

Università degli Studi di Camerino

Scuola di Scienze e Tecnologie

Corso di Laurea in Informatica classe L-31



STUDIO E REALIZZAZIONE DI UN SISTEMA PER IL TRACCIAMENTO DI OGGETTI TRAMITE IDENTIFICATORI RFID ATTIVI

Tesi di Laurea sperimentale

Studente:

Giuliano Jr. Alberti

Relatore:

Dott. Fausto Marcantoni

Correlatore:

Dott. Francesco De Angelis

Anno Accademico 2009/2010 - Esame Finale del 22/07/2010

Dedicata a...

Maresciallo Capo Cav. Uff. Mario Tanziani

Valentina De Amicis

Ai miei Genitori

Ringraziamenti

Giunto al termine di questo lavoro desidero ringraziare ed esprimere la mia riconoscenza nei confronti di tutte le persone che, in modi diversi, mi sono state vicine e hanno permesso e incoraggiato sia i miei studi che la realizzazione e stesura di questa tesi. I miei più sentiti ringraziamenti vanno a chi mi ha seguito durante la redazione del lavoro di tesi:

- *Prof. Fausto Marcantoni*, per la fiducia fin da subito dimostratami nell'aver accettato questo argomento di tesi e per avermi seguito durante lo svolgimento del lavoro aiutandomi a intraprendere le scelte più appropriate.
- *Dott. Francesco De Angelis*, per la continua disponibilità e prontezza nei chiarimenti e suggerimenti, per la rilettura critica di tutti i capitoli della tesi e per avermi guidato con i suoi suggerimenti durante la conclusione di questo percorso formativo. Gli insegnamenti e gli incoraggiamenti offertimi hanno contribuito a farmi credere nelle mie possibilità e perseguire le mie ambizioni. Non scorderò la continua disponibilità e pazienza che mi ha dedicato

durante l'intero periodo di tesi, gli incoraggiamenti, i consigli e al tempo stesso gli apprezzamenti per quanto realizzato.

Rimarrà in me il piacevole ricordo di questi anni di studio che ho trascorso, “anche se non a tempo pieno”, in questo dipartimento e per aver trovato quasi sempre professori disponibili al dialogo e a confrontarsi con le idee altrui, qualità non da tutti. Desidero ringraziare tutti i docenti che hanno contribuito, con consigli e suggerimenti, al raggiungimento di questo traguardo, per me importante e in modo particolare:

- *Prof. Emanuela Merelli*, è stata la prima a farmi riflettere sul mio futuro, presentandomi le diverse possibilità, incoraggiandomi e alimentando in me nuove sfide e conseguenti motivazioni.

Per ultimi, ma di certo non per importanza, ringrazio la mia **famiglia**, la mia **ragazza** e gli **amici** che mi sono stati molto vicini in tutti questi anni “da studente”, che oltre ad avermi sempre “supportato” mi hanno più di tutto “sopportato”:

- Il mio primo pensiero, ovviamente, va ai miei *genitori*, a cui dedico questo lavoro di tesi: senza il loro aiuto non avrei mai raggiunto questa meta. Sono davvero grato per tutto il sostegno economico, ma più di ogni altra cosa di quell'aiuto tacito o esplicito che è venuto dal loro cuore. Mi auguro che tutti i sacrifici spesi siano in questo modo, almeno in parte, ripagati.
- Un pensiero particolare va a mio *nonno Mario* che desiderava per me, questo prezioso traguardo più di ogni altra cosa; non scorderò mai i suoi continui incoraggiamenti; l'unico mio grande rammarico è che non posso condividere con lui questa immensa gioia.
- *Valentina* che con estrema pazienza ha sopportato i miei sbalzi di umore e le mie paranoie quando, sotto stress, mi sfogavo in modo particolare con lei. Se ho raggiunto questo traguardo lo devo anche alla sua continua presenza, per avermi fatto capire che potevo farcela, incoraggiandomi a “non mollare mai”. Ripenso a tutte quelle

volte che mi ha incoraggiato vedendomi preso dai libri, da un esame e da questa tesi, ma soprattutto per la soddisfazione che ha saputo donarmi dimenticandosi del mio “caratterino”, ultimamente non di certo facile.

Come non ringraziare tutti gli **Amici dell’Università** e in modo particolare i miei “*Compagni di corso*” con i quali ho condiviso più da vicino questi anni di studio (ma anche di piacevoli svaghi). Ora, più di tutti, possono comprendere il mio grado di soddisfazione: *Antonio Pucci, Luigi Santini, Sandro Nepi, Andrea Durastanti, Marco Camilli, Francesco De Angelis*; inoltre non posso dimenticare *Pierluigi, Gino, Federico, Tommaso Mauro, Claudia, Daniela, Antonio, Nazzareno, Roberto, Paolo, Silvio, Giorgio, Cesare, Monia, Emanuela, Amelio, Enrico, Elya, Antonio Ezio...*

Desidero ringraziare tutte quelle persone con cui ho iniziato e trascorso i miei studi, con cui ho scambiato qualche pensiero, qualche idea, qualche risata all’interno del dipartimento. In diversi modi hanno contribuito nel mio percorso formativo, aiutandomi a credere in me stesso, suscitando in me nuovi interessi e soprattutto mi hanno suggerito, direttamente o indirettamente, le modalità per poterli raggiungere. Infine un ringraziamento particolare va a *Sara Buratti* compagna di corso e amica.

Luglio 2010

Giuliano Jr.

Indice

1 Introduzione	13
1.2 Obiettivi.....	14
1.3 La struttura della tesi.....	15
1.3.1 QR Code.....	16
2 Tecnologie RFID passive	
2.1 RFID Radio Frequency IDentification	17
2.2 Vantaggi tecnologie RFID.....	21
2.3 Applicazioni degli RFID	24
2.3.1 Controllo presenze ed accessi.....	25
2.3.2 Monetica.....	26
2.3.3 Passaporto	26
2.3.4 Bigliettazione elettronica.....	27
2.3.5 Logistica magazzini	28
2.3.6 Logistica trasporti	28
2.3.7 Tracciamento pratiche.....	29
2.3.8 Assistenza e manutenzione.....	30
2.3.9 Identificazione degli animali	30
2.3.10 Antitaccheggio.....	31
2.3.11 Registro scolastico elettronico	32
2.3.12 Monitoraggio raccolta rifiuti	32
2.3.13 Rilevazione dei parametri ambientali.....	33
2.4 Problemi di privacy e sicurezza legati all'utilizzo della tecnologia RFID.....	33

3 Tecnologia RFID attiva

3.1 Panoramica	37
3.2 Il Tag.....	38
3.2.1 Funzionalità di base	39
3.3 Caratteristiche fisiche.....	40
3.3.1 Fonte di energia	41
3.3.2 Air Interface	41
3.3.3 Codifica.....	41
3.3.4 Anti-collisione.....	43
3.3.5 Identità	45
3.3.6 Antenna	45
3.3.7 Controller	46
3.3.8 Interfaccia di rete.....	47
3.3.9 Forme e dimensioni	47
3.4 RFID nella storia.....	48
3.5 Conclusione.....	49

4. La Tecnologia SNAP

4.1 SNAPpy	51
4.2 Portal e Snap connect.....	52
4.3 Panoramica	52
4.3.1 Remote Procedure Call	53
4.3.2 Snappy Scripting	53
4.4 Rete SNAP	54
4.5 Lo Stack SNAP.....	55
4.6 Portabilità	56
4.7 La macchina virtuale Snappy	56
4.8 Snap e reti wireless mesh	56
4.8.1 Reti wireless basate su SNAP	57

4.8.2 Sviluppo e distribuzione di applicazioni Snappy.....	57
4.8.3 Sleepy, ultra mesh e routig a bassa potenza	58

5 Prototipo di un sistema attivo di tracciamento

5.1 Panoramica	61
5.1.1 Hardware	62
5.1.2 Software	65
5.2 System design.....	66
5.3. System hardware.....	67
5.3.1 Tag hardware.....	67
5.3.2 Nodo Reader hardware.....	68
5.3.3 PC Reader hardware	69
5.3.4 SNAPstick hardware.....	70
5.4 System software.....	72
5.4.1 Tag software.....	72
5.4.2 Reader software	76
5.4.3 PC Reader software	77
5.4.4 Inoltro dei dati da seriale del PC in HTTP	79
5.4.5 Display GUI	82
5.5 Test ed analisi dei dati.....	84
Conclusioni.....	91
Appendice A	
Snappy Vs Python.....	93
Appendice B	
Portal software.....	99
Bibliografia.....	113



1 Introduzione

RFID, acronimo di *Radio Frequency IDentification* è un termine che indica l'insieme di tecnologie d'identificazione basate sulla trasmissione dei dati tramite radiofrequenza.

L'identificazione implica l'assegnazione di un'identità univoca a un oggetto che consenta di distinguerlo in modo non ambiguo. Il fine principale di questa tecnologia, pertanto, è di assumere da parte di un "identificatore", informazioni su oggetti, animali o persone identificati, per mezzo di piccoli apparati a radiofrequenza associati ai medesimi.

L'assunzione d'informazioni è relativa a operazioni di ricerca, identificazione, selezione, localizzazione spaziale e tracciamento. Identificatore e identificato comunicano mediante segnali a radio frequenza, quindi senza necessità di contatto fisico (a differenza, ad esempio, delle carte a banda magnetica) e senza che gli apparati siano né visibili (a differenza, ad esempio, dei codici a barre), né in visibilità reciproca (*non-line-of-sight*).

L'antenato degli RFID è comunemente riconosciuto nel sistema “*Identification Friend or Foe (IFF)*” sviluppato in Inghilterra durante la seconda guerra mondiale (1940). L'apparato a bordo degli aerei alleati, rispondeva, se interrogato, identificando così gli aerei alleati e distinguendoli da quelli nemici.

La tecnologia è poi evoluta in sistemi per seguire la rotta dei carri ferroviari, per l'automazione di processi nell'industria automobilistica, per la localizzazione del bestiame e degli animali selvatici, in agricoltura e nelle riserve naturali, per l'antitaccheggio nel commercio al minuto, per chiavi e documenti elettronici, ecc. Uno dei primi sistemi d'identificazione automatica è stato il codice a barre (*barcode label*), che scatenò una vera e propria rivoluzione nei sistemi d'identificazione automatica. Quest'ultimo, negli ultimi tempi, però, sta diventando inadeguato per moltissime applicazioni, pur essendo estremamente economico. Il suo limite principale è nella scarsa capacità di memorizzazione dei dati e nella impossibilità di essere riutilizzato.

La soluzione RFID è tecnicamente più vantaggiosa tramite la memorizzazione dei dati in un chip di silicio. La diffusione di questi sistemi è avvenuta principalmente dagli anni '90 in poi e attualmente esistono numerose soluzioni commerciali a costi contenuti.

1.2 Obiettivi

Quest'elaborato si pone come obiettivo quello di presentare una breve descrizione di questa tecnologia, di alcune delle sue possibili applicazioni e la realizzazione di un semplice sistema in grado di effettuare il tracciamento di oggetti o persone in un ambiente chiuso.

Questa tesi non vuole essere esaustiva, ma si è cercato di presentare, in modo sintetico, un insieme di concetti che tra qualche anno potrebbero diventare di uso comune nella nostra vita quotidiana.

1.3 La struttura della tesi

La tesi presenta nel Capitolo 2 una descrizione panoramica della tecnologia RFID passiva, delle sue possibili applicazioni in campo industriale, tecnico e amministrativo, e nella sfera dei servizi che riguardano l'uomo e la catalogazione dei prodotti, e di conseguenza, le modalità di gestione delle informazioni relative ai prodotti da parte di produttori, venditori e consumatori. Successivamente saranno presentati i principali ambiti di utilizzo dei servizi basati sulla tecnologia RFID.

Il Capitolo 3, estende il Capitolo 2 presentando una sintetica e generale descrizione dei sistemi RFID attivi, con particolare riguardo agli elementi architetturali che sono alla base della tecnologia che sarà utilizzata per lo sviluppo del progetto. Nel Capitolo 4, troviamo invece una descrizione generale della specifica tecnologia utilizzata per la creazione del progetto che è basato sui dispositivi per reti *mesh* realizzati dall'azienda "*Synapse Wireless*".

Il Capitolo 5 descrive il progetto realizzato di un sistema di tracciamento. Per ciascun elemento attivo sono presentate le caratteristiche peculiari e le modalità di funzionamento. Viene presentata l'architettura hardware e software del sistema e se ne delinea il funzionamento.

Le appendice A e B completano il lavoro offrendo un supporto per la configurazione e la programmazione dei dispositivi utilizzati.

1.3.1 QR Code

QR Code, è un codice a barre bidimensionale, ed è un'applicazione orientata verso la comodità, finalizzata a sollevare l'utente dal noioso compito di inserire i dati nel proprio telefono cellulare. Infatti, tramite la scannerizzazione di questo codice, è possibile avere sul telefonino, in maniera del tutto facile e immediata, link, mail, e tanti altri dati, senza doverli digitare sulla tastiera, la quale a volte risulta essere troppo piccola e scomoda oltre alla difficoltà oggettiva di inserire caratteri particolari, presenti su link molto lunghi e complessi.

Questa tesi, fa un largo uso della tecnologia, applicando un *QR Code* nella pagina iniziale di ogni capitolo, rendendolo così reperibile in rete e aggiungendo il codice nelle pagine dove sono presenti oggetti multimediali scaricabili (filmati esaustivi, ecc.) che possono essere utili ad ampliare l'elaborato proposto.



Codice bidimensionale della tesi pubblicata in rete



2 Tecnologie RFID passive

2.1 RFID Radio Frequency IDentification

RFID è l'acronimo di Radio Frequency Identification, ed è una tecnologia per la identificazione automatica di oggetti, animali o persone (*AIDC Automatic Identifying and Data Capture*) basata sulla capacità di memorizzare e accedere a distanza a dei dati usando dispositivi elettronici (chiamati TAG o transponder) che sono in grado di rispondere comunicando le informazioni in essi contenute quando "interrogati" con l'ausilio di frequenze radio o variazioni di campo magnetico. In un certo senso sono un sistema di lettura "senza fili" e a distanza, di informazioni contenute in un Tag RFID usando dei lettori RFID.

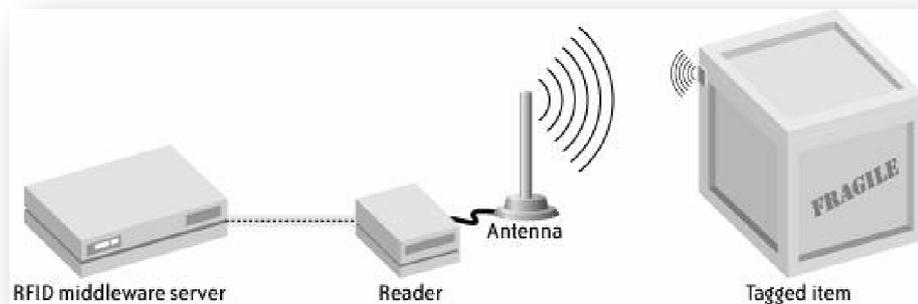


Figura 1 - Struttura RFID completa

Nello specifico, il sistema, è composto da : il Tag, che è il dispositivo di identificazione allegato alla voce che vogliamo monitorare; e il lettore (Reader), che è un dispositivo che può riconoscere la presenza di etichette RFID e leggere le informazioni memorizzate su di essi. Il lettore, dotato di un'antenna, emette un campo elettromagnetico con il quale può leggere i dati contenuti nel transponder e, se richiesto, può sovrascriverne dei nuovi; inoltre può informare un altro sistema della presenza degli elementi di Tag. Il sistema con cui il lettore comunica, di solito è un software che si frappone tra i lettori e le applicazioni; questo software si chiama *RFID middleware*. L'etichetta RFID, o Tag che dir si voglia, è costituito da un microchip che contiene dati (tra cui un numero univoco universale scritto nel silicio), una antenna, e una batteria, e può essere attivo o passivo. Il tipo passivo, contiene semplicemente un microchip, privo di carica elettrica, che viene "eccitato" al passaggio, da un lettore magnetico che emette un segnale radio a frequenze medie. Il microchip, viene alimentato dalla radiofrequenza la quale attraverso il principio di induzione, si trasforma in energia elettrica necessaria al chip per poter rispondere al lettore, ritrasmettendogli un segnale contenente le informazioni memorizzate nel chip.

2 Tecnologie RFID passive

Oltre ai transponder di tipo passivo esistono anche transponder di tipo attivo, i quali dispongono oltre che di un'antenna per inviare il segnale di lettura e ricevere le risposte anche e soprattutto di una batteria "for life" incorporata (quali i nuovi telepass), che permettono una ricetrasmisione a distanze maggiori in quanto amplificano il segnale in uscita dal transponder aumentandone così la potenza. Sempre più transponder sono dotati di caratteristiche anticollisione ossia ammettono il "bulk reading", la lettura in gruppo, cioè sono identificabili dal lettore anche quando passano in gruppo nell'area del lettore.

La forma e le dimensioni dei transponder sono le più svariate: dalla classica "a bottone" con diversi diametri, a quella cilindrica ridottissima (sistema di blocco d'avviamento delle vetture, riconoscimento animali), a quelle a forma di carta di credito (con o senza banda magnetica).

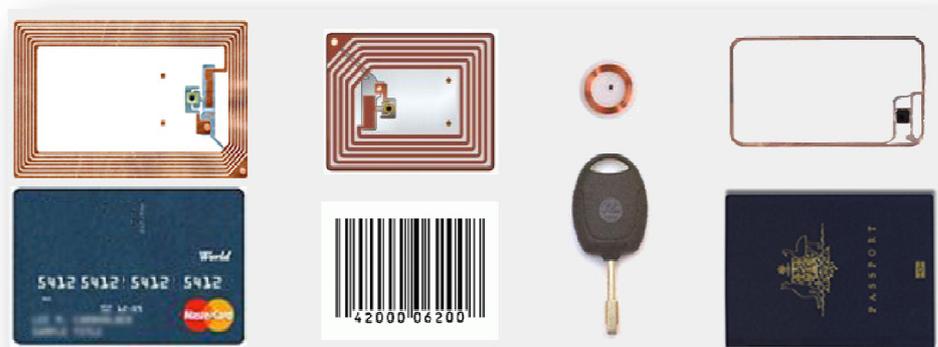


Figura 2 –Esempi di Tag di differenti forme e dimensioni

Come avviene per il lettore, anche per il Tag, più grande è l'antenna e maggiore è la distanza che si può raggiungere tra transponder e lettore.

Per la comunicazione fra etichetta e lettore sono utilizzate varie frequenze, attribuite da organizzazioni normatrici, che fissano delle regole molto precise per l'utilizzo di ciascuna di queste. La tecnologia RFID non può quindi prescindere da queste regolamentazioni, che sono parte integrante dell'ambiente in cui essa opera. Oggi, grazie alla significativa opera degli attori della normalizzazione, innanzitutto dell'ISO (*International Standard Organisation*), ma anche di singoli raggruppamenti di utenti, si possono avere sistemi RFID realmente interoperabili.

Tipi d TAG	VANTAGGI	SVANTAGGI	OSSERVAZIONI
Passivi	<ul style="list-style-type: none"> - Tempi di vita più lunghi - Vasta gamma di forme - Meccanicamente più flessibili - Basso costo 	<ul style="list-style-type: none"> - Distanze limitate a 4-5 m - Controllati rigorosamente dalle regolamentazioni locali 	I più usati nei sistemi RFID
Semipassivi	<ul style="list-style-type: none"> - Grande distanza di comunicazione - Può essere usato per controllare altri dispositivi come i sensori (Temperatura, pressione, etc.) - Non rientra nelle stesse regolazioni rigorose di alimentazione imposte ai dispositivi passivi 	<ul style="list-style-type: none"> - Costosi, dovuto alla batteria e al contenitore - Affidabilità - impossibile determinare se una batteria sia buona o difettosa, specialmente negli ambienti con Tag multipli - La proliferazione larga di transponder attivi presenta un rischio ambientale, di prodotti chimici potenzialmente tossici usati nelle batterie 	<ul style="list-style-type: none"> - Usati principalmente nei sistemi in tempo reale per rintracciare materiali di alto valore.
Attivi			<ul style="list-style-type: none"> - Usati nella logistica per rintracciare container sui treni, camion, etc.

Figura 3 - Confronto tra Tag attivi e passivi

2.2 Vantaggi tecnologie RFID

La tecnologia RFID è considerata per la sua potenzialità di applicazione una tecnologia *general purpose* e presenta un elevato livello di pervasività, ad esempio una volta trovata la sua applicazione in un punto di una filiera produttiva, i benefici si propagano velocemente a monte e a valle della stessa.

2 Tecnologie RFID passive

Detta tecnologia, infatti, ha alcuni vantaggi semplici rispetto alle tradizionali tecnologie dei codici a barre e delle bande magnetiche:

- Non deve essere a contatto per essere letto come le bande magnetiche
- Non deve essere visibile per essere letto come per i codici a barre
- Si possono anche aggiungere informazioni sui chip in funzione della tipologia del chip (*Read Only*: si possono solo leggere le informazioni contenute, *Write Once, Read Many*: si possono scrivere nel chip le informazioni una sola volta, ma leggerle un numero illimitato di volte; *Read and Write*: si possono leggere e memorizzare informazioni per un numero limitato ma grande di volte)
- L'identificazione e la verifica avvengono in 1/10 di secondo
- La comunicazione può essere in chiaro o cifrata
- La lettura è affidabile
- Eliminazione della necessità di "vedere" l'etichetta (le etichette radio possono essere contenute all'interno dei prodotti ed essere lette anche in più esemplari contemporaneamente)
- La capacità di lavorare in ambienti contaminati, e sporchi
- La Capacità di resistere, con opportune protezioni, all'aggressione di agenti chimici e ambientali, di poter operare immerso in un fluido, dentro l'oggetto che si vuole identificare oppure all'interno di un altro contenitore (purché non completamente metallici)
- La possibilità di leggere, nello stesso contenitore, il codice di decine o centinaia di etichette in un lasso temporale di pochi secondi, e di trasmetterlo al sistema informativo di gestione.

Inoltre, i Tag dotati di memorie non volatili (qualche kilobyte) possono contenere informazioni molto articolate sull'oggetto cui sono associate. La modalità *Read/Write* permette non solo una trasmissione di informazioni ma

un loro aggiornamento sul chip. Il Tag diventa un sistema di identificazione che può tenere traccia della storia di un prodotto fin dalla fase di lavorazione ed essere poi utilizzata in modo interattivo lungo tutta la filiera fino alla distribuzione al dettaglio e in alcuni casi sino al consumatore.

Alcuni vantaggi di questa modalità sono costituiti dalla possibilità di memorizzare dati relativi agli indici di qualità, ai problemi riscontrati e successivamente, dalla semplice lettura del Tag, valutare le caratteristiche positive e negative dei prodotti o dei lotti; per esempio applicati alle confezioni di prodotti deperibili alle alte temperature sono in grado di informare il consumatore che il livello di guardia di queste è stato superato (esempio: camion guasto fermo per ore sotto il sole).

Nei sistemi industriali particolarmente complessi e operanti in ambienti ostili, la presenza di un Tag con queste modalità può sostituire sia il *network* sia la necessità di avere sempre attivo il controllo di un sistema di gestione e in questo modo automatizzare alcuni processi amministrativi o industriali, localizzare in magazzino i differenti modelli, smistare in distribuzione modelli e prodotti in funzione di alcune caratteristiche (prezzo, dimensioni, packaging, ecc.). Questi Tag si rivelano utili anche per generazione automatica di bolle e fatture, grazie alla possibilità di leggere contemporaneamente più codici. Anche la fase di vendita trova vantaggi dall'uso dei Tag, sia per realizzare inventari *real time* all'ingresso e alla vendita del prodotto, sia perché i Tag possono essere utilizzati come dispositivi antitaccheggio.

Le tecnologie RFID offrono vantaggi pratici a chiunque abbia bisogno di tenere traccia di un'attività; ai produttori e ai rivenditori, per controllare e incrementare l'efficienza nelle loro catene di approvvigionamento e di migliorare la pianificazione della domanda; infatti, negli ultimi anni molte catene di grande distribuzione e produttori di beni di consumo hanno

cominciato ad inserire Tag in pallet e merci per migliorare la gestione delle spedizioni ai clienti.

Ciò che ha reso possibile la crescita e l'espansione sempre maggiore delle tecnologie RFID sono state le riduzioni di costo e di dimensioni dei componenti per semiconduttori. Alcuni dei primi Tag RFID sono stati grandi come forni a microonde, e oggi si possono avere dei Tag piccolissimi e sempre meno costosi.

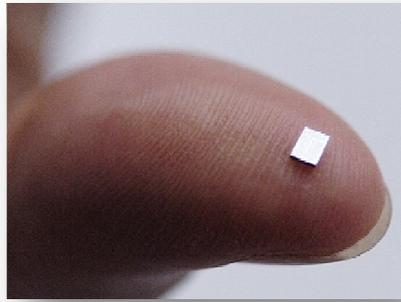


Figura 4 - Tag di dimensioni ridotte

2.3 Applicazioni degli RFID

Diversi sono i campi di applicazione della tecnologia. In particolare i campi di adozione principali esistenti sono:

- Controllo accessi
- Monetica
- Bigliettazione Elettronica
- Passaporti
- Logistica Magazzini
- Logistica Trasporti
- Tracciamento pratiche
- Assistenza e manutenzione

- Identificazione animali
- Antitaccheggio
- Registro Scolastico Elettronico
- Rilevazione dei parametri ambientali
- Monitoraggio raccolta rifiuti

2.3.1 Controllo presenze ed accessi

L'RFID è una valida alternativa sia alle tecnologie di *personal identification* tradizionali (*badge*, tesserini, ecc.), sia alle tecnologie di *strong authentication* basate sul riconoscimento degli attributi biometrici di un individuo. A differenza di tali tecnologie non richiede contatto visivo per l'identificazione e permette il riconoscimento anche "a distanza". L'identificazione tramite RFID oltre a rendere più agile l'impiego di varchi motorizzati, distinguere gli ingressi dalle uscite e verificare automaticamente l'elenco delle presenze all'interno di una determinata zona, permette l'avvio o l'arresto di un PC a seconda che il proprietario si trovi o meno nelle vicinanze. I Tag possono essere stampati o inseriti in oggetti di forma diversa, come ad esempio un *badge* identificativo e, quindi, personalizzati con stampe di immagini, scritte, loghi, fotografie e codici a barre. Possono essere registrate informazioni come: dati anagrafici, foto di riconoscimento, data e ora di transito, verso di transito e altre informazioni. In alcuni casi è ipotizzabile che si permetta il funzionamento di una macchina o apparecchiatura solo in presenza di operatori di macchina (es. una pressa o una TAC o semplici ascensori). Tali apparecchiature possono richiedere particolari procedure per assicurare la sicurezza delle persone o della macchina stessa impedendo a non autorizzati il loro utilizzo o arrestare il suo funzionamento se nell'area non sia presente un operatore autorizzato.

2.3.2 Monetica

Per quanto riguarda la monetica, Visa, MasterCard e American Express stanno lanciando nuove carte di credito che per sicurezza, velocità e flessibilità superano le tradizionali Chip Card. In Italia ci sono attualmente diversi progetti pilota tra cui: Postepay Poste Mobile, che implementa il PayPass di MasterCard, TellCard, una prepagata del Credito Valtellinese con il PayWave di Visa, e Sanpaolo di Torino. In questo ambito sono attese le principali applicazioni futuribili di *Near Field Communication* (NFC) che prevedono l'associazione di un telefono *NFC-enabled* che permetterà di effettuare transazioni di monetica o accedere servizi.



Figura 5 – Transazioni monetarie e accesso a diversi servizi

2.3.3 Passaporto

I Tag RFID sono utilizzati in passaporti in diversi paesi. Il primo paese ad adottare i passaporti con RFID ("*e-passports*") è stato il governo malese nel 1998. In tale passaporto, il chip contiene oltre alle informazioni dell'intestatario del passaporto e la storia dell'utilizzo (data, giorno e luoghi) per tutte le entrate e uscite dal paese. Anche il Governo italiano ha adottato tale standard, i nuovi passaporti sono disponibili dal 26 ottobre 2006.

I Tag RFID sono utilizzati anche nei nuovi passaporti Britannici e Americani dal 2006. Gli Stati Uniti hanno emesso 10 milioni di passaporti nel 2005 e 13 milioni nel 2006. I chip memorizzeranno le stesse informazioni stampate nel passaporto e includeranno anche una fotografia digitale del proprietario. I passaporti conterranno anche una sottile membrana metallica al fine di rendere difficile le letture non autorizzate (*skimming*) quando il passaporto è chiuso.

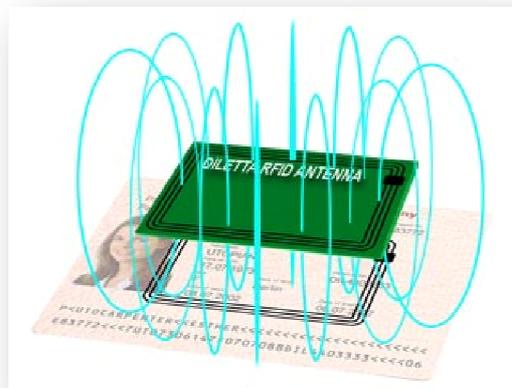


Figura 6 – Passaporto con RFID (*e-passports*)

2.3.4 Bigliettazione elettronica

Uno degli ambiti di maggiore applicazione planetaria delle soluzioni RFID è data dai sistemi di bigliettazione elettronica. Le città di Parigi, Londra, Roma, Torino, Milano, Venezia utilizzano *contactless smartcard* con tecnologia RFID per permettere l'accesso ai mezzi di superficie e metropolitana. Anche i sistemi di accesso degli impianti di risalita sulle piste da sci, sono ormai quasi tutti basati su soluzioni RFID. Grazie a questa tecnologia, i siti web di alcune società di gestione degli impianti consentono

all'utilizzatore, a fine giornata, di ottenere un riepilogo degli impianti utilizzati e dei chilometri di pista percorsi.



Figura 7 - Contactless smartcard

2.3.5 Logistica magazzini

Identificare ogni contenitore e ogni scaffale di magazzino con Tag riduce gli errori nei prelievi e fornisce una identificazione certa. Non è necessario aprire gli imballaggi per verificare il contenuto cercando il codice a barre, così come non è più necessario effettuare il conteggio manuale per la verifica dell'inventario fisico. Con una serie di scansioni a distanza è possibile identificare e verificare la presenza di specifici oggetti in magazzino. Infatti la tecnologia permette di leggere contemporaneamente più etichette (Tag). La tecnologia permette di conoscere in tempo reale le giacenze di magazzino e riordinare i capi esauriti (in tempo reale).

2.3.6 Logistica trasporti

In questo caso i Tag sono applicati sia sugli oggetti (scatole, pallet, ecc.) da trasportare, sia sui mezzi di trasporto (vagoni, automobili, ecc.). In Italia, Francia e in Giappone sono già funzionanti milioni di tessere RFID che permettono ai pendolari di utilizzare diversi tipi di trasporto con le diverse forme di abbonamento. Un'altra applicazione della tecnologia RFID è in

sostituzione del codice a barre come identificativo sui bagagli in aeroporto permettendo un maggiore "tasso di lettura" ed errore lungo gli scivoli di smistamento.

I sistemi RFID contribuiscono a migliorare la qualità dei sistemi di identificazione del mezzo di trasporto (l'esempio più comune è il telepass) in termini di efficienza e servizio.

2.3.7 Tracciamento pratiche

L'applicazione di una etichetta RFID a ogni pratica cartacea consente di automatizzare la loro ricerca negli archivi, di effettuare automaticamente la registrazione del prelievo/restituzione e di mantenere traccia dei vari spostamenti tra uffici e depositi. Applicare un Tag RFID su una pratica significa poterne gestire gli spostamenti, e quindi poterne gestire il *workflow*. Un sistema del genere mira all'ottimizzazione della gestione delle pratiche negli uffici, poiché è possibile stabilire quanto tempo una pratica rimane ferma su di una.

Stessa soluzione si avrebbe applicando i Tag sui beni delle biblioteche (libri, video, cd, audio, ecc.) in modo da rilevare a distanza le informazioni in esso contenute (tipo di bene, descrizione, numero inventario, rappresentazioni fotografica, ecc.), consentendo di amministrare i beni in dotazione con estrema facilità ed efficacia.



Figura 8 –Tag inseriti nelle copertine dei libri

2.3.8 Assistenza e manutenzione

Interessante è l'applicazione di sistemi RFID nella manutenzione degli impianti. Un esempio per tutti, è quello delle aziende petrolchimiche dove devono effettuare manutenzione sulle valvole. Con una semplice lettura del Tag applicato direttamente sulle valvole sarà possibile ottenere la storia delle manutenzioni e riparazioni della specifica valvola.

2.3.9 Identificazione degli animali

Rispetto agli altri metodi utilizzati per l'identificazione degli animali (marchiatura sull'orecchio, tatuaggio, passaporto cartaceo), con l'applicazione dei Tag tutte le informazioni necessarie sono residenti anche sui capi di bestiame e, grazie all'emissione di onde elettromagnetiche a bassa frequenza del tutto innocue, risultano accessibili ovunque si trovi l'animale. Le etichette possono contenere le informazioni indispensabili a garantire la qualità del capo come ad esempio:

- Codice dell'animale;
- Dati anagrafici (passaporto) proprietario;
- Aziende presso le quali il capo è transitato;

- Controlli veterinari a cui l'animale è stato sottoposto;
- Trattamenti subiti.



Figura 9 – Esempio di Tag applicato agli animali

2.3.10 Antitaccheggio

Il primo aiuto tecnologico, oggi largamente utilizzato su scala mondiale, è arrivato dalla EAS (*Electronic Article Surveillance*). Mediante l'applicazione di un piccolo Tag (senza chip) agli oggetti in vendita, un negozio può rilevare un eventuale transito non autorizzato di un articolo attraverso un varco. Il varco (composto da antenna) è collegato ad un dispositivo di segnalazione acustica e visiva. L'EAS tuttavia non permette di rilevare né la tipologia né il numero degli oggetti rilevati.



Figura 10 – Varco e Tag antitaccheggio



Figura 11 – Tag antitaccheggio inserito sotto l’etichetta del codice a barre

2.3.11 Registro scolastico elettronico

Attraverso la tecnologia RFID è possibile creare applicazioni per il controllo delle presenze nelle scuole in modo automatico ed efficiente. L'Istituto Tecnico Industriale Statale Vittorio Emanuele Marzotto ha iniziato nel mese di Settembre 2006 un progetto scolastico con l'aiuto di alcune aziende locali per la creazione di un sistema in grado di rendere i processi scolastici di registrazione delle presenze, dei voti, delle comunicazioni molto veloci, sicuri ed automatici. Con la tecnologia RFID vengono gestiti gli accessi al sistema e le presenze degli alunni vengono automaticamente registrate dalle applicazioni all'ingresso a scuola. Le applicazioni sono state sviluppate dagli studenti delle due classi di Informatica dell'anno 2006/2007 con l'aiuto dei loro professori.

2.3.12 Monitoraggio raccolta rifiuti

In Inghilterra, in Australia, e da tempo anche da aziende di raccolta italiane i chip RFID vengono installati nei cassonetti, consentendo la registrazione di dati come il peso di ogni contenitore, e l'identificazione delle zone di appartenenza, così da creare mappe per la valutazione del territorio.

In Svizzera nel Canton Ticino l'Azienda Cantonale dei Rifiuti ha adottato un sistema di pesa automatica e di identificazione della provenienza dei rifiuti al fine di meglio ripartire le spese di discarica locali.

2.3.13 Rilevazione dei parametri ambientali

L'ultima frontiera tecnologica in ambito RFID riguarda l'introduzione di Tag attivi equipaggiati con sensori in grado di rilevare i parametri climatici (temperatura, pressione, umidità, ecc.) dell'ambiente in cui sono immersi. Le grandezze rilevate dai sensori vengono memorizzate in una apposita memoria interna, e lì permangono fino a quando un operatore, dotato di apposito lettore, non ne esegue lo scarico su un PC palmare. Queste caratteristiche si rivelano strategiche per il monitoraggio dei parametri operativi dei macchinari in particolari realtà industriali, dove è necessario garantire regimi operativi controllati.

I Tag, grazie alle ridottissime dimensioni, possono essere collocati in punti "scomodi", dove sarebbe difficile portare il cavo necessario ad alimentare un apparecchio di misura, ed offrono, a costi decisamente contenuti, una soluzione affidabile e di facile implementazione.

2.4 Problemi di privacy e sicurezza legati all'utilizzo della tecnologia RFID

Tutte le informazioni contenute a bordo del chip (a volte anche personali a seconda dell'utilizzo del Tag), potrebbero essere oggetto di letture "non autorizzate" da parte di terze parti per finalità diverse da quelle originarie. La lettura di un "Tag" su un capo a fine inventariale è identica (trattandosi di soluzioni a breve raggio) di un codice a barre, che non contiene informazioni sul possessore. Diverso trattamento le informazioni contenute

sul Tag RFID del passaporto e in alcune regioni sulla tessera del servizio sanitario. In particolare tali provvedimenti sono di attualità alla luce che le architetture RFID nel campo dei servizi (es. passaporto elettronico, chip biomedico) sono una realtà, e ciò comporta nuovi problemi e rischi legati alla Privacy dei cittadini in generale (rischio di geolocalizzazione in tempo reale e remota, accesso a dati personali, indagini di mercato ecc.).

L'annuncio dell'utilizzo della tecnologia RFID ha generato opinioni diverse tra i consumatori. Di diversa portata è la questione del chip sottocutanei. Tali chip servono per memorizzare talune informazioni di un paziente e permettere, in caso di ricovero in emergenza, di attingere alla storia del paziente. Sono informazioni personali e il chip viene impiantato su richiesta dell'interessato. Il problema è di inferiore urgenza nel settore della logistica, il quale prevede che i Tag contengano solamente un unico codice: un numero di serie.

Le preoccupazioni principali riguardano la possibilità che la tecnologia possa essere utilizzata per violare la privacy del possessore degli oggetti taggati.

Al momento le principali obiezioni sono articolate sui seguenti elementi:

- la consapevolezza o meno dell'utente che un prodotto contenga un Tag RFID
- le informazioni contenute nel Tag e l'associazione possessore/Tag.

Il primo punto è al momento allo studio in molte legislazioni, ma emerge un quadro univoco sulla necessità che i clienti siano informati dell'esistenza del Tag, che i Tag possano essere disattivati (tramite una funzionalità specifica dei Tag stessi) con la possibilità per gli utenti di verificarne la disattivazione, e che i Tag possano essere rimossi.

Il secondo punto riguarda le informazioni contenute nei Tag. Nel caso di prodotti di largo consumo le informazioni nei Tag sono codificate, ovvero il Tag sui prodotti contiene solo un indicativo seriale. La presenza di un indicativo seriale può comunque porre problemi di privacy, come notato in altre tecnologie già presenti. È possibile che un malintenzionato possa leggere a distanza i codici dei Tag dei prodotti acquistati da una persona, e possa quindi ricostruire le abitudini di consumo di ciascuna famiglia. Va notato tuttavia che ciò può avvenire anche esaminando i carrelli degli acquisti, accedendo al database delle tessere di fedeltà, esaminando la spazzatura (come fanno le aziende di analisi dei dati di mercato) ecc..

Il terzo punto riguarda la possibilità di associare una persona ad un prodotto. Così come avviene per i telefoni cellulari, un Tag non identifica direttamente una persona; è necessario disporre di un accesso al database che associa gli identificativi dei Tag alle persone, come ad esempio il database delle tessere fedeltà dei supermercati o degli abbonati dell'operatore telefonico.



3 Tecnologia RFID attiva

3.1 Panoramica

Quando si pianifica l'architettura di un sistema RFID, i passi chiave da compiere comprendono: la selezione di un sistema di numerazione; la codifica delle etichette RFID, allegando i Tag, agli elementi che si desidera monitorare, e leggendo gli articoli dotati di Tag mentre si muovono attraverso l'ambiente fisico; l'integrazione delle informazioni RFID in applicazioni aziendali e processi di business, ed infine, la condivisione delle informazioni RFID all'interno e tra aziende.

I sistemi RFID presentano più livelli, ed i componenti di livello fisico includono Tag RFID, antenne, lettori e sensori; è quindi fondamentale conoscere gli aspetti hardware dell'architettura del sistema RFID. Come per ogni applicazione *enterprise*, è fondamentale capire quali sono i requisiti principali per aumentare la qualità del sistema, prima di progettarlo. Importanti qualità del sistema RFID includono la privacy e la sicurezza, le prestazioni, la scalabilità, la gestibilità e la flessibilità.

3.2 Il Tag

Lo scopo di un Tag RFID è quello di collegare fisicamente i dati su un oggetto. Ogni Tag ha un meccanismo interno per memorizzare dati e un modo di comunicare tali dati. Un Tag RFID è una sorta di microchip con una fonte di energia, avente un'etichetta o una bobina che funge da antenna di qualche tipo. La classificazione dei Tag è effettuata in base:

- caratteristiche fisiche;
- interfacce di rete (come comunicano con i lettori);
- memorizzazione delle informazioni;
- capacità di trasformazione.

Con questi criteri potremo osservare alcune delle funzionalità di base dei Tag.



Figura 12 – Esempi di Tag attivi

3.2.1 Funzionalità di base

Le funzionalità di base dei Tag sono:

- inserimento del Tag; ogni Tag RFID deve essere collegabile a un elemento in qualche modo;
- la lettura del Tag; ogni Tag RFID deve essere in grado di comunicare informazioni su delle frequenze radio in qualche modo.

Molti Tag, inoltre, offrono anche una o più delle seguenti caratteristiche e capacità:

Disable

Alcuni Tag consentono ad un lettore di ordinarli di smettere di funzionare in modo permanente (*Kill*); il Tag, ricevuto detto codice, non potrà nuovamente rispondere a un lettore.

Write once

Molti Tag sono realizzati con dati permanenti impostati già in fabbrica, ma un Tag scrivibile una sola volta può essere impostato su un valore particolare da un utente finale, dopo di che, il Tag non può essere cambiato, tranne che per essere disabilitato.

Rewriting

Alcuni Tag possono essere riscritti con nuovi dati più e più volte.

Anti-collisione

Quando molti Tag sono nelle immediate vicinanze, un lettore potrebbe avere difficoltà a dire se la risposta di un Tag finisce e un'altra comincia. I Tag anti-collisione devono saper aspettare il proprio turno quando rispondono a un lettore.

Sicurezza e crittografia

Alcune etichette sono in grado di partecipare a comunicazioni criptate, e alcuni Tag di rispondere unicamente ai lettori che possono fornire una password segreta.

3.3 Caratteristiche fisiche

I Tag RFID hanno un ampio assortimento di forme e dimensioni e possono essere alloggiati in vari tipi di materiale determinandone le caratteristiche fisiche.

I Tag in PVC o plastica, sono durevoli e riutilizzabili, per esempio le etichette RFID a forma di carte di credito. I Tag inseriti negli strati di carta, chiamati "etichette intelligenti" possono essere applicati con applicatori automatici simili a quelli utilizzati per le etichette dei codice a barre. I Tag più piccoli possono essere incorporati in oggetti di uso comune come l'abbigliamento, orologi e braccialetti, chiavi e portachiavi; infine i Tag in capsule di vetro, possono sopravvivere anche in ambienti corrosivi o in liquidi.

Quindi, il *packaging* è una proprietà, molto importante, che ci dice come il Tag può essere legato ad un elemento.



Figura 13 - Tag inseriti in materiali differenti

3.3.1 Fonte di energia

L'alimentazione, è uno dei fattori principali che determina il costo e la longevità di un Tag. I Tag passivi ottengono tutta la loro energia da alcune modalità di trasmissione da parte del lettore, mentre i Tag attivi utilizzano una batteria di bordo per le comunicazioni e per tutte le proprie funzioni vitali. I Tag che utilizzano la batteria per le proprie funzioni, sono stati denominati "attivi" o semi-passivi.

Come ci si potrebbe aspettare, avendo una batteria a bordo per un chip, il Tag diventa più costoso, ma i Tag semi-attivi e passivi hanno diversi vantaggi rispetto ai Tag passivi. Nel caso del Tag semi-passivi, la portata di lettura può essere più lunga ed ampia perché la comunicazione passiva può utilizzare tutta la potenza fornita dal lettore per le comunicazioni invece di condividere parte dell'energia con il chip.

3.3.2 Air Interface

L'Air interface descrive il modo in cui avviene la comunicazione di un Tag con un lettore. Conoscendo *l'Air interface* di un Tag, possiamo determinare la portata di lettura dello stesso ed identificare i lettori compatibili con i Tag. Gli attributi principali che caratterizzano *l'Air Interface* includono la fonte di alimentazione del Tag, la frequenza di funzionamento, la modalità di comunicazione e la codifica di accoppiamento.

3.3.3 Codifica

Come il codice "Morse" che usa toni lunghi per rappresentare i trattini e toni brevi per rappresentare i puntini, se si sostituissero "0" per "tratto" e "1" per "punto" di codifica, il codice "Morse" avrebbe lavorato per l'invio di informazioni attraverso un bus seriale. Anche il Tag e il lettore, devono avere una codifica per poter interpretare i cambiamenti nel supporto

3 Tecnologia RFID attiva

analogico e poter rappresentare i dati digitali. Potrebbe sembrare strano pensare gli RFID, come dispositivi di comunicazione su un bus seriale, ma anche se i componenti sono senza fili anziché via cavo, la maggior parte dei Tag e dei lettori utilizzano la comunicazione seriale a causa delle limitate gamme di frequenze disponibili.

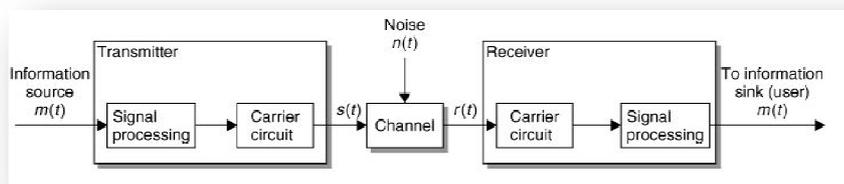


Figura 14 - Sistema di comunicazione digitale

Lo schema a blocchi nella figura descrive un sistema di comunicazione digitale che è composto da tre blocchi: trasmettitore, canale di trasmissione, ricevitore. Analogamente i sistemi RFID, per il trasferimento dati utilizzano il seguente processo:

- nel blocco trasmettitore, l'informazione da trasmettere viene codificata e modulata (traslazione in frequenza del segnale);
- il segnale modulato attraversa il canale di trasmissione, il quale aggiunge inevitabilmente un segnale di disturbo (rumore).
- Il segnale viene ricevuto dal ricevitore, che lo demodula riportandolo in banda base così da essere decodificato;

La codifica dei dati trasmessi è rappresentata da livelli logici binari e cioè da serie di 0 e di 1.

Nei sistemi RFID questa serie binaria corrisponde normalmente ad una delle codifiche seguenti: NRZ, Manchester, RZ Unipolare, DBP, Miller, Miller Modificato, Differenziale, come mostrato nella figura sottostante.

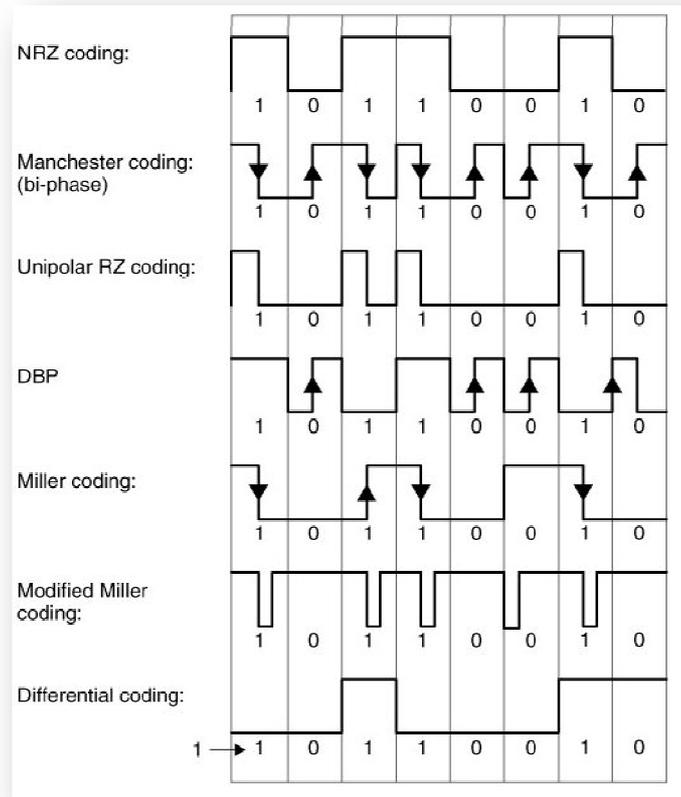


Figura 15 - Varie codifiche binarie

3.3.4 Anti-collisione

Questo termine descrive l'insieme delle procedure che impediscono ai Tag di interrompere un altro Tag e di comunicare a sproposito. L'anti-collisione dei Tag deve regolare la tempistica delle risposte e trovare il modo di

randomizzazione quelle risposte in modo che un lettore sia in grado di capire ogni Tag in mezzo a tutte le risposte ricevute.

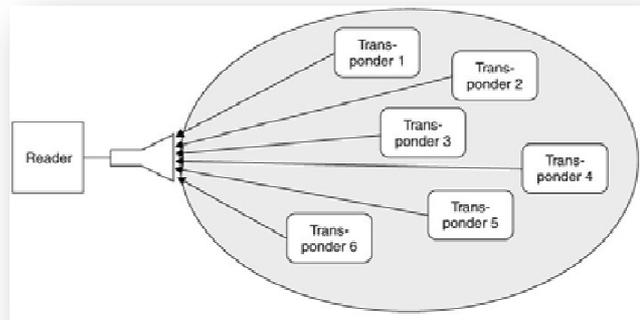


Figura 16 - Procedura di anticollisione

Il processo di comunicazione comincia quando il lettore inizia ad emettere un segnale ad una certa frequenza; i Tag che si trovano nella zona d'interrogazione si attivano. Una volta che il Tag ha decodificato il segnale ricevuto, risponde al lettore. Se molti Tag sono presenti all'interno della suddetta zona, tutti risponderanno simultaneamente; se questo accade il lettore rileva una collisione del segnale e questo indica la presenza multipla di Tag. In questo caso il lettore per comunicare con i singoli Tag usa una procedura di anticollisione, ossia organizza la comunicazione tramite una lista ordinata di modo tale che possano essere selezionati singolarmente. Esistono molte procedure di anticollisione (albero binario, etc.) che fanno parte delle specifiche del protocollo di comunicazione.

Il numero di Tag che possono essere identificati dipende dalla frequenza e dal protocollo impiegati nella comunicazione, tipicamente varia da 50 a 200 Tag. Una volta che il Tag è stato selezionato, il lettore può realizzare tutte le operazioni permesse (lettura ID, scrittura in memoria, etc.). Dopo che sono state fatte tutte le operazioni opportune sul Tag selezionato, il lettore

procede mettendo in stand-by questo, e attivando il successivo della lista, e così via fino all'ultimo della lista .

3.3.5 Identità

L'identità è un nome, il numero o l'indirizzo che si riferisce unicamente ad una cosa o luogo e caratterizza e distingue ogni dispositivo. Poiché il lettore comunica con i Tag utilizzando radio frequenze, qualsiasi lettore RFID deve essere dotato di una o più antenne. Inoltre, perché un lettore possa comunicare con un altro dispositivo, deve avere anche una interfaccia di rete di qualche tipo. Esempi di interfacce di rete comune, possono essere i *Serial Universal Asynchronous Receiver*, trasmettitori (UART) per RS 232 o RS 485 , il *jack* RJ45 per 10BaseT oppure i cavi *Ethernet* per 100BaseT, oltre ad alcuni lettori che hanno anche le comunicazioni *Ethernet*, *Wireless* o *Bluetooth*.

3.3.6 Antenna

Anche se le antenne sono concettualmente semplici, gli ingegneri lavorano costantemente per ottenere una migliore ricezione a bassa potenza e ad adeguare le antenne a circostanze particolari. Alcuni lettori hanno solo una o due antenne, altri possono essere in grado di gestire molte antenne in modalità remota. Il limite principale del numero di antenne che un lettore è in grado di controllare è dovuto alla perdita di segnale sul cavo che collega il trasmettitore e il ricevitore. Alcuni lettori utilizzano un'antenna per la trasmissione e una per la ricezione. In questo tipo di configurazione, la direzione di movimento del Tag attraverso il campo d'azione del lettore è particolarmente importante. Se l'antenna di trasmissione è "avanti" all'antenna ricevente, l'antenna ricevente avrà una quantità di tempo più lungo per ricevere i segnali dal Tag. Il Tag spenderà molto meno energia, se

sarà nel raggio di copertura dell'antenna di ricezione e disporrà di un periodo più lungo per ricevere la risposta di quello che servirebbe se le due antenne fossero state invertite, il che significa che vi è una migliore possibilità che i Tag verranno letti.

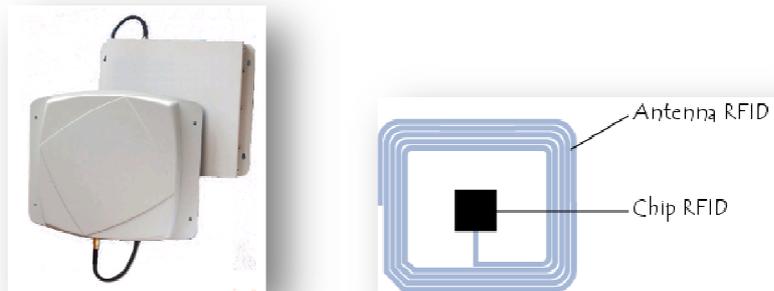


Figura 17 - Esempi di antenne di diverse dimensioni

3.3.7 Controller

Il dispositivo di elaborazione che controlla un lettore può variare in complessità, da una macchina a stati semplice su un chip, che potrebbe essere utilizzato per un piccolo lettore incorporato in un telefono o un PDA, ad un sistema microcomputer completo in grado di eseguire un sistema operativo, con applicazioni utente e accumulando una grande quantità di dati su un disco rigido interno. Il controller è responsabile per il controllo dal lato del lettore del protocollo di Tag, così come la determinazione nell'invviare alla rete le informazioni lette.

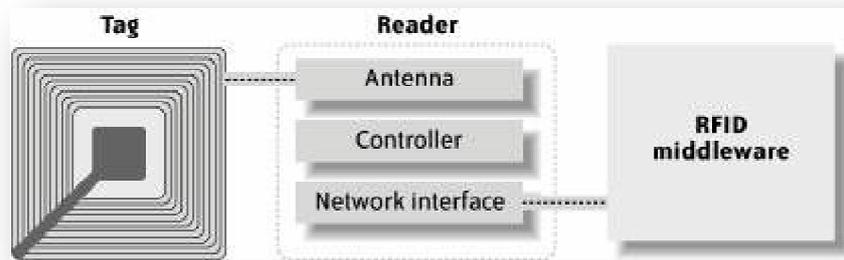


Figura 18 - Componenti fisici di un lettore

3.3.8 Interfaccia di rete

I lettori comunicano con la rete e gli altri dispositivi attraverso una varietà di interfacce. Storicamente, la maggior parte dei lettori RFID, ha utilizzato interfacce seriali RS 232 o RS 422 (punto a punto) o RS 485 (indirizzabile). Negli ultimi anni, i lettori hanno utilizzato collegamenti *Ethernet*, e negli ultimi periodi, alcuni hanno cominciato ad utilizzare reti *Wireless* e *Bluetooth*.

3.3.9 Forme e dimensioni

I lettori hanno una gamma che può variare da un formato da mezzo pollice (due centimetri) alla dimensione di un computer desktop. I lettori possono essere integrati in dispositivi portatili o anche telefoni cellulari. Essi possono essere fissati alla parete, in custodie di varia natura e possono anche essere integrati in scaffalature e porte con antenne intelligenti per la piattaforma e le applicazioni del portale.

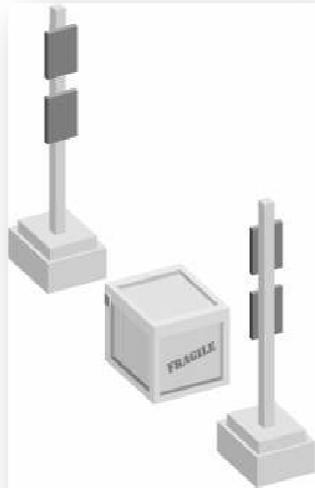


Figura 19 – Esempio di un portale con antenne

3.4 RFID nella storia

Gli RFID sono stati utilizzati per la prima volta durante la seconda guerra mondiale, quando gli inglesi svilupparono un sistema automatizzato per i loro aerei i quali comparivano sul radar e pulsavano secondo un ritmo specificato. Questo nuovo sistema è stato il predecessore di tutti i sistemi transponder sugli aeromobili ed è ancora oggi in uso (*Identification Friend or Foe* (IFF)).

Come i sistemi RFID di oggi, i componenti più importanti del sistema sono stati un lettore e un transponder. Nei sistemi IFF, il lettore era il sistema radar stesso e il transponder era una scatola ingombrante di tubi con manopole e interruttori situata negli aerei. Il sistema funzionava così: la stazione a terra inviava un segnale radar, e il transponder che riceveva questo segnale lo rimandava indietro; il transponder rispondeva con un segnale in avanti e indietro su una piccola gamma di frequenze, causando il ritorno radar pulsante secondo un ritmo specifico, e grazie all'antenna radar era possibile ricevere un ritorno più forte di quanto sarebbe stato altrimenti.

3.5 Conclusione

In conclusione, i Tag possono avere caratteristiche differenti a seconda dell'utilizzo che ne viene fatto, con caratteristiche fisiche durevoli per ambienti ostici, o etichette di carta per ambienti controllati dove si necessita una rapida e automatizzata applicazione degli stessi. Un'interfaccia aerea, stabilisce quali Tag possono comunicare con un determinato lettore, e a seconda del tipo di connessione definito per l'interfaccia aerea, possiamo determinare la distanza massima di azione tra un lettore e un Tag. I Tag disponibili hanno una varietà di capacità di memoria per lo stoccaggio dei dati ed alcuni hanno anche ulteriori capacità di elaborazione delle informazioni ma queste funzionalità aggiuntive hanno un costo maggiore, ripercuotendosi sul costo finale dell'apparecchio stesso.

Quindi, la scelta dei materiali e del tipo di Tag da utilizzare, va fatta in base a molti fattori, che vanno considerati sin dall'inizio, a seconda della rete e del tipo di progetto che si va a sviluppare, ad esempio, utilizzando etichette intelligenti per applicazioni automatizzate in un magazzino, piuttosto che l'uso di vetro o pvc utilizzato per ambienti più ostici; oppure scegliere Tag passivi per il costo più basso, o semi-passivi o etichette attive quando necessario per le loro funzionalità aggiuntive, ma con conseguente aumento dei costi.



4. La Tecnologia SNAP

4.1 SNAPpy

“Synapse” è una piattaforma estremamente potente e flessibile per lo sviluppo e la distribuzione di applicazioni *Embedded Wireless*. SNAP è l'acronimo di *Synapse Network Appliance Protocol*, ed è il protocollo utilizzato da tutti i nodi “Synapse” *wireless*. La combinazione di SNAP e del linguaggio Python è nota come "Snappy".

Il termine SNAP si è evoluto nel tempo per indicare genericamente la linea di prodotti. Ad esempio, si parla spesso di "SNAP Networks", "SNAP Node", e "Application SNAP". Ogni nodo viene implementato in Python, un linguaggio di script pseudo compilato. Questo significa che, similmente al Perl, ogni programma sorgente deve essere eseguito da un interprete. L'interprete è un normale programma che va installato sulla propria macchina, e si occuperà di interpretare il codice sorgente e di eseguirlo. Quindi, diversamente dal C#, non abbiamo un fase di compilazione (*linking*) che trasforma il sorgente in eseguibile, ma avremo a disposizione

solo il sorgente che viene direttamente eseguito dall'interprete. Il Python è un linguaggio orientato agli oggetti, supporta le classi, l'ereditarietà e si caratterizza per il “*binding* dinamico”. Ragionando in termini di C# potremo dire che tutte le funzioni sono virtuali. Il software di base SNAP funziona quindi su ogni nodo. Questo codice gestisce le comunicazioni senza fili oltre che l'interfacce di dispositivi esterni.

4.2 Portal e Snap connect

Synapse Portal è un software che gira su un PC standard, utilizzando una interfaccia USB, si connette a qualsiasi nodo del “*Wireless Network SNAP*”, diventando una interfaccia grafica tra l'utente e l'intera rete. Utilizzando il “Portal”, è possibile creare, implementare, configurare e monitorare le applicazioni di rete molto rapidamente e facilmente. Una volta collegato, il “PC portale” ha un proprio ed unico indirizzo di rete, e può partecipare alla rete SNAP come un nodo, facendo da veicolo per le informazioni. Ciò consente di sviluppare, configurare e distribuire applicazioni SNAP direttamente sui nodi della rete.

4.3 Panoramica

SNAP, la *Synapse Network Appliance Protocol*, è una famiglia di tecnologie software che formano una soluzione integrata per il monitoraggio e il controllo senza fili.

Le caratteristiche principali di SNAP sono :

- Tutti i dispositivi sono paritari; tutti possono essere un ponte per il portale, un nodo di *routing*, ecc.. Non ci sono "coordinatori" in SNAP.

- SNAP implementa una topologia di rete completa. Ogni nodo può parlare direttamente con qualsiasi altro nodo entro la portata radio, e può parlare a qualsiasi nodo all'interno della rete SNAP.
- La comunicazione tra i dispositivi può essere *unicast* o *multicast*.
- *Remote Procedure Call* (RPC) è il metodo fondamentale di messaggistica tra dispositivi.
- L'interfaccia con l'utente avviene grazie al Portal (un nodo funge da portale) e lo stesso Portal, appare come un nodo normale sulla rete SNAP.

4.3.1 Remote Procedure Call

Tutti i dispositivi SNAP possiedono un insieme di funzioni *built-in* (procedure) per gestire la configurazione di rete di base, servizi di sistema e hardware di controllo del dispositivo. Queste funzioni possono essere invocate in remoto dal portale o da qualsiasi altro dispositivo in rete SNAP. Ulteriori funzioni definite dall'utente possono essere caricate sui dispositivi. Questo processo di caricamento può essere eseguito direttamente su porte seriali, o via etere. Una volta caricate, queste funzioni sono richiamabili anche a livello locale o in remoto.

4.3.2 Snappy Scripting

Il sistema di *scripting*, chiamato Snappy, è un sottoinsieme del linguaggio di programmazione Python, ottimizzato per dispositivi *Embedded* a bassa potenza. Uno "*Snappy script*", è un insieme di funzioni e di dati che vengono trattati dal Portal e caricati su dispositivi SNAP. Il Portal agisce come un "nodo" della rete SNAP, e può inviare e ricevere chiamate RPC come qualsiasi altro nodo. Come per tutti gli altri nodi, Portal è l'immagine

di un dispositivo (*script*) che definisce le funzioni richiamabili da parte di messaggi RPC.

4.4 Rete SNAP

Progressivamente, si sta vedendo una crescita esponenziale nell'uso di reti *Wireless* per il monitoraggio, il controllo ed il consumo sia in ambito commerciale, che industriale, e nella Pubblica Amministrazione. Il loro utilizzo va dalla gestione di edifici (illuminazione, riscaldamento, A/C ...), automazione per il controllo industriale e di comunicazione della macchina, monitoraggio per le applicazioni di sicurezza ecc.. Purtroppo, le reti *Wireless* tradizionali richiedono particolari conoscenze di programmazione *Embedded* (in genere linguaggio C e linguaggio assembly) per sviluppare e distribuire applicazioni che vengono eseguite sui nodi che fanno parte della rete. Inoltre, le convenzionali reti wireless richiedono notevoli quantità di tempo e di risorse per configurarle e gestirle.

In confronto, una rete *Wireless* basata sul software SNAP è *instant-on*, auto-formante ed auto-riparante. Il processo per la creazione di applicazioni per l'esecuzione sui nodi che formano una rete basata su tecnologia SNAP è veloce, efficiente, e non richiede conoscenze di programmazione *Embedded*. Inoltre, gli utenti possono amministrare e gestire una rete basata su SNAP senza dover sapere nulla di reti *Wireless*. Questo approccio rende la tecnologia *Wireless* accessibile a una gamma molto più ampia di utenti e sviluppatori.

4.5 Lo Stack SNAP

Una visione del software SNAP *stack* di alto livello è illustrato in figura 20. Questo *stack* ad alte prestazioni è progettato per funzionare in modo efficiente su microprocessori a 8-bit.

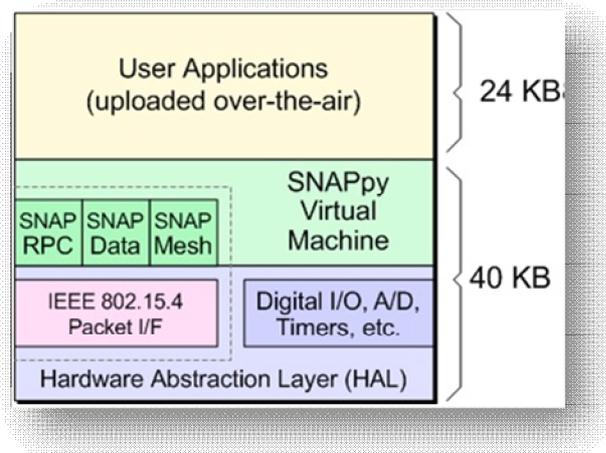


Figura 20 – Lo stack SNAP

SNAP, ha un ingombro di memoria molto piccola di soli 40 KB, lasciando così più spazio per le applicazioni utente.

Lo SNAP *stack* comprende una macchina virtuale in Python (la combinazione di SNAP e Python è denominata come Snappy); Inoltre la stessa applicazione può essere eseguita su qualsiasi processore senza la necessità di ricompilazione.

4.6 Portabilità

Il software SNAP *stack* è stato progettato da zero ed è stato ottimizzato pensando alla portabilità. Due aspetti chiave della portabilità sono le Snappy Virtual Machine e l'*Hardware Abstraction Layer* (HAL) illustrato in figura 20. La HAL fornisce un livello di astrazione tra il corpo principale del software *stack* e il mondo fisico, tra le specifiche di accesso dei registri del processore, la lettura/scrittura dei valori di *Input/Output* (I/O), il controllo analogico-digitale, e così via. Ciò significa che il processo di portabilità dello *stack* in un nuovo processore richiede modifiche solo per il "sottile strato" della HAL.

4.7 La macchina virtuale Snappy

Le applicazioni che vengono eseguite su nodi SNAP sono create in Python, un linguaggio di *scripting* di alto livello, utilizzando l'applicazione Portal. Le applicazioni Snappy vengono automaticamente tradotte in "*byte code*" e sono quindi scaricate nei nodi della rete direttamente tramite il wireless. Come illustrato nella figura 20, il software *stack* SNAP include la *Snappy Virtual Machine*, che esegue il "*byte code*" che forma le applicazioni Snappy. Questo garantisce una portabilità estrema, perché la *Snappy Virtual Machine* fornisce un livello alto di astrazione che separa le applicazioni dalla parte fisica (l'*hardware*). Ciò significa che un'applicazione Snappy è eseguibile immediatamente su qualsiasi processore senza richiedere alcuna modifica o ricompilazione.

4.8 Snap e reti wireless mesh

Il protocollo SNAP può teoricamente supportare fino a 16 milioni di nodi in una singola rete. Dato che queste sono reti *mesh*, non vi è un singolo "*point of failure*": ogni nodo può parlare direttamente con qualsiasi altro nodo che è nelle vicinanze, e ogni nodo può parlare indirettamente, a qualunque altro

nodo attraverso nodi intermedi. Un punto chiave è che i percorsi tra i nodi non devono essere pre-configurati dall'utente. Le reti SNAP sono auto-configuranti, quando un nuovo nodo viene alimentato, è automaticamente integrato nella rete. I nodi SNAP, sono anche *instant-on*, appena un nodo viene acceso, diventa pienamente operativo in una frazione di secondo. Inoltre, le reti SNAP sono auto-riparanti, se un nodo non riesce per qualsiasi motivo a comunicare, altri nodi invieranno automaticamente i segnali cambiando il percorso intorno al nodo in difficoltà.

4.8.1 Reti wireless basate su SNAP

Quando si tratta di creare applicazioni che devono funzionare su reti wireless, è possibile utilizzare il tradizionale approccio di programmazione dei dispositivi *Embedded* utilizzando il linguaggio C o il linguaggio assembly, utilizzando hardware e software speciali, ma ciò richiede un alto livello di competenze sulla programmazione di reti *wireless*, quindi l'alternativa è di utilizzare la programmazione e il linguaggio di *scripting* in Python. Un moderno linguaggio di alto livello, che può essere imparato in pochi giorni, con conseguenti notevoli aumenti di produttività Snappy è un sottoinsieme del linguaggio Python che è stato ottimizzato per l'utilizzo in bassa potenza per dispositivi *Embedded*, che offre uno sviluppo di applicazioni *Embedded* senza precedenti con una produttività elevata.

4.8.2 Sviluppo e distribuzione di applicazioni Snappy

Per configurare e gestire la rete, può essere utilizzato il Portal, un software che può fornire funzioni aggiuntive come la registrazione dei dati, il monitoraggio e il *debugging*. Piccolo, veloce ed estremamente efficiente (è scritto in Python), il Portal è visto dalla rete come "un altro nodo". Il PC

con il Portal può essere fisicamente collegato a qualsiasi nodo SNAP presente nella rete, ed agisce in seguito come un ponte nella rete. Quando un gruppo di nodi sono alimentati, verranno automaticamente messi in rete, indipendentemente dal fatto che il Portal sia attualmente presente sulla rete. Quando Portal è collegato alla rete, può essere utilizzato per configurare la rete stessa, monitorare e registrare tutte le attività richieste dall'utente. Quindi se il Portal è disconnesso dalla rete, la rete continua a funzionare da sola. Gli script in Python, sono composti di una o più funzioni. Ogni funzione, in ogni nodo della rete, può essere chiamato da qualsiasi Portal o altro nodo della rete. Allo stesso modo, ogni nodo della rete può invocare uno script in esecuzione presente sul Portal. È quindi possibile richiamare la fonte di tale script, apportare una modifica, e caricare il nuovo codice nel nodo bersaglio “*over-the-air*” in poco tempo.

4.8.3 Sleepy, ultra mesh e routig a bassa potenza

In molti ambienti è necessario che i nodi che fanno parte della rete vengano alimentati da delle batterie, questo comporta un consumo di energia e può essere un problema significativo. Se un nodo è costantemente attivo (sia per "parlare", sia se è in "Ascolto"), due batterie AA saranno in grado di alimentare il nodo per solo un paio di giorni. Questo non è accettabile in termini di fabbisogno di risorse (qualcuno che deve occuparsi della sostituzione delle batterie) ed anche in termini di spese (le batterie non sono a buon mercato), per la stragrande maggioranza degli impianti, soprattutto nel caso di reti che coinvolgono centinaia o migliaia di nodi *Wireless*. La soluzione sta nell'alternare i nodi tra l'essere "svegli" per un breve lasso di tempo e quindi entrando successivamente in uno stato di *sleep* in cui si ha un consumo molto più basso.

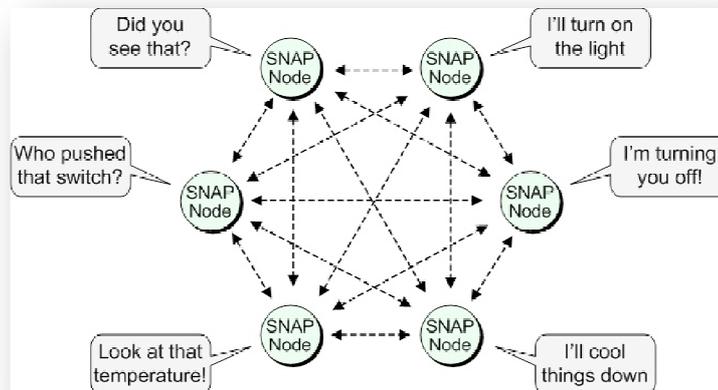


Figura 21 – Rete SNAP nella modalità “sveglio”

Si osservi che una rete di nodi SNAP, è dotata di protocolli di comunicazione “*peer-to-peer*”, il che significa che essi costituiscono la rete senza la necessità di un router speciale o nodi ponte. Le reti *mesh*, hanno difficoltà ad attuare completamente uno scenario dove tutti i nodi della rete dormono, mentre, una rete SNAP può essere facilmente configurata per implementare una “*sleepy mesh*” in cui tutti i nodi di vanno a dormire nello stesso tempo (figura 22).

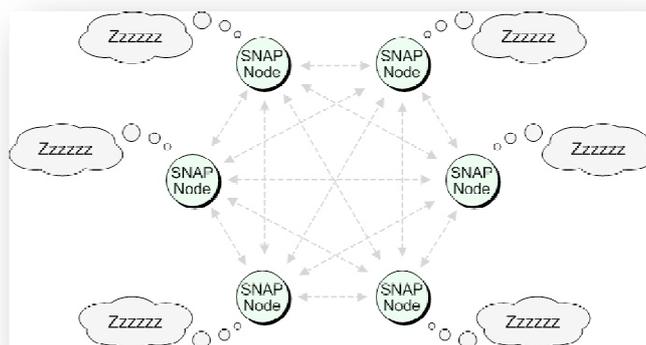


Figura 22 - Rete SNAP nella modalità *sleepy mesh*

Il risparmio energetico fornito dai nodi nel “sonno” in una “*sleepy mesh*” può essere davvero sorprendente, infatti a seconda delle esigenze, il risultato può essere quello di prolungare la durata della batteria di ciascun nodo da poco più di due giorni per più di un anno.



5 Prototipo di un sistema attivo di tracciamento

5.1 Panoramica

Il progetto sviluppato, consiste nel realizzare un prototipo di un sistema di tracciamento (*tracking system*) che utilizzi moduli RFID attivi. Tale progetto è basato su tre principali componenti hardware, e 5 componenti software; viene utilizzata una tipologia “*mesh networking*” e solo un unico dispositivo deve essere fisicamente collegato ad un PC, mentre tutti gli altri richiedono solo apporto di energia elettrica la quale potrebbe essere anche generata tramite l’uso di piccoli pannelli solari.

5 Prototipo di un sistema attivo di tracciamento

5.1.1 Hardware

- Tag, è l'elemento in movimento, che deve essere monitorato e tracciato (attivo dispone di batterie per l'alimentazione).
- Nodi Reader, rilevatore situato nella zona in cui le attività dei Tag devono essere monitorate.
- PC Reader, collegato al PC, riceve i dati dai nodi lettore (Reader) e li invia al PC.
- SNAPstick, modulo USB progettato per consentire velocemente e facilmente la connessione dei moduli al PC e alla rete *Wireless* SNAP

Il componente hardware principale utilizzato è il Synapse RF Engine, un modulo *transceiver* da 2.4 GHz con supporto incorporato per l'esecuzione degli script Python, due UART, un certo numero di GPIO, e una modalità di alimentazione a bassa tensione.

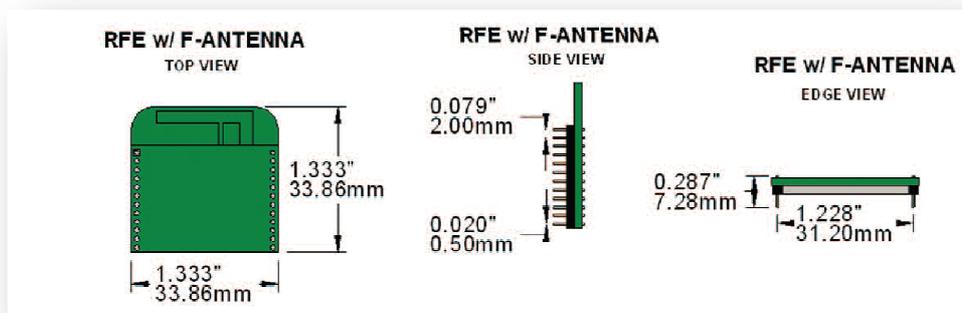


Figura 23 - Dimensioni del modulo RF Engine della Synapse

5 Prototipo di un sistema attivo di tracciamento

Specifications		
Performance	Indoor Range	Up to 1000 ft. (** 200 ft.)
	Outdoor LOS Range	Up to 3.0 miles (** up to 3000 ft.)
	Transmit Power Output	18 dBm (** 0 dBm)
	RF Data Rate	250,000 bps
	Receiver Sensitivity	-102 dBm (1% PER)
Power	Supply Voltage	2.7 – 3.4V
Requirements	Transmit Current (Typ)	110 mA (** 40 mA)
	Receive Current (Typ)	65 mA
	Idle Current (Typ)	15 mA
	Sleep Current (Typ)	2.5 μ A
General	Frequency	ISM 2.4 GHz
	Spreading Method	Direct Sequence
	Modulation	O-QPSK
	Dimensions	1.333" x 1.333"
	Operating Temperature	-40 to 85 deg C.
	Antenna Options	Integrated F, External RPSMA
Networking	Topology	SNAP or ZigBee
	Number of Channels	16
Available I/O	UARTS with HW Flow Control	2 ports – 8 total I/O
	GPIO	19 total, 8 can be analog in with 10-bit ADC
Agency Approvals	FCC Part 15.247	Yes, Class B
	Industry Canada (IC)	Yes

Figura 24 - Specifiche tecniche dei moduli Synapse

5 Prototipo di un sistema attivo di tracciamento

Pinout			
Pin No.	Name	Direction	Description
1	GND	-	Power Supply/Return
2	GPI00_TPM1CH2	Bidirectional	GPI/O, or Timer1 Channel 2 (PWM)
3	GPI01_KBI0	Bidirectional	GPI/O, Keyboard In
4	GPI02_KBI1	Bidirectional	GPI/O, Keyboard In
5	GPI03_RX_UART0	Input	UART0 Data In
6	GPI04_TX_UART0	Output	UART0 Data Out
7	GPI05_KBI4_CTS0	Bidirectional	GPI/O, Keyboard In, or UART0 CTS
8	GPI06_KBI5_RTS0	Bidirectional	GPI/O, Keyboard In, or UART0 RTS
9	GPI07_RX_UART1	Input	UART1 Data In
10	GPI08_TX_UART1	Output	UART1 Data Out
11	GPI09_KBI6_CTS1	Bidirectional	GPI/O, Keyboard In, or UART1_CTS
12	GPI010_KBI7_RTS1	Bidirectional	GPI/O, Keyboard In, or UART1_RTS
13	GPI011_AD7	Bidirectional	GPI/O, or Analog In
14	GPI012_AD6	Bidirectional	GPI/O, or Analog In
15	GPI013_AD5	Bidirectional	GPI/O, or Analog In
16	GPI014_AD4	Bidirectional	GPI/O, or Analog In
17	GPI015_AD3	Bidirectional	GPI/O, or Analog In
18	GPI016_AD2	Bidirectional	GPI/O, or Analog In
19	GPI017_AD1	Bidirectional	GPI/O, or Analog In
20	GPI018_AD0	Bidirectional	GPI/O, or Analog In
21	VCC	-	Power Supply
22	Reserved	-	-
23	RESET_L	Bidirectional	Module Reset, Active Low
24	GND	-	Power Supply/Return

Figura 25 - Descrizione dei 24 pin presenti su ogni modulo

5 Prototipo di un sistema attivo di tracciamento

5.1.2 Software

- Tag - il software in esecuzione sul Tag, serve per inviare un "ping" a un intervallo predefinito, per poi mettere il Tag in *sleep* per qualche secondo prima di rinviare nuovamente un altro *ping* segnalando così la propria posizione grazie all'intensità ed alla qualità del segnale inviato.
- Nodi Reader - il software in esecuzione sul nodo lettore, serve per ascoltare il "ping" effettuato dal Tag, e dopo averlo ricevuto, inviare il Tag ID, lettore nodo ID, e la potenza del segnale ricevuto al PC Reader.
- PC Reader - il software in esecuzione sul PC Reader serve per ricevere i dati via etere dai nodi lettore, e li trasmette alla porta seriale (USB) del PC al quale è connesso.
- Seriale del PC e spedizione dati HTTP - il software in esecuzione sul PC che attende i dati dalla porta seriale/USB, e li trasmette tramite HTTP a un server.
- display GUI - software basato sul Web per elaborare i dati del Tag in un determinato percorso e visualizzarlo su una mappa.

Il software utilizzato dal Tag , dai Nodi Reader e dal PC Reader, è strutturato in Python. Il C# è utilizzato dalla porta seriale (USB) per la spedizione dei dati HTTP, ed il PHP, l'HTML, il Java Script, e l'SVG (tecnologia in grado di visualizzare oggetti di grafica vettoriale su figure bidimensionali statiche o animate) vengono utilizzati per la trasformazione, localizzazione e il display dell'intero processo sul web.

5.2 System design

I Nodi Reader, vengono posti in posizioni fisse e conosciute in ogni zona in cui si vogliono monitorare i Tag. L'approccio per il tracciamento del Tag, consiste nel segnalare il Tag più vicino al lettore, determinandolo in base alla forza del segnale (*Link Quality*). Questo significa che i lettori devono essere posti in ogni ambiente (stanze) monitorato (idealmente al centro), ed è possibile determinare se un Tag è, o non è in quella stanza, ma non la sua esatta posizione nella stanza. Utilizzando un sistema del genere, dove i dati ricevuti vengono catalogati e passati ad un data base su un server Web, con la visualizzazione direttamente da un qualsiasi *browser*, si hanno una serie di vantaggi:

- La visualizzazione GUI è multiplatforma (si può accedere tramite i browser più moderni, da qualsiasi PC).
- Nessun software da installare sul PC Client per la visualizzazione della mappa.
- Più utenti possono accedere e visualizzare la mappa nello stesso istante.

L'uso del "*mesh networking*" implica che solo un singolo dispositivo deve essere direttamente collegato ad un PC, mentre tutti gli altri hanno solo bisogno di essere all'interno del campo d'azione di una qualsiasi altra unità vicina al PC o vicino ad un altro Reader che è vicino al PC. Ciò rende l'impostazione del sistema molto più semplice in quanto non vi è alcuna necessità di legare ogni nodo al PC/rete o di avere un PC accanto a ogni nodo. I nodi richiedono solo 3V di energia elettrica.

5.3. System hardware

5.3.1 Tag hardware



Figura 26 - Tipologie di moduli RF Engine

L'hardware, è praticamente lo stesso per ogni componente, varia solo il tipo di programma caricato in esso e la fonte di alimentazione. I tre Tag in figura, differiscono solo per un particolare : il primo è un normale modulo RF Engine con antenna integrata (RFE), il secondo è un modulo RF Engine con antenna integrata e un amplificatore di potenza (RFET), ed il terzo, come si può ben notare, ha il connettore SMA (inversione di polarità) per un'antenna esterna. Tutti dispongono di 24 piedini ed al pin 21 (+) e al pin 24 (-) viene collegata l'alimentazione (2 batterie da 1,5 V).

5 Prototipo di un sistema attivo di tracciamento

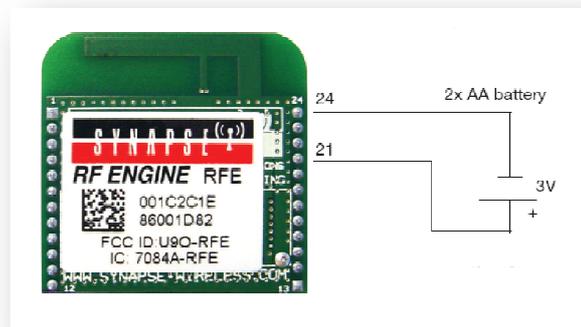


Figura 27 - Pin per l'alimentazione del modulo

5.3.2 Nodo Reader hardware

I Reader, sono costituiti esattamente dagli stessi componenti utilizzati per il Tag ma con una programmazione differente. Esattamente come i Tag, i Reader hanno bisogno dell'alimentazione (pin 21 e 24), ma essendo fissi, invece delle batterie sarebbe meglio utilizzare il "Modulo di interfaccia USB SNAPstik" per evitare i lavori di saldatura, con un alimentatore USB per esempio sotto forma di spina (come mostrato in figura) o anche cablato.

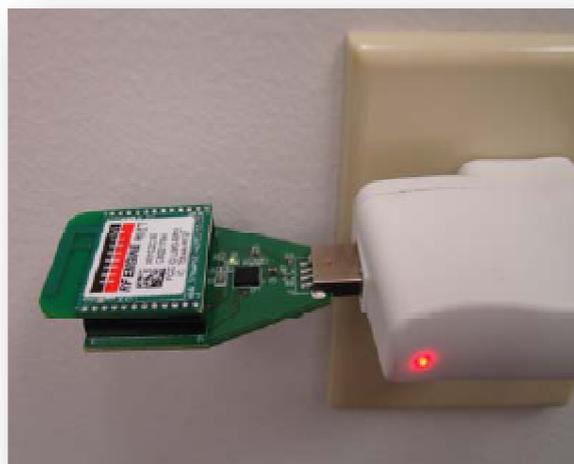


Figura 28 - Alimentazione del modulo tramite adattatore CA USB

5 Prototipo di un sistema attivo di tracciamento

I moduli possono essere così facilmente e rapidamente rimossi ad esempio per la riprogrammazione anche se comunque è possibile riprogrammarli *over-the-air*.

5.3.3 PC Reader hardware



Figura 29 - PC Reader collegato alla porta seriale virtuale (adattatore cavo USB)

Il PC Reader, è costituito dallo stesso modulo RF utilizzato per gli altri componenti, ma è l'unico che deve essere costantemente collegato ad un PC. Per semplificare il processo di collegamento del modulo della Synapse al PC, è stato usato un cavo USB il quale simula una porta seriale virtuale, mentre l'altra estremità si collega al pin UART sul modulo Synapse. In termini di potenza, per evitare la necessità di un alimentatore esterno sono stati apposti dei diodi per convertire l'energia di 5V fornita dalla porta USB ai 3V necessari per il corretto funzionamento del modulo.

5 Prototipo di un sistema attivo di tracciamento

5.3.4 SNAPstick hardware

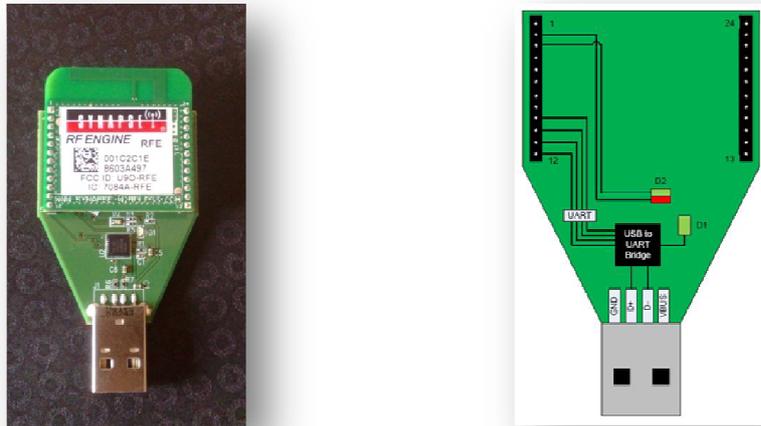


Figura 30 - Snap Stick Synapse

SNAPstick Synapse è stato progettato per essere un modo compatto e facile per connettere un PC a una rete *wireless* SNAP. Il modulo supporta tutte le forme esistenti di Synapse RF ed è completamente compatibile con il software di gestione (Portal). E' dotato di un indicatore di output, un LED a tre colori che ha la capacità di emettere differenti colorazioni: rosso, verde, ambra o luce. Si può controllare tramite script Snappy (in esecuzione sulla SNAPstick) che manipolano i pin GPIO 0 e 1.

5 Prototipo di un sistema attivo di tracciamento

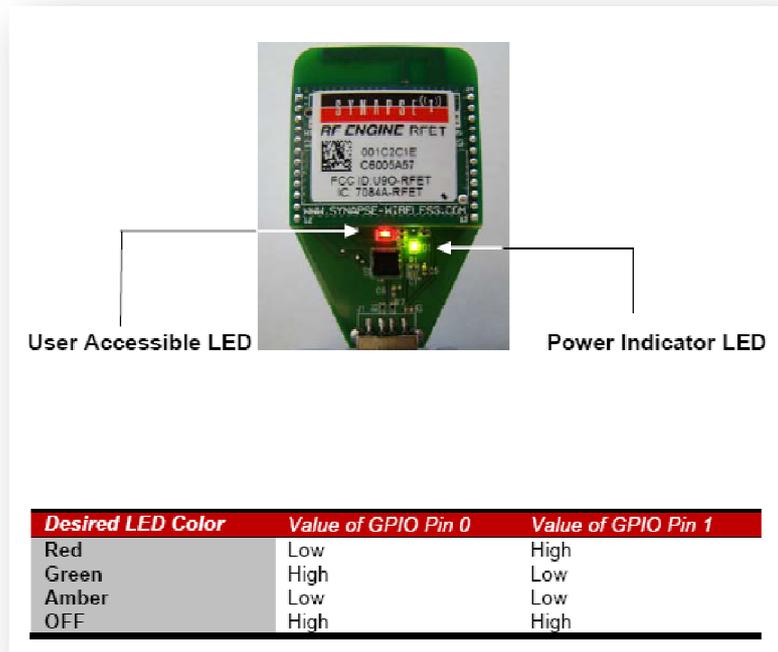


Figura 31 - Possibili combinazioni e colorazioni per l'utilizzo dei Led

Un secondo LED verde è usato per indicare che il modulo è alimentato e non può essere controllato dall'utente. L'interfaccia USB di SNAPstick comunica con il modulo RF Engine tramite collegamento interno alla UART 1. Questa UART è collegata al pin GPIO 7-10.

Pin Name	Direction of Pin	Description
GPIO 7	Input	UART1 Rx Data
GPIO 8	Output	UART1 Tx Data
GPIO 9	Bidirectional	UART1 CTS
GPIO 10	Bidirectional	UART1 RTS

Figura 32 – Pin di collegamento per UART 1

5 Prototipo di un sistema attivo di tracciamento

SNAPstick può essere alimentato con una qualsiasi forma standard di connessione *USB-powered* (ad esempio: un PC/porta *laptop*, un *hub* USB, o un adattatore CA USB), e non richiede software o driver da installare se utilizzato solamente per trarre energia dalla porta USB.

5.4 System software

5.4.1 Tag software

Il software programmato in Python e caricato nel Tag esegue 5 funzioni principali:

1. Imposta il livello di potenza di trasmissione per i Tag.
2. Inizializza tutti i GPIO per ottimizzare il consumo di corrente nella modalità *sleep*.
3. Disattiva l'inoltro di messaggi per gli altri dispositivi in rete (è solo essere acceso di una piccola quantità di tempo ogni intervallo specificato).
4. Invia un *ping* (via *multicast* RPC) a tutti i nodi lettore(PC Reader) nel campo d'azione del modulo con l'ID del Tag, e un TTL (Time To Live) pari a 1.
5. Mette il modulo in modalità *sleep*, ad esempio per 3 secondi.

Il software è caricato sul Tag tramite il Portal. Quando il Tag viene programmato per la prima volta, sarebbe meglio collegarlo allo SNAPstick USB, piuttosto che usare l'alimentazione a batteria, perché lasciando il

5 Prototipo di un sistema attivo di tracciamento

modulo sul banco di lavoro non ancora programmato, anche solo per pochi minuti, si avrebbe un rapido esaurimento delle batterie.

Il codice del Tag

```
#####  
# Active RFID People / Asset Tracking System  
#####  
  
from synapse.nvparams import *  
from synapse.switchboard import *  
  
def startup():  
    global seccounter  
  
    # Imposta il livello di trasmissione al massimo.  
  
    txPwr(17)  
  
    # Disabilita l'interfaccia di comunicazione seriale UART 0  
  
    initUart(0, 0)  
  
    flowControl(0, False)  
  
    # Disabilita l'interfaccia di comunicazione seriale UART 1  
  
    initUart(1, 0)  
  
    flowControl(1, False)
```

5 Prototipo di un sistema attivo di tracciamento

```
# Impostazione dei piedini per limitare l'utilizzo di corrente
```

```
setPinDir(0, 0)
```

```
setPinPullup(0, 1)
```

```
setPinDir(1, 0)
```

```
setPinPullup(1, 1)
```

```
setPinDir(2, 0)
```

```
setPinPullup(2, 1)
```

```
setPinDir(3, 0)
```

```
setPinPullup(3, 1)
```

```
setPinDir(4, 0)
```

```
setPinPullup(4, 1)
```

```
setPinDir(5, 0)
```

```
setPinPullup(5, 1)
```

```
setPinDir(6, 0)
```

```
setPinPullup(6, 1)
```

```
setPinDir(7, 0)
```

```
setPinPullup(7, 1)
```

```
crossConnect(DS_NULL, DS_PACKET_SERIAL)
```

```
setPinDir(8, 0)
```

```
setPinPullup(8, 1)
```

```
setPinDir(9, 0)
```

```
setPinPullup(9, 1)
```

```
setPinDir(10, 0)
```

5 Prototipo di un sistema attivo di tracciamento

```
setPinPullup(10, 1)

setPinDir(11, 0)

setPinPullup(11, 1)

setPinDir(12, 0)

setPinPullup(12, 1)

setPinDir(13, 0)

setPinPullup(13, 1)

setPinDir(14, 0)

setPinPullup(14, 1)

setPinDir(15, 0)

setPinPullup(15, 1)

setPinDir(16, 0)

setPinPullup(16, 1)

setPinDir(17, 0)

setPinPullup(17, 1)

setPinDir(18, 0)

setPinPullup(18, 1)

# Disabilita l'inoltro dei pacchetti verso nodi non in visibilità diretta

saveNvParam(NV_MESH_OVERRIDE_ID, 1)

def poll100ms():

# Stampa il ping inviato

mcastRpc(1, 1, "tagging", localAddr())
```

5 Prototipo di un sistema attivo di tracciamento

```
#Torna in modalità sleep

sleep(0, 3)

# Impostazione degli handlers

snappyGen.setHook(SnapConstants.HOOK_STARTUP, startup)

snappyGen.setHook(SnapConstants.HOOK_100MS, poll100ms)
```

5.4.2 Reader software

Il Software per il Reader, è incaricato di ricevere un *ping multicast* dal Tag, ritrasmetterlo al modulo collegato al PC(PC Reader), con la potenza del segnale del Tag che ha inviato il *ping* (lq), e l'ID del lettore stesso.

Il codice del Reader

```
#####

# Active RFID People / Asset Tracking System

#####

# Variabile che indica l'indirizzo del reader

PC_LINKED_READER = '\x03\xa5\x71'

# Riceve il ping dal Tag

def tagging(address):
```

5 Prototipo di un sistema attivo di tracciamento

```
# Legge la qualità del segnale

lq = getLq()

energy = getEnergy()

#Invia il tag id, il segnale, l'id reader connesso al server (via mesh network)
rpc(PC_LINKED_READER, 'tagging_collect', address, lq, energy, localAddr())
```

5.4.3 PC Reader software

Il software del PC Reader, gestisce la ricezione dei dati inviati dai nodi lettore (i quali a loro volta ricevono i dati dai Tag). Si inizializza la porta seriale, con la connessione alla UART 1 per STDIN/STDOUT, quindi inizializza a 9600 *baud*. Il software stampa i dati dei ping del Tag sulla porta seriale (dopo la decodifica dei valori esadecimali in testo semplice).

Esempio: **003c02**, **001a06**, **57**

Tag [**id**], lettore [**id**], [**lq**] la potenza del segnale

5 Prototipo di un sistema attivo di tracciamento

Il codice del PC Reader

```
#####  
  
# Active RFID People / Asset Tracking System  
  
#####  
  
from synapse.switchboard import *  
  
#Inizializzazione  
  
def startup():  
    initUart(1, 9600) #Set UART1 to 9600 baud  
  
    #Connette la porta seriale con la STDIO  
  
    crossConnect(DS_UART1, DS_STDIO)  
  
    print "Reader Powered Up"  
  
#Funzione che invia I dati verso la UART 1  
  
def tagging_collect(tag_addr, lq, energy, reader_addr):  
  
    #Stampa la stringa che poi viene inviata al server  
  
    print bin2hex(tag_addr), ",", bin2hex(reader_addr), ",", lq, ",", energy  
  
#Conversione in esaddecimale  
  
def bin2hex(str):  
    hex = ""  
  
    count = len(str)  
  
    index = 0  
  
    while index < count:
```

5 Prototipo di un sistema attivo di tracciamento

```
byte = ord(str[index])

hex += bin2hex_nibble(byte >> 4)

hex += bin2hex_nibble(byte)

index += 1

return hex

def bin2hex_nibble(nibble):

    hexStr = "0123456789ABCDEF"

    return hexStr[nibble & 0xF]

# Impostazione degli handlers

snappyGen.setHook(SnapConstants.HOOK_STARTUP, startup)

#snappyGen.setHook(SnapConstants.HOOK_STDIN, stdinHandle)
```

5.4.4 Inoltro dei dati da seriale del PC in HTTP

Per consentire il passaggio dei dati all'applicazione finale che li gestirà è necessario catturare i dati inviati al PC Reader tramite la porta COM ed inviarli tramite HTTP alla pagina di gestione dell'applicazione. Di questo, se ne occupa il “*forwarder*”, un piccolo programma scritto in C# il quale acquisisce dalla porta COM selezionata (figura 33, punto 1) i dati catturati e formattati come semplici stringhe e li invia all’indirizzo del server selezionato (figura 33, punto 2) dove l'applicazione finale in grado di comprendere tali dati li immagazzina in un data base. Tale scelta si è rivelata piuttosto semplice ma consente di poter realizzare un prototipo in

5 Prototipo di un sistema attivo di tracciamento

breve tempo. Il formato della stringa catturata ed inviata ad intervalli regolari al server è:

```
“ print bin2hex(tag_addr), ",", bin2hex(reader_addr), ",", "lq, ”.
```

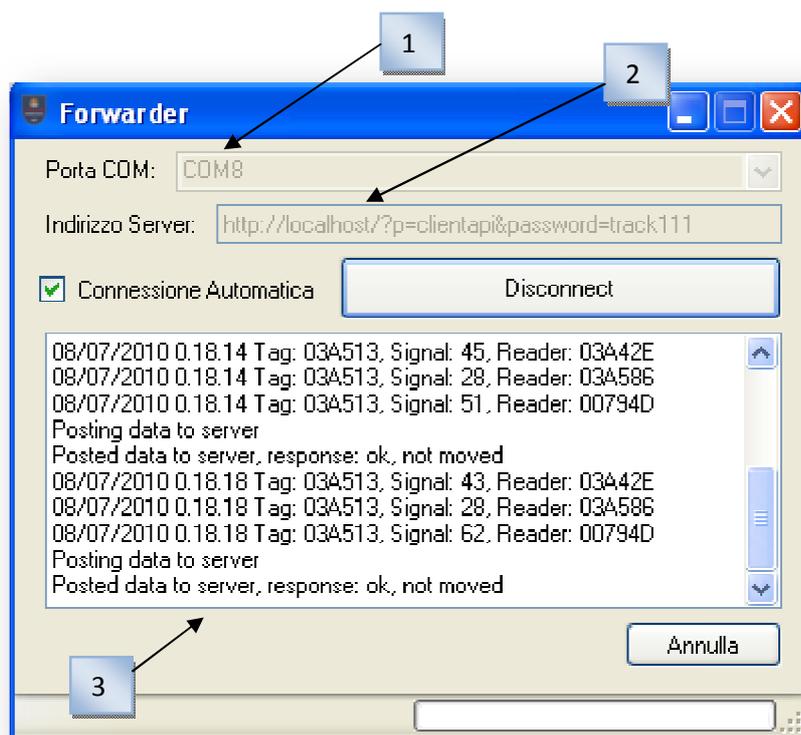


Figura 33 - Screenshot del Forwarder connesso

Nel punto 3 della figura 33, notiamo la casella di testo, che visualizza la stringa formattata, ricevuta dalla seriale, infatti stampa prima la data e l'ora, poi l'indirizzo del Tag, la qualità del segnale ricevuto dal Reader, e l'indirizzo del Reader al quale si riferisce il segnale. La dicitura “*not moved*”, sta a significare che il *link quality* non è cambiato e quindi il Tag non ha cambiato la propria posizione.

5 Prototipo di un sistema attivo di tracciamento

I Reader, vengono associati alle stanze, grazie all'interfaccia grafica ed al *database* che associa la posizione sulla mappa, all'indirizzo di rete.

Live Map					
Tags	Readers	Maps	Users	Logout	
Page 1 of 1					
Name ▼		Node Address	Map	X Pos	Y Pos
Auditorium		00794D	Mappa Universita'	350	100
Aula B		03A586	Mappa Universita'	250	290
Aula C		000001	Mappa Universita'	470	290
Aula Prof.		03A42E	Mappa Universita'	650	80
Aula studio		000004	Mappa Universita'	65	80
Aula studio 1		000005	Mappa Universita'	635	360

Live Map		Tags	Readers	Maps
Page 1 of 1				
Name ▼		Node Address		
Auditorium		00794D		
Aula B		03A586		
Aula C		000001		
Aula Prof.		03A42E		
Aula studio		000004		
Aula studio 1		000005		

Figura 34 - Node Address associati alle stanze

5 Prototipo di un sistema attivo di tracciamento

5.4.5 Display GUI

La sigla GUI corrisponde al termine inglese *Graphical User Interface*, comunemente abbreviata in interfaccia grafica, ed è la parte di codice responsabile della conversione dei dati spediti dal “forwarder” in un segnale di posizione sulla mappa, e del salvataggio dei dati in un database.

L’interfaccia fa uso di alcuni *plug-in* jQuery e java Script per animare dinamicamente le nuove posizioni del Tag sullo schermo. Quando l’interfaccia è aperta e la mappa viene caricata, i Tag ripercorreranno le ultime due posizioni (salvate nel *database*), fermandosi nella posizione corrente.

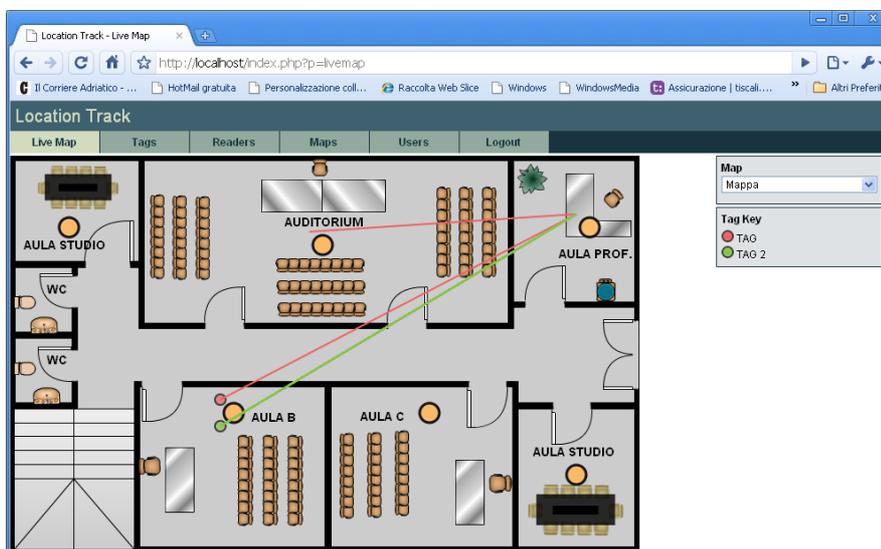
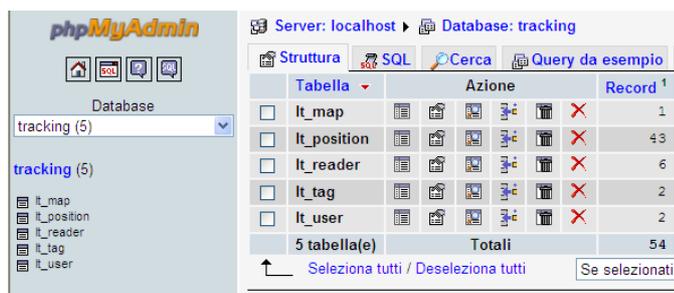


Figura 35 - Esempio di due Tag fermi in posizione corrente dopo aver ripercorso le ultime due posizioni

5 Prototipo di un sistema attivo di tracciamento

Il *database* utilizzato, è strutturato in maniera molto semplice; è formato da cinque tabelle (figura 36) suddivise in :

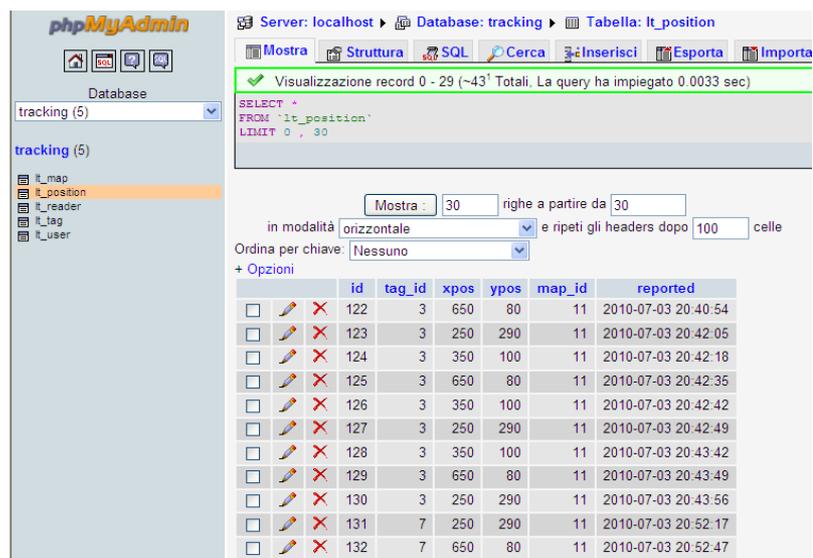
- *it_map* dove vengono memorizzate le mappe da caricare.
- *it_position* qui vengono memorizzati: il Tag, la mappa, la posizione sulla stessa (x,y) e infine la data e l'ora (figura 37).
- *it_reader* in questa tabella, vengono memorizzati i Reader presenti sulla mappa, con le posizioni e l'indirizzo di rete.
- *it_tag* memorizza il nome e il colore dei Tag presenti.
- *it_user* qui memorizza la mail, la password e gli ultimi accessi effettuati.



The screenshot shows the phpMyAdmin interface for the 'tracking' database. The left sidebar lists the tables: it_map, it_position, it_reader, it_tag, and it_user. The main area displays a table with columns: Tabella, Azione, and Record. The data is as follows:

Tabella	Azione	Record
<input type="checkbox"/> it_map	[Icons]	1
<input type="checkbox"/> it_position	[Icons]	43
<input type="checkbox"/> it_reader	[Icons]	6
<input type="checkbox"/> it_tag	[Icons]	2
<input type="checkbox"/> it_user	[Icons]	2
5 tabella(e) Totali		54

Figura 36 - Struttura del Database



The screenshot shows the phpMyAdmin interface for the 'it_position' table. The table is displayed with columns: id, tag_id, xpos, ypos, map_id, and reported. The data is as follows:

	id	tag_id	xpos	ypos	map_id	reported
<input type="checkbox"/>	122	3	650	80	11	2010-07-03 20:40:54
<input type="checkbox"/>	123	3	250	290	11	2010-07-03 20:42:05
<input type="checkbox"/>	124	3	350	100	11	2010-07-03 20:42:18
<input type="checkbox"/>	125	3	650	80	11	2010-07-03 20:42:35
<input type="checkbox"/>	126	3	350	100	11	2010-07-03 20:42:42
<input type="checkbox"/>	127	3	250	290	11	2010-07-03 20:42:49
<input type="checkbox"/>	128	3	350	100	11	2010-07-03 20:43:42
<input type="checkbox"/>	129	3	650	80	11	2010-07-03 20:43:49
<input type="checkbox"/>	130	3	250	290	11	2010-07-03 20:43:56
<input type="checkbox"/>	131	7	250	290	11	2010-07-03 20:52:17
<input type="checkbox"/>	132	7	650	80	11	2010-07-03 20:52:47

Figura 37 - Screenshot della tabella it_position



5.5 Test e analisi dei dati

Il progetto in esame è stato realizzato grazie all'utilizzo di n° 5 moduli RF Engine identici. Come già spiegato in precedenza, è sufficiente riprogrammare il modulo per assegnargli funzionalità differenti.

Con 5 moduli, i test possibili erano due considerando che un modulo viene programmato per essere collegato al PC tramite porta seriale ed essere quindi programmato come PC Reader:

- 3 moduli programmati come Reader per il tracciamento di un Tag.
- 2 moduli programmati come Reader per il tracciamento dei due restanti moduli programmati come Tag.

Nel primo test, il Tag, è fermo nell'aula Professori (figura 38) poi come mostrato dalle figure 39 e 40, il Tag viene tracciato prima al passaggio in aula B, poi al passaggio in Auditorium; nella figura 41 il Tag ritorna nell'aula Professori E si ferma nell'aula B.

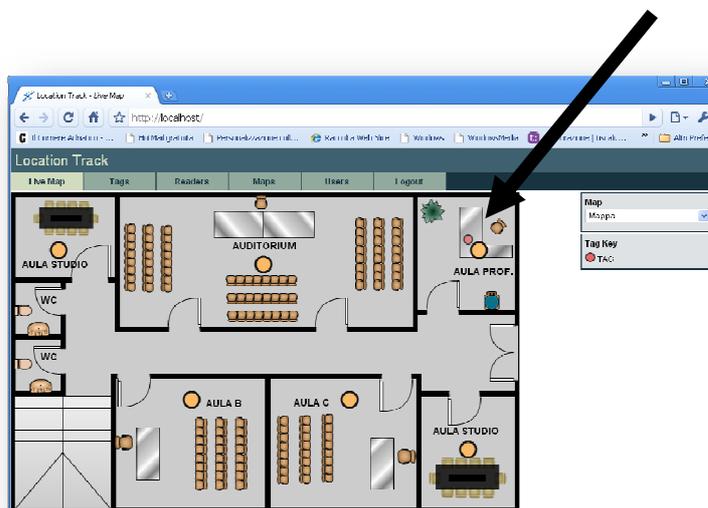


Figura 38 - Tag fermo nell'aula Professori

5 Prototipo di un sistema attivo di tracciamento

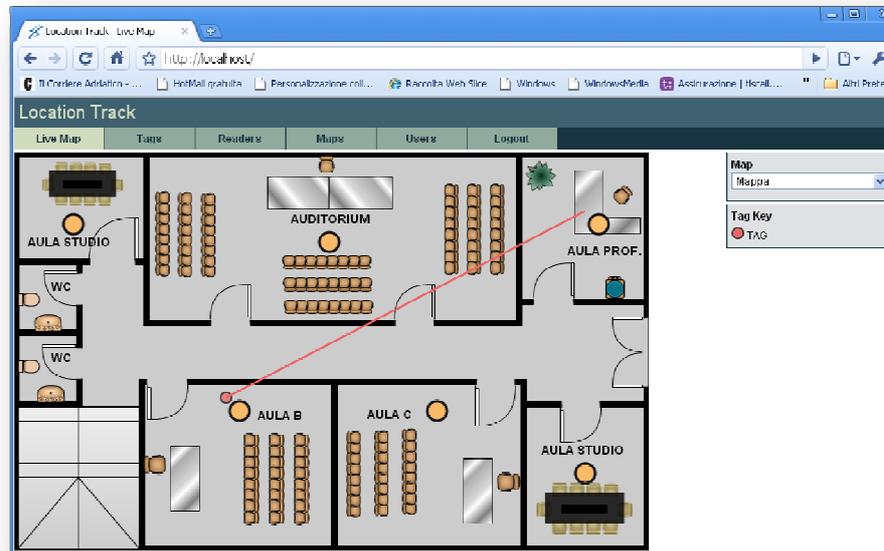


Figura 39 - Tag Aula B

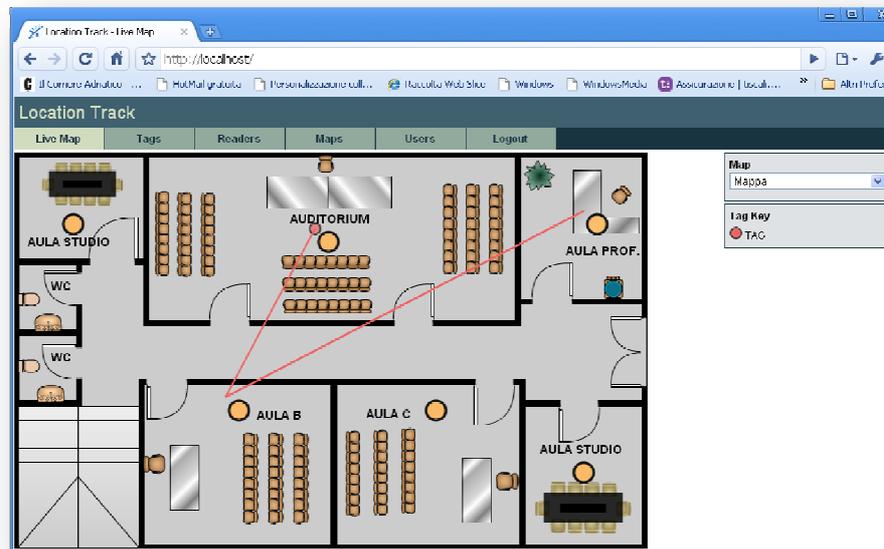


Figura 40 - Tag in Auditorium

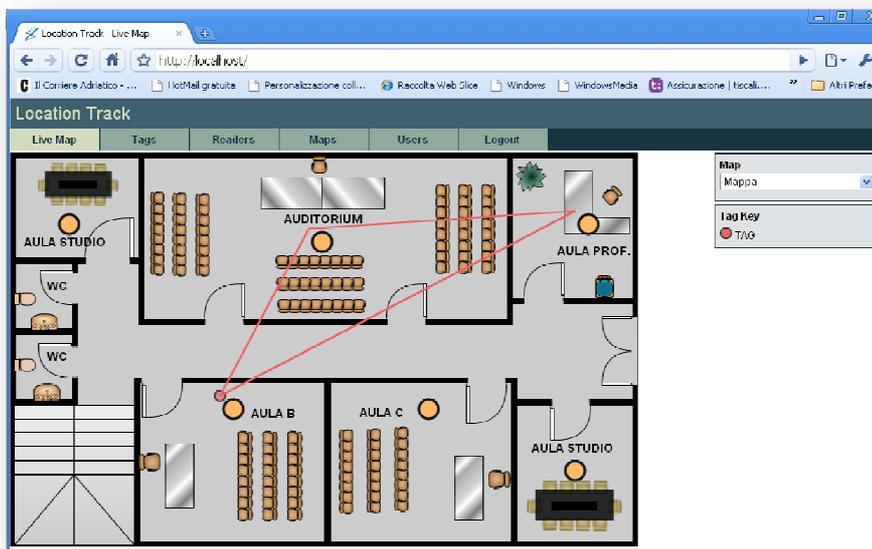


Figura 41 - Tag fermo nell'aula B

Il secondo test, consiste nel tracciare due Tag con due Reader. I Reader sono stati piazzati uno nell'aula Professori e uno nell'aula B. Il secondo Tag è di colore verde, e come si evince dalla figura 42, il Tag rosso, ha in memoria le ultime due posizioni, per cui ha un passaggio dall'Auditorium, che altrimenti sarebbe impossibile effettuare.

5 Prototipo di un sistema attivo di tracciamento

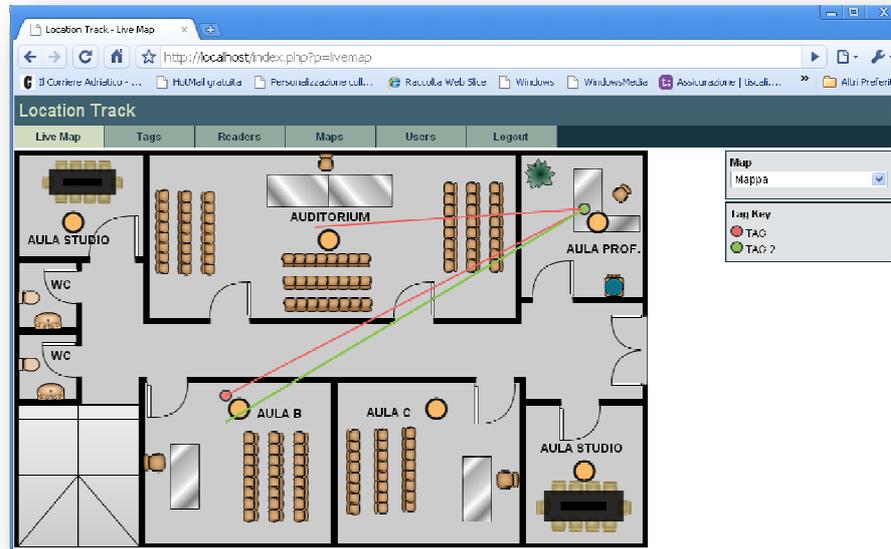


Figura 42 - Test con due Tag

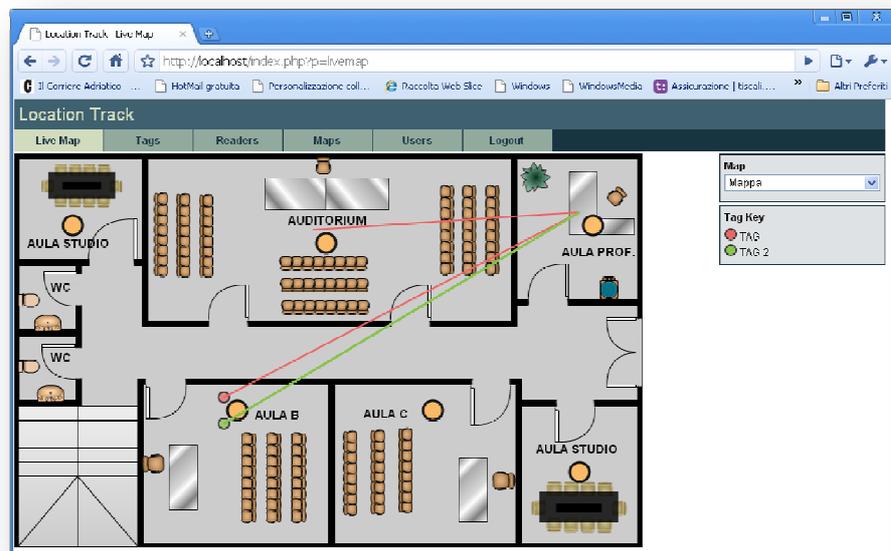


Figura 43 - Spostamento di un singolo Tag

5 Prototipo di un sistema attivo di tracciamento

Un altro test importante, è l'analisi della qualità del *Link*, cioè capire se la qualità è costante e se cambia in base ad abbassamenti di tensione o a dispersioni dovuti al fatto che i moduli sono alimentati a batterie. Quest'analisi, è stata fatta in maniera empirica, monitorando per un'ora circa un Tag e tenendolo immobile davanti ad un Reader per analizzare poi i risultati. Una correlazione tra il *link quality* e l'alimentazione, si ottiene modificando la stringa sul Reader aggiungendo la richiesta dell'indicatore di energia

```
energy = getEnergy()
```

e modificando la stringa del PC Reader che viene catturata dalla porta seriale per essere inviata al server.

```
print bin2hex(tag_addr), ",", bin2hex(reader_addr), ",", lq, ",", energy
```

I risultati, sono stati raccolti con una connessione con la porta seriale tramite "Hyper Terminal" (figura 44) e i risultati inseriti in un foglio di calcolo per poter così disegnare il grafico in figura 45.

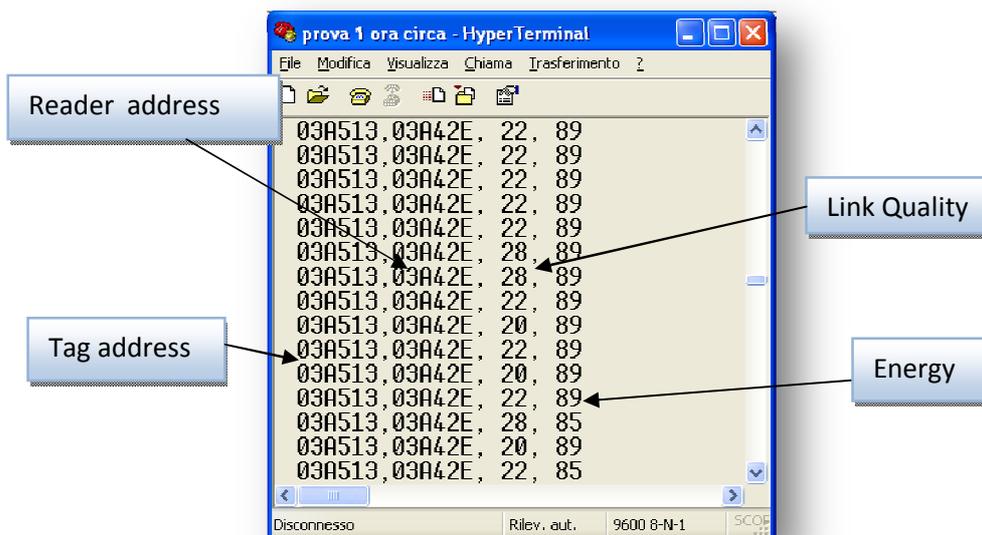


Figura 44 - Dati raccolti con Hyper Terminal

5 Prototipo di un sistema attivo di tracciamento

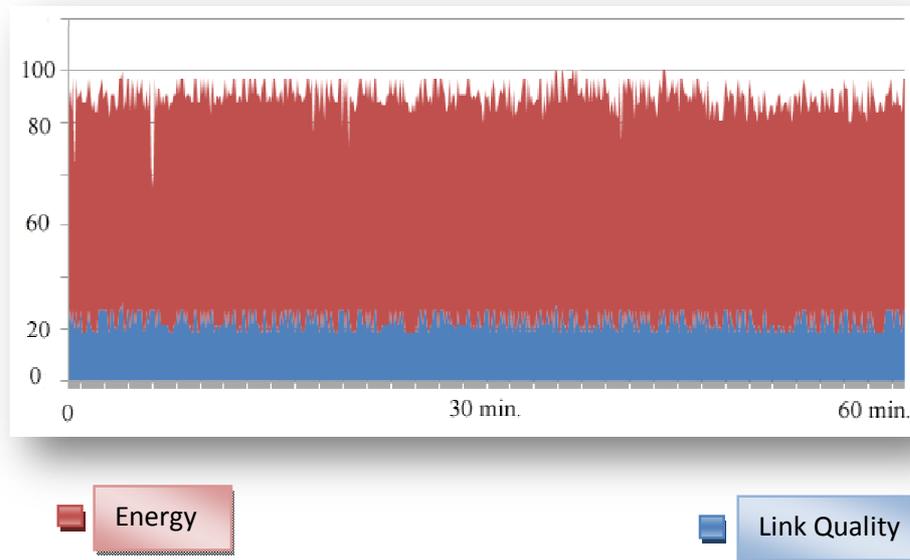


Figura 45 – Variazione *Link Quality* nel tempo

Dal grafico, si evince che la qualità del link, non è sempre la stessa, ma si stabilizza entro un *range* di 10/12 punti, senza essere influenzata dai continui abbassamenti di tensione. Quindi non vi è una diretta correlazione.



Conclusioni

La tecnologia RFID modificherà le abitudini di vita delle persone. Questa frase può riassumere l'impatto che l'introduzione di questa tecnologia dovrebbe avere in un futuro non molto lontano. Quelle lunghe file alle casse dei supermercati cui si è costretti a sottoporsi ogni volta che si fa la spesa, forse tra qualche anno non si vedranno più, grazie alla tecnologia RFID, infatti, sarà possibile calcolare il totale di quanto acquistato senza togliere la merce dal carrello e senza dover quindi passare ogni singolo oggetto sotto al lettore di codice a barre.

Le sue applicazioni sono molteplici e toccano numerosi aspetti della vita di ogni persona: servizi pubblici, lavoro, salute, tempo libero, pagamenti, trasporti, acquisti. In tutti i campi, i vantaggi sono notevoli, sia per chi produce, fornisce o vende prodotti e servizi, sia per chi acquista o usufruisce dei prodotti o dei servizi. La semplificazione della gestione degli oggetti da un lato e la riduzione dei tempi di attesa e dei costi di prodotti e servizi dall'altro, riassumono i numerosi vantaggi che si possono trarre da un uso massiccio della tecnologia RFID. Da non sottovalutare l'importanza che questa tecnologia avrà per il concetto di tracciabilità. Tutte le fasi della vita di un prodotto saranno memorizzate e sarà facile ricostruire la storia di un qualsiasi oggetto partendo da un qualunque momento della sua esistenza (realizzazione, vendita, utilizzo, assistenza, smaltimento). In questo momento la tecnologia è ancora in fase sperimentale: questo non permette la piena valutazione della fattibilità e dell'impatto che la sua applicazione potrà determinare.

Va valutato positivamente il fatto che, organismi internazionali, grandi aziende e centri di ricerca si sono già attivati affinché la tecnologia non resti

solo in fase sperimentale, ma possa ben presto diventare realtà, testimoniano le potenzialità della tecnologia per la realizzazione di servizi innovativi atti a migliorare la qualità della vita.

Fra gli ostacoli per il RFID, oltre a quelli di natura tecnologica, figurano i costi, ancora troppo alti per essere concorrenziali con quelli dei codici a barre, e le preoccupazioni per la privacy e la sicurezza dei dati.

Molto importante è l'impatto sociale, che a detta degli esperti sarà simile a quello che in passato hanno avuto internet o i terminali mobili. Fenomeni simili sono caratterizzati da un andamento non lineare: dopo un periodo di crescita lenta possono esplodere in modo dirompente, e non solo a causa dell'abbattimento dei costi. Per questo motivo le applicazioni che potrebbero riscuotere più successo sono quelle vicine alla sfera delle emozioni, come quelle legate alla sicurezza o alla salute.

Per concludere sembra largamente plausibile che, anche con l'ausilio delle tecnologie più avanzate, presto o tardi ogni prodotto venduto sarà dotato di TAG e le applicazioni RFID diverranno velocemente e senza clamore, una delle tecnologie più massivamente realizzate nella nostra epoca.

Il sistema sviluppato appare un come un primo tentativo di sviluppo di sistemi basati sulla tecnologia Synapse *Wireless*. Il prototipo del sistema di tracciamento può essere la base per la realizzazione di piccole applicazioni per la localizzazione *indoor* che non richiedano elevata precisione e per le quali sia possibile (in termini di costo e di caratteristiche hardware) utilizzare Tag attivi. Lo sviluppo del sistema ha inoltre consentito di acquisire conoscenza della tecnologia in generale, utile per future applicazioni diverse da quella proposta nella tesi.

Appendice A

Snappy vs Python

Le differenze principali tra Snappy e Python sono molteplici.

Moduli

Snappy supporta l'importazione di moduli standard predefiniti dalla libreria Python.

```
from module import * # supportato  
import module # Non supportato
```

Variabili

Sono supportate sia le variabili locali, che le variabili globali.

Funzioni

Ci sono fino a 255 funzioni "pubbliche" che possono essere definite. Queste sono richiamabili da remoto utilizzando il protocollo SNAP RPC.

Le funzioni non pubbliche sono invece limitate solo dalla dimensione della memoria *Flash* presente sul dispositivo.

Tipi di Dati

Snappy supporta i seguenti tipi fondamentali di dati di Python:

NoneType, int, bool, string, tuple, function

int è un numero intero a 16 bit

string è una stringa ed ha dimensione massima di 255 byte

tuple è un contenitore di sola lettura di altri tipi di dati, ad esempio (1,2,3)

Snappy, attualmente, non supporta i seguenti tipi comuni di Python:

float, long, complex, list, dict, set

Keywords

I seguenti identificatori riservati di Python sono supportati in Snappy:

and, from, not, while, elif, global, or, else, if, pass, break, import, print, continue, return, def, is.

I seguenti identificatori, invece non sono supportati in Snappy:

del, as, with, assert, yield, except, class, exec, in, raise, finally, for, lambda.

Operatori

Snappy supporta tutti gli operatori di Python, ad eccezione di

(*/ /*) e (****).

+ - * / %

<< >> & | ^ ~

<> <= >= == != <>

Slicing

Lo *slicing* è supportato per stringa e tupla tipi di dati. Ad esempio, se x è "ABCDE", quindi $x[1:4]$ è "BCD".

Concatenazione

La concatenazione è supportato per i tipi di dati stringa. Ad esempio,

se $x = \text{"Ciao"}$ e $y = \text{", mondo"}$, $x + y$ sarà "Ciao, mondo".

Espressioni

Snappy supporta tutti i tipi di operazioni booleane di Python, aritmetica, e le espressioni di confronto :

```
x = +1 if a > b else -1 #
```

x sarà +1 o -1 a seconda del valore di a e b .

Python Builtins

Python builtins supportati :

len, ord, chr, int, str

Inoltre, sono supportati molti moduli specifici.

Stampa

Snappy supporta anche una dichiarazione di stampa. Normalmente ogni riga di output stampato appare su una riga separata. Se non si desidera avanzare automaticamente alla riga successiva alla fine della sua dichiarazione basterà inserire un carattere di stampa, la virgola (",").

```
print "line 1"
```

```
print "line 2"  
print "line 3 ",  
print "and more of line 3"  
print "valore di x è ", x, "e y è ", y
```

Sviluppo di applicazioni Snappy

Di seguito vengono illustrate alcune delle questioni fondamentali da considerare nello sviluppo di applicazioni basate su SNAP.

Event Driven Programming

Le applicazioni in Snappy hanno spesso diverse attività in corso contemporaneamente. Ciò è possibile, nonostante ci sia una sola CPU nei dispositivi, perché la concorrenza si realizza attraverso la programmazione *event-driven*. Questo significa che le funzioni vengono eseguite rapidamente fino a compimento, e non si arriva mai al "blocco" o "*loop*" in attesa di qualcosa.

SNAP Hooks

Ci sono una serie di eventi nel sistema che possono innescare funzioni in Snappy. Nel definire gli script, c'è un modo per associare funzioni con questi eventi esterni, che si fa specificando un "*HOOK*", cioè un identificatore per la funzione. Gli *hook* sono definiti:

- ***HOOK_STARTUP*** – Chiamata per il boot del dispositivo
- ***HOOK_GPIN*** – Chiamata per la monitorizzazione della transizione hardware dei pin
- ***HOOK_1MS*** – La chiamata avviene ogni millisecondo
- ***HOOK_10MS*** – La chiamata avviene ogni 10 millisecondi

- ***HOOK_100MS*** – La chiamata avviene ogni 100 millisecondi
- ***HOOK_1S*** – La chiamata avviene ogni secondo
- ***HOOK_STDIN*** – La chiamata avviene quando "i dati di input utente sono pervenuti"
- ***HOOK_STDOUT*** – La chiamata avviene quando "i dati di output sono stati inviati"
- ***HOOK_RPC_SENT*** – La chiamata avviene quando una RPC in uscita viene inviata.

All'interno di uno script, ci sono due metodi per specificare l'*handler* corretto per un determinato evento:

- ***@ setHook ()***

Immediatamente prima della funzione che si desidera richiamare setHook @ (HOOK_codice predefinito in precedenza)

- ***snappyGen.setHook ()***

Dopo la funzione che si desidera richiamare, inseriamo una riga di codice del tipo: "snappyGen.setHook (SnapConstants.HOOK_XXX, eventHandlerXXX)" eventHandlerXXX dovrebbe essere sostituito con il vero nome della routine di gestione prevista.

Dati trasparenti (porta seriale Wireless)

SNAP supporta efficientemente il *bridging* di dati seriali attraverso una rete wireless. Le connessioni dati utilizzando la modalità trasparente possono esistere a fianco di messaggistica basata su RPC.

Scripted Serial I/O (SNAPpy STDIO)

Il metodo trasparente di SNAP (porta seriale *Wireless*) prende i dati da un'unica interfaccia e li inoltra ad un'altra interfaccia, ma i dati non vengono alterati in alcun modo. Gli script possono anche interagire direttamente con le porte seriali, permettendo di implementare protocolli seriali personalizzati.

Il flusso di dati attraverso un dispositivo SNAP è configurato tramite lo “*Switchboard*”. I seguenti comandi sono definiti nel “switchboard.py” file, e possono essere importati da altri script Snappy:

DS_NULL
DS_UART0
DS_UART1
DS_TRANSPARENT
DS_STDIO
DS_PACKET_SERIAL

Ad esempio, per configurare UART1 in modalità trasparente (*Wireless* di serie), è sufficiente inserire la seguente istruzione nel gestore di avvio Snappy: `crossConnect(DS_UART1, DS_TRANSPARENT)`

crossConnect(<i>dataSrc1,dataSrc2</i>)	Cross-connect SNAP data-sources (bidirezionale)
uniConnect(<i>dst, src</i>)	Connect src->dst SNAP data-sources (unidirezionale)

Snap API per creare le connessioni con lo Swichboard

Appendice B

Portal software

Synapse Portal è un Portale interattivo che gira su Windows, Linux o Mac ed utilizzando un cavo USB o un'interfaccia RS232, si collega a qualsiasi nodo della rete Wireless SNAP e diventa un'interfaccia grafica per l'utente per l'intera rete. E' possibile così sviluppare applicazioni utilizzando un *editor* in Python per creare gli script e per caricarli sui nodi SNAP in modalità wireless. Il Portal riesce in maniera molto rapida a visualizzare in tempo reale la rete SNAP; a configurare e monitorare i dispositivi; analizzare i canali, i dati e la registrazione degli eventi e altro ancora. Esso fornisce l'accesso completo a tutte le funzioni in ogni nodo della rete SNAP, ed attraverso il portale, i dispositivi remoti, possono accedere a tutta la potenza del PC agendo come un ponte tra la rete SNAP e il PC stesso. E' dotato di un'interfaccia semplice e flessibile basata su Python, un linguaggio di alto livello, il quale non richiede nessuna esperienza di programmazione *Embedded* in particolare.

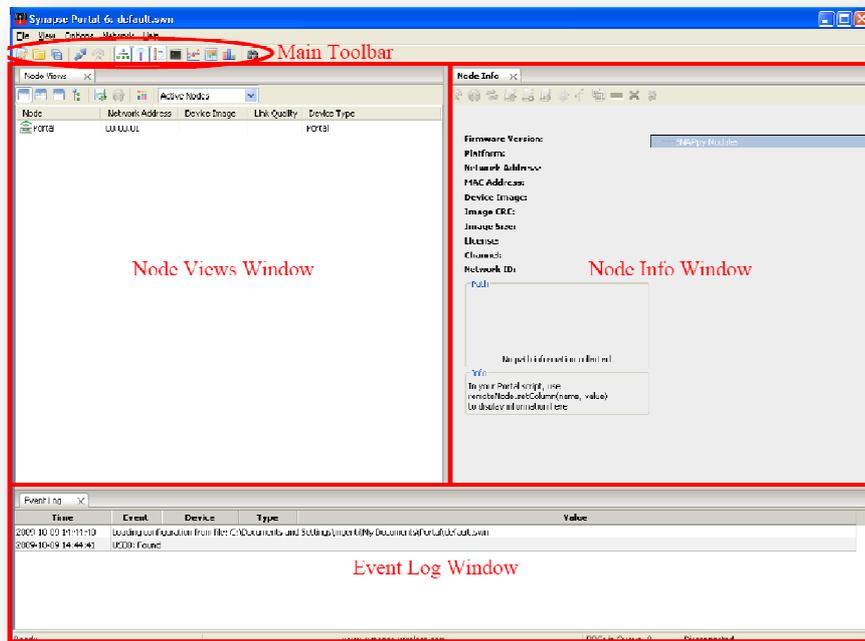
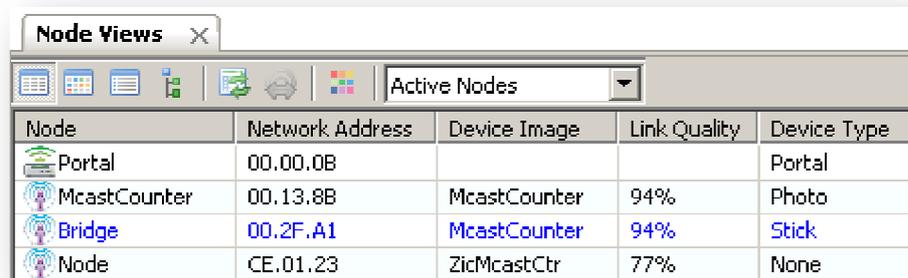


Figura 46 - Schermata principale del Portal

Tutti i nodi presenti nella rete verranno visualizzati nella finestra “*Node Views*”, e facendo clic su un nodo della lista, verranno mostrate le informazioni dettagliate su quel nodo nella finestra “*Node Info*”. Infine, il registro eventi, viene visualizzato nella finestra “*Event Log*” che tiene traccia di tutti gli eventi che si sono verificati nel sistema e nella rete SNAP.



The screenshot shows a window titled "Node Views" with a toolbar and a dropdown menu set to "Active Nodes". Below is a table with the following data:

Node	Network Address	Device Image	Link Quality	Device Type
Portal	00.00.0B			Portal
McastCounter	00.13.8B	McastCounter	94%	Photo
Bridge	00.2F.A1	McastCounter	94%	Stick
Node	CE.01.23	ZicMcastCtr	77%	None

Figura 47 - Particolare finestra Node Views

Come si evince dall'immagine qui sopra, nella finestra “*Node Views*”, la colonna “*node*” mostra tutti i nodi presenti nella rete, essi devono essere unici e a questo scopo, un identificatore numerico finale viene aggiunto al nome di base; questo è un requisito importante in Portal, dove è possibile rinominare le unità, ma ciascuna deve essere univoca.

La colonna “*Network Address*”, mostra gli indirizzi di rete per ciascun nodo; l'indirizzo è semplicemente composto dagli ultimi tre byte dell'indirizzo MAC, e non è definibile dall'utente. La colonna “*Device Image*” mostra il nome dello *script Snappy* caricato nel dispositivo. La colonna “*Link Quality*” mostra una fotografia istantanea del livello della frequenza radio ricevuta espressa in percentuale.

Per quanto riguarda il “*Link Quality*” è importante fare delle precisazioni :

1. Questo campo, non è continuamente aggiornato, salvo che non sia l'utente a chiedere al programma di farlo, utilizzando il pulsante (punto 1 in figura 48) per il *broadcast ping*, aggiornando così i campi della qualità di tutte le unità attive, oppure forzando un aggiornamento per singolo nodo, da un pulsante presente sulla barra

degli strumenti nella finestra “*Node Info*”. Infine, tramite un altro pulsante (punto 2 in figura 48), è possibile effettuare un “*broadcast ping*” automatico con un certo intervallo di tempo.

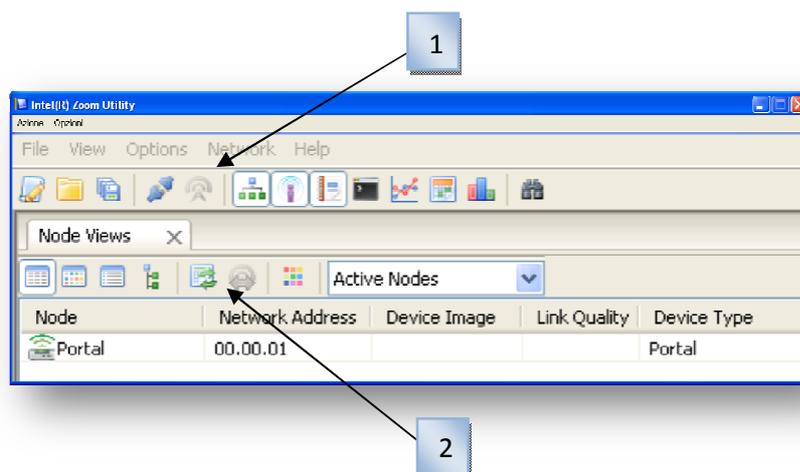


Figura 48 - Particolare della barra degli strumenti

2. Il valore indicato è basato sulla forza del segnale ricevuto più recente da qualsiasi altro nodo wireless e non rappresenta la potenza del segnale tra il Portal e il nodo.
3. È possibile che al momento del *ping*, il campo del *Link Quality* sia settato a 0, ciò può accadere perché non aveva ancora ricevuto eventuali messaggi radio da qualsiasi altro nodo. Questo non significa che l'unità sia difettosa, ma semplicemente non ha ancora fatto alcuna comunicazione radio. Questo spesso accade con il nodo che funge da "ponte" per il Portale, perché Portal può dialogare con questo nodo collegato direttamente, senza necessariamente generare del traffico radio.

L'ultima colonna mostra i parametri di configurazione del dispositivo. Il tipo di dispositivo è semplicemente un'etichetta che può essere applicata a un nodo, e diversamente dal nome del nodo, questa etichetta non deve essere unica.

Configurazione di un singolo nodo

La configurazione di un singolo nodo può essere fatta tramite la scheda *Node Info*. Tale scheda è già visibile sulla prima schermata di Portal, e selezionando il nodo interessato dalla lista (*Node Views*) vengono visualizzate tutte le informazioni di quel nodo specifico.

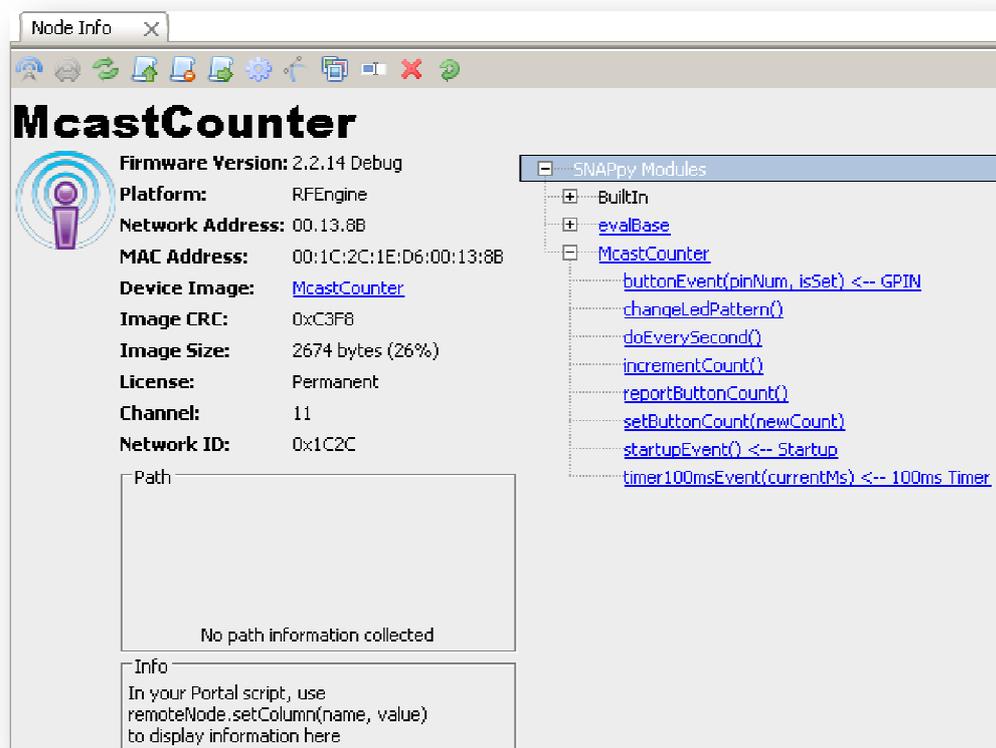


Figura 49 - Finestra dei dettagli di ogni nodo

La scheda indica le seguenti informazioni per ogni nodo:

- "Nome" del nodo
- La versione del *firmware*
- Alcuni parametri di configurazione di rete
- *I'Image Device* e CRC
- Licenza (permanenti o Demo)
- Alcuni ulteriori parametri di configurazione di rete

Il lato destro, mostra una vista ad albero delle funzioni standard che possono essere chiamate. Ci sono quattro parametri di configurazione di rete: *Network Address*, *MAC Address*, canale e Identificatore di rete (ID). *SNAP MAC Address* è un indirizzo standard a 64-bit, univoco ed è assegnato al dispositivo dalla fabbrica; l'indirizzo di rete è formato da tre byte che corrispondono agli ultimi tre byte del *MAC Address*. *Channel*, dichiara il canale in uso per la trasmissione dei dati, dei 16 canali disponibili per la frequenza di 2.4 GHz. L'ID di rete può essere pensato come un "canale logico", esso è un numero esadecimale e deve essere uguale per ogni dispositivo presente nella rete, infatti i dispositivi in una rete di SNAP devono condividere sia lo stesso canale sia lo stesso ID di rete per poter comunicare. Questo consente a più reti di condividere lo stesso canale, se necessario, anche se è preferito collocare reti indipendenti su distinti canali fisici per ridurre le probabilità di collisioni.

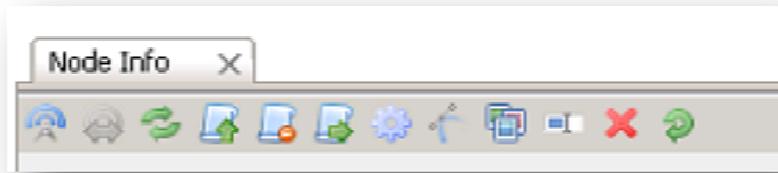


Figura 50 - Particolare della barra dei comandi della finestra "Node Info"

La barra degli strumenti, della finestra “ *Node Info*” ci permette di configurare il nostro nodo una volta selezionato e partendo dalla sinistra dell’immagine sopra citata abbiamo i seguenti comandi :

- *Ping* - sonda la rete in cerca di un’unità per la connettività
- *Traceroute* - serve a raccogliere informazioni sul percorso di rete
- *Refresh* - sonda un'unità per i valori indicati nella scheda Nodo Info
- *Upload Snappy Image* - con questo comando, riusciamo a caricare nel nodo un nuovo script
- *Erase Snappy Image* - rimuove lo script precedentemente caricato
- *Export SNAPpy Image* - esporta un'immagine su file
- *Change Configuration Parameters* - ci permette di settare alcuni parametri
- *Intercept STDOUT* - devia l'output dello script del nodo
- *Change Icon* - sostituisce un file alternativo (PNG) come icona del nodo
- *Rename Node* - cambia il nome del nodo
- *Remove Node* - rimuove il nodo dalla rete
- *Reboot Node* - riavvia il nodo

Ping

Premendo il tasto del *ping*, ordiniamo al portale di fare un rapido test di connettività al nodo corrente. Possiamo controllare nel Registro Eventi per avere la conferma dell'attività del *ping*. Il comando *Ping Broadcast* (visto da tutti i nodi) è disponibile anche nella barra degli strumenti del Portal, ed esattamente qui nell'immagine sottostante indicato dalla freccia.

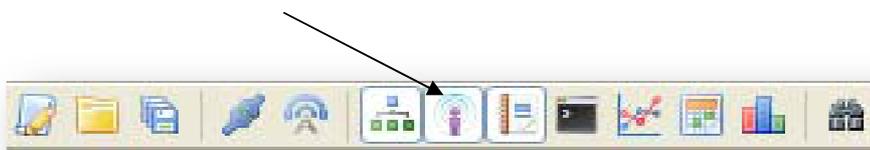


Figura 51 - Ping Broadcast

Traceroute

Questa azione invia un pacchetto di “*traceroute*” al nodo corrente. Un pacchetto *traceroute* raccoglie gli indirizzi di rete e la qualità della connessione di ciascun nodo che attraversa per raggiungere la sua destinazione. Ad esempio, un *traceroute* da portale verso un altro nodo, 00.13.8B, potrebbe prendere il seguente percorso: 00.00.0B a 00.2F.A1 a 00.13.8B indietro 00.2F.A1 e, infine, torna alla 00.00.0B. Questo percorso dovrebbe essere visualizzato nel riquadro *Path* (figura 52).

Path		
Portal - 00.00.0B	↓	↑
Bridge - 00.2F.A1	↓	97%
McastCounter - 00.13.8B	96%	↑

Total Round Trip Time: 8ms

Figura 52 – Path

Le frecce rappresentano un salto nel percorso in cui non esistono informazioni di qualità del collegamento necessarie per essere raccolte. Il collegamento tra Portal, 00.00.0B, e il nodo di Bridge 00.2F.A1, è una connessione seriale che non ha alcun legame. La qualità del collegamento del 96% indicato nell'esempio rappresenta la qualità del collegamento, come viene ricevuta dal nodo 00.13.8B quando il nodo 00.2F.A1 ha inviato i dati. Infine, il *Round Trip Time* è il tempo necessario per il Bridge per inviare la richiesta di *traceroute* e ricevere la risposta indietro. Si potrebbe notare che il tempo varia per le richieste di *traceroute* su uno stesso nodo; ciò è dovuto a fattori esterni quali l'individuazione del percorso di rete, o l'attesa di un altro nodo per finire la trasmissione.

Upload Snappy Image

Tramite questa azione, apparire una finestra di dialogo che permette di scegliere quali script caricare nell'unità selezionata.

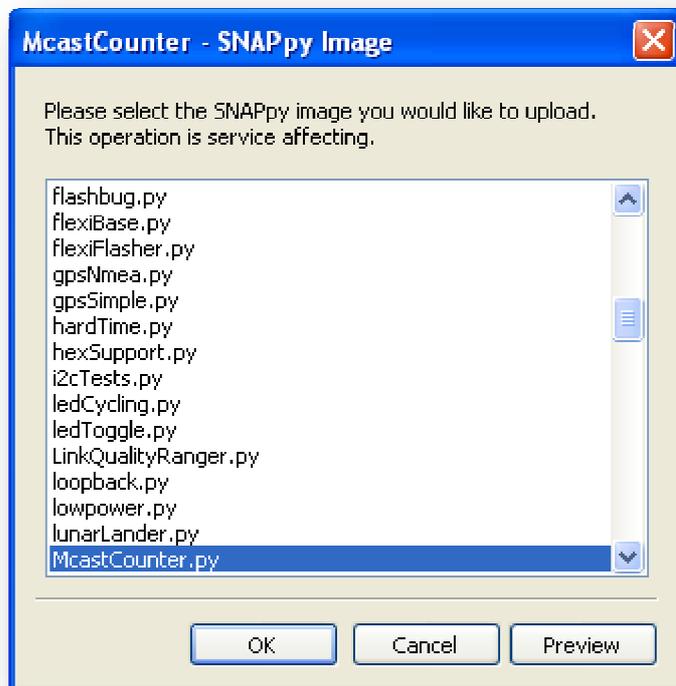


Figura 53 - Elenco di programmi da caricare nel nodo selezionato

Un nodo può contenere solo uno script alla volta, ma lo script può essere costituito da diversi file, utilizzando la funzione *"import"* in Python caricando così funzioni differenti.

```

McastCounter.py
-----
Sample SNAPpy file for Evaluation Kit Boards (Bridge, End Device, Pr
  Press the select-switch on any device to increment counter on all
-----
# Use Synapse Evaluation Board definitions
from synapse.evalBase import *

secondCounter = 0
buttonCount = 0

def startupEvent():
    """This is hooked into the HOOK_STARTUP event"""
    print "Begin startup"
  
```

Figura 54 – Esempio di script selezionato

Una volta che lo script è stato scelto, il Portal dirà al nodo di cancellare qualsiasi script precedente e quindi caricare il nuovo script per il nodo.

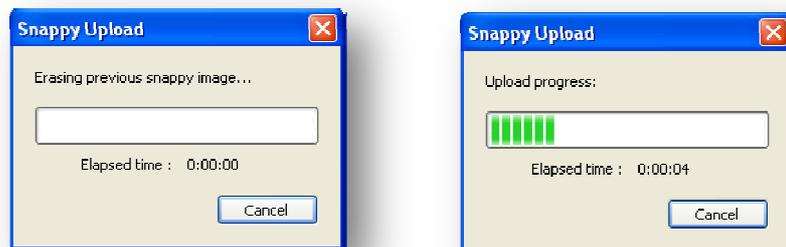


Figura 55 - Cancellazione dell'immagine del nodo e caricamento di un nuovo script

Una volta che l'intero script è stato caricato nel nodo, il Portal riavvia automaticamente il nodo, il quale esegue qualsiasi script contenuto all'interno dell'evento HOOK_STARTUP.

Erase Snappy Image

Cliccando su questa azione si cancellano le funzioni e gli script attualmente caricati (e funzionante) sul nodo selezionato. Non è necessario cancellare manualmente uno script precedente per caricarne uno nuovo, lo stesso verrà sovrascritto.

Change Configuration

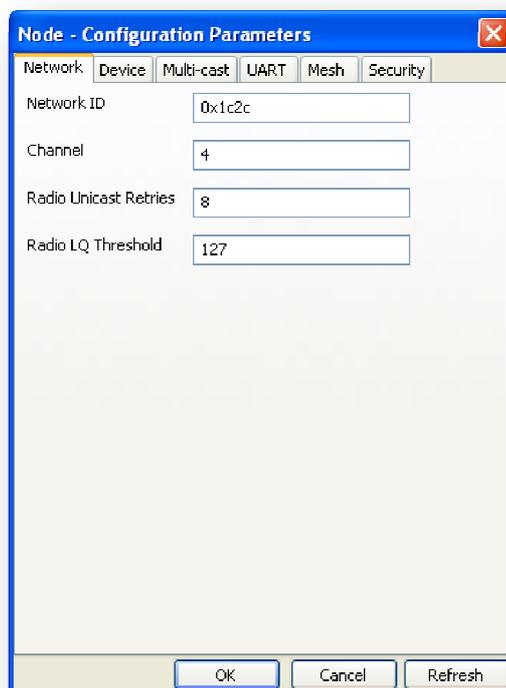


Figura 56 - Finestra per il cambio di configurazioni

Da questa finestra di dialogo, è possibile visualizzare e modificare molti parametri di configurazione del nodo stesso, come gli elenchi dei parametri relativi alle operazioni radio in generale, le liste dei parametri relativi all'identità del nodo e di alcune caratteristiche hardware, le liste di parametri relativi alla comunicazione in *multicast*, le liste dei parametri relativi al *buffering* dei dati ricevuti da uno dei due sistemi UART, ed i parametri relativi alla funzionalità di *routing* delle maglie del nodo.

Bibliografia

1. **O'Reilly**. Rfid Essentials. 2004.
2. **Dr. Paul Sanghera**. RFID+ Study Guide and Practice Exam. 2007.
3. **Python** . <http://www.python.it/>
4. **Synapse**. Portal Reference Manual.2009.
5. **Synapse**.Snap Reference Manual-v2.1.2007-2008.
6. **Synapse**.Snap Reference Manual-v2.2.2009.
7. **Paolo Talone, Giuseppe Russo**. Tecnologia Rfid .2008
8. **I Quaderni**. Centro Nazionale per l'Informatica nella Pubblica Amministrazione.2007
9. **Musiari, F**. RFID Introduzione alla tecnologia delle etichette intelligenti.2005
10. **La Tecnologia RFID**. <http://www.comefunziona.net/arg/rfid>
11. **Building an active RFID people** . <http://www.ns-tech.co.uk/>
12. **Un'introduzione alla tecnologia**.
<http://ondamultimediale.blogosfere.it/2008/09/rfid>.
13. **Paolo Talone, Giuseppe Russo** . Fondamenti di una tecnologia silenziosa pervasiva.2008

