.org/10.1186/1687-1499-2011-47>
- [22] F. Guidi, N. Decarli, S. Bartoletti, A. Conti, and D. Dardari, "*Detection of multiple tags based on impulsive backscattered signals*", IEEE Trans. Commun., vol. 62, no. 11, pp. 3918-3930, Nov 2014.
- [23] D. Dardari, "*Detection and accurate localization of harmonic chipless tags*", EURASIP Journal on Advances in Signal Processing, vol. 2015, no. 77, pp. 1-13, 2015.