

**Università degli Studi di Padova**

**Facoltà di Ingegneria**



Corso di Laurea Triennale di Ingegneria Biomedica

# **Sviluppo di un codice di trasmissione wireless per attuatori a pedale di un dispositivo medico**

**Relatore:**

Ch.mo Prof. Alfredo Ruggeri

**Studente:**

Catello Chierchia

Matricola: 562745

**Correlatori:**

Ing. Gianantonio Pozzato

Gianmarco Moresco

Anno Accademico 2011/2012



## INDICE DEGLI ARGOMENTI

Introduzione .....	4
Capitolo 1.....	8
1.1 Bisturi a Risonanza Quantica Molecolare Vesalius® .....	8
Capitolo 2.....	12
2.1 Standard IEEE 802.15.4.....	12
2.2 Definizione di layer .....	14
2.3 Trasferimento dei dati .....	15
2.3.1 Trasferimento dei dati al coordinatore.....	15
2.3.2 Ricezione dei dati al coordinatore .....	15
2.3.3 Peer to Peer .....	16
2.4 Robustezza .....	16
2.5 Consumo di potenza.....	17
2.6 Specifiche a livello fisico.....	18
2.6.1 Specifiche nella banda a 2.4 GHz.....	19
2.7 Specifiche generali.....	19
2.8 Direttiva R&TTE 1995/5/CE.....	20
Capitolo 3.....	22
3.1 Analisi del Chipset commerciale: Xbee 802.15.4 OEM RF.....	22
3.1.1 Caratteristiche tecniche.....	22
3.2 Software X-CTU .....	25
Capitolo 4.....	28
4.1 Linguaggio di programmazione C .....	28
4.1.1 Vantaggi del linguaggio C .....	29
4.2 Elementi fondamentali di un programma in C.....	29

4.2.1 Variabili .....	29
4.2.2 Operatori .....	30
4.3 Controlli .....	31
4.3.1 Controlli condizionali .....	31
4.3.2 Controlli iterativi.....	32
4.3.3 Array .....	33
4.4 Funzioni e puntatori .....	33
4.4.1 Funzioni .....	33
4.4.2 Puntatori.....	33
Capitolo 5.....	34
5.1 Sviluppo del codice realizzato .....	34
5.1.1 CRC .....	35
5.1.2 Codice di Hamming .....	37
5.1.3 Costruzione del codice.....	37
5.1.4 Codice Tx/Rx.....	41
Capitolo 6.....	44
6.1 Conclusioni e prospettive future .....	44
Appendice A .....	46
Bibliografia .....	48

## Introduzione

Lo sviluppo delle tecnologie biomediche negli ultimi 40 anni ha generato una rivoluzione degli approcci diagnostici terapeutici in numerose discipline medico-chirurgiche come la cardiologia, ortopedia, oculistica o cardiocirurgia. In generale si può dire che il progredire dello sviluppo scientifico ha portato notevoli vantaggi all'uomo, migliorando sensibilmente il livello della qualità di vita [1].

Il binomio tecnologia–chirurgia, ha uno stretto legame che parte dall'Antico Egitto (nel papiro di Smith viene riportato l'uso di un bastoncino infuocato per cauterizzare un tumore della mammella) [2] e che arriva fino all'età moderna: basti pensare ad illustri fisici-fisiologici come Hermann von Helmholtz (1821 – 1894) ed al suo lavoro scientifico riguardo la produzione del calore degli esseri viventi, oppure ad Arsène d'Arsonval (1851 – 1940), che fu il primo ad utilizzare i campi elettromagnetici in medicina: nel corso dei suoi studi, d'Arsonval scoprì che una corrente elettrica alla frequenza di 10 KHz produceva calore nell'attraversamento dei tessuti, senza però portare alla contrazione muscolare che si avverte alle frequenze inferiori [2].

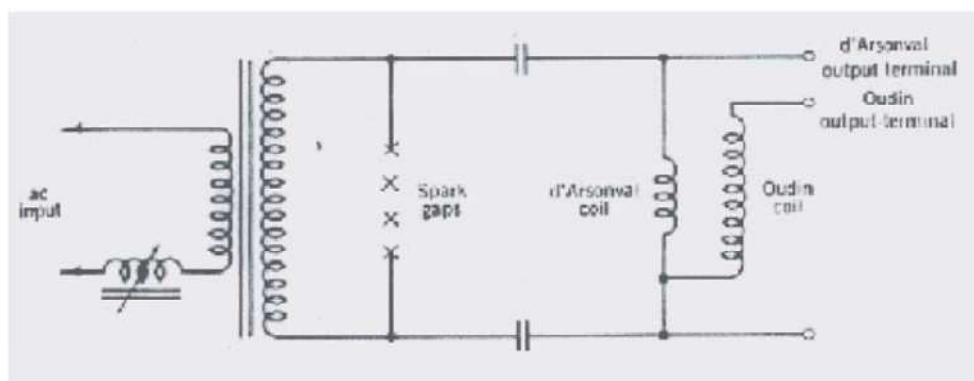


Figura 1 Circuito d'Arsonval

Con l'invenzione del triodo, nel 1907, Lee de Forest riuscì a generare correnti ad alta frequenza non smorzate e realizzò un dispositivo con il quale fece incisioni sul tessuto umano, usando una potenza superiore ai 70 W e una frequenza di circa

2 MHz; nel 1936 alcuni chirurghi iniziarono ad utilizzare questo tipo di apparecchiature per operazioni chirurgiche e nel 1970 si ebbe un'ulteriore modifica con la sostituzione dei tubi elettronici con dei dispositivi allo stato solido [2].

Arrivati ai giorni nostri, possiamo parlare di quattro tipi di strumenti chirurgici tradizionali a taglio:

- bisturi a lama;
- elettrobisturi;
- bisturi laser;
- bisturi a Risonanza Quantica Molecolare Vesalius®.

Il progetto realizzato si concentrerà su quest'ultimo tipo di bisturi, considerato all'avanguardia rispetto alle altre categorie in quanto consente di ottenere il taglio e il coagulo di un tessuto ad una temperatura inferiore ai 60 °C evitando così la necrosi delle cellule e quindi una notevole riduzione di dolore post-operatorio da parte dei pazienti [3].

In particolare si parlerà di un nuovo modello di bisturi a Risonanza Quantica Molecolare Vesalius®, ad oggi in fase di sviluppo e progettazione, che tra le sue caratteristiche offrirà la possibilità di utilizzare degli attuatori a pedale per le modalità di taglio e coagulo di un tessuto comandabili a distanza, senza l'ingombro dei cavi di collegamento.

Le apparecchiature attuali, prodotte da Telea Electronic Engineering Srl, sono dotate di pedali pneumatici collegati al dispositivo con cavi di lunghezza di circa quattro metri, che eliminano l'inconveniente di avere cavi elettrici in tensione sul pavimento della sala operatoria dove notoriamente c'è un diffuso utilizzo di liquidi che possono potenzialmente costituire una fonte di pericolo [rif. Manuale uso Vesalius®].

Inoltre un pedale pneumatico ad aria offre il vantaggio di essere intrinsecamente immune ai disturbi sia irradiati che condotti dalle diverse tipologie di dispositivi che si trovano in un ambiente come la sala operatoria.

Per contro, esiste una certa debolezza che nel corso degli anni si è manifestata nell'utilizzo di questa tecnologia, spesso legata ad un uso non corretto o non attento da parte degli utilizzatori finali che compromettono la tenuta stagna del tubo o addirittura il distacco di parti dovute a tensioni improvvise.

L'utilizzo di pedali wireless nasce quindi dall'esigenza di migliorare ed eliminare i problemi che nascono dall'uso dei pedali ad aria (otturazione o bucatura dei tubi, rottura delle boccole con conseguente perdita d'aria, sgancio accidentale dei tubi) e che possono risultare ingombranti in uno spazio limitato come una sala operatoria.

In questa tesi si cercherà inoltre di risolvere eventuali problematiche relative alla trasmissione come attivazioni involontarie dei pedali provocate dai disturbi del segnale, o la riduzione dei consumi della batteria in caso di lunga durata di intervento.

In particolare in questo lavoro di tesi verrà studiata e progettata la realizzazione di un collegamento wireless tramite il protocollo standard IEEE 802.15.4 ed utilizzando le schede radio Xbee 802.15.4 OEM RF che lavorano ad una frequenza di 2.4 GHz e rispettano i requisiti essenziali posti dalle direttive europee e dai regolamenti internazionali di riferimento per i dispositivi medici.

Per permettere la trasmissione e la ricezione dei dati tra queste due schede verrà implementato un codice realizzato con il linguaggio di programmazione C, inoltre, con lo stesso programma si scriveranno altri due codici che permetteranno la rilevazione e la correzione del pacchetto dati che viene inviato, in caso di errori dovuti ad interferenze tra l'invio e la ricezione del messaggio.





# Capitolo 1

## 1.1 Bisturi a Risonanza Quantica Molecolare Vesalius®

La maggior parte dell'energia che si fornisce ad un sistema si trasforma in calore, ovvero in energia cinetica molecolare: quando questa energia viene fornita ad un tessuto biologico, si ha un aumento della vibrazione molecolare (aumento dell'energia cinetica e potenziale) con conseguente innalzamento della temperatura che porta prima alla denaturazione della cellula e poi alla necrosi.

Poiché un sistema assorbe energia tramite un numero discreto di pacchetti, detti quanti, possiamo supporre che un legame molecolare assorba un'energia quantizzata il cui valore dipende dalla frequenza dell'onda che li trasporta, quindi:

- il valore dell'energia dei quanti dipende dalla frequenza della sorgente che li ha prodotti;
- se il valore dell'energia del quanto è diverso dall'energia di legame si ha solo un aumento dell'energia cinetica, senza ottenere la rottura del legame stesso (aumento della temperatura);
- se il valore dell'energia del quanto è uguale a quelle del legame, allora tutta l'energia viene assorbita e utilizzata per rompere il legame (senza l'aumento della temperatura).

Il bisturi a Risonanza Quantica Molecolare Vesalius® sfrutta proprio questo principio, permettendo di effettuare il taglio e il coagulo di tessuti molli senza riscaldare le cellule e quindi portarle alla necrosi. La grande particolarità che rende i bisturi Vesalius® ad essere una delle più innovative tecnologie nel mercato dei dispositivi medici sta appunto nel riuscire a realizzare un'operazione di taglio e coagulo a temperature molto basse (intorno ai 50°C) rispetto al centinaio di gradi raggiunti dai tradizionali elettrobisturi o bisturi laser, consentendo la sopravvivenza delle cellule

e quindi a un recupero più breve da parte del paziente nella fase post-operatoria [3].



Figura 2 Bisturi a Risonanza Quantica Molecolare Vesalius<sup>®</sup>, modello MC

Il Vesalius<sup>®</sup> genera corrente alternata con onde in alta frequenza così fissate: l'onda fondamentale è a 4 MHz, mentre le onde successive sono a 8, 12, 16 MHz con ampiezza decrescente e aumento di energia cinetica nelle cellule colpite, e quindi senza innalzamento della temperatura (50 °C – 65°C).

L'uso degli elettroni per trasferire i quanti di energia è dovuto al fatto che sono le particelle più piccole esistenti in natura e perché si è ipotizzato che il quanto energetico associato all'elettrone avesse un'energia maggiore del fotone alla stessa frequenza.

Il taglio è quindi provocato dalla rottura della struttura cellulare e non, come avviene negli elettrobisturi tradizionali, dal surriscaldamento delle cellule stesse (con temperature che superano le centinaia di gradi) e conseguente carbonizzazione che porta all'origine di guarigioni per seconda intenzione e a decorsi post-operatori molto dolorosi, oppure, come accade con i bisturi laser, dove

il valore energetico del quanto è diverso da quello dei legami molecolari interessati e quindi la temperatura risulta essere comunque molto elevata.

I principali vantaggi che il bisturi Vesalius® presenta sono:

- minor danno termico al tessuto biologico nella zona di interesse;
- guarigioni più rapide e meno traumatiche;
- riduzione della reazione infiammatoria e diminuzione del dolore post-operatorio;
- coagulazione a bassa tensione;
- taglio “ a freddo” nitido e preciso;
- selettività dell’ azione in caso di operazioni chirurgiche delicate quali:
  - o Neurochirurgia;
  - o Chirurgia spinale;
  - o Otorinolaringoiatria;
  - o Maxillo Facciale;
  - o Ginecologia;
  - o Ortopedia.



## Capitolo 2

### 2.1 Standard IEEE 802.15.4

L'IEEE 802 LAN/MAN Standards Committee è una commissione dell'IEEE (acronimo di Institute of Electrical and Electronic Engineers) preposta a sviluppare norme e regolamenti che disciplino le reti locali (LAN) e per le reti metropolitane (MAN) [sito comitato 802].

Lo Standard IEEE 802.15.4, gestito dal gruppo IEEE 802.15, sono una serie di norme approvate nel 2003 [4] che definiscono il protocollo di interconnessione tramite comunicazione radio tra diversi dispositivi rientranti in una PAN (Personal Area Network<sup>1</sup>). Tali dispositivi sono utilizzati per effettuare comunicazioni a brevi distanze, senza cavi di collegamento e situati in piccoli ambienti o infrastrutture, di conseguenza devono essere dispositivi relativamente economici ed utilizzabili per un'ampia gamma di applicazioni [4].

Lo Standard definisce in particolare le specifiche del livello fisico (PHY) e del MAC (Media Access Control<sup>2</sup>) per garantire una connessione wireless tra i dispositivi che richiedono un basso consumo di potenza e che lavorano ad una distanza dell'ordine di una decina di metri, tipicamente con un raggio di 15 metri.

Le specifiche ZigBee, WirelessHART, e MiWi (descritte nel dettaglio in seguito) sono basate su questo standard: esse sviluppano i livelli superiori del modello ISO/OSI<sup>3</sup>, non coperti dallo standard, per offrire una soluzione completa di rete di trasmissione dati.

La velocità di trasmissione è regolata in modo tale da poter consentire la connettività di periferiche largamente diffuse, (una fra tutte il Personal Computer)

---

<sup>1</sup> In telecomunicazioni una rete personale o Personal Area Network è una rete informatica utilizzata per permettere la comunicazione tra diversi dispositivi vicini a un singolo utente.

<sup>2</sup> In telecomunicazioni nell'ambito delle reti di calcolatori MAC (acronimo di Media Access Control) è il nome di uno strato del modello architetturale standardizzato ISO-OSI, definito nello standard IEEE 802 e che contiene funzionalità di controllo dell'accesso al mezzo fisico per canali broadcast, funzionalità di framing e controllo di errore.

<sup>3</sup> Il modello ISO/OSI, concepito per reti di telecomunicazioni a commutazione di pacchetto, è costituito da una pila (stack) di protocolli attraverso i quali viene ridotta la complessità implementativa di un sistema di comunicazione per il networking.

ma anche a sensori ed a sistemi di controllo che generalmente richiedono un basso baud rate<sup>4</sup>.

Si ottiene quindi una rete di comunicazione semplice e di basso costo, definita **LR-WPAN** (Low Rate Wireless PAN) con le seguenti caratteristiche:

- ✓ Data rate di 250 kb/s, 40 kb/s e 20 kb/s
- ✓ Operabilità in configurazione a stella o “peer-to-peer”
- ✓ 16 o 64 bit d’indirizzo allocati
- ✓ Accesso al canale in modalità CSMA-CA
- ✓ Completa definizione del protocollo per il trasferimento dei dati
- ✓ Basso consumo di potenza
- ✓ Indicazione della qualità del canale
- ✓ 16 canali nella banda attorno a 2.4 GHz, 10 canali nella banda attorno a 915MHz, un canale a 868MHz

Una LR-WPAN può includere due tipologie distinte di dispositivi: chiamati FFD (Full Function Device) e RFD (Reduced Function Device).

I FFD possono operare fungendo da coordinatori della rete (PAN Coordinator), interagendo come coordinatori semplici o da terminali di comunicazione con qualsiasi altro nodo in visibilità radio, a prescindere dalla modalità di funzionamento di quest'ultimo quindi dialogare anche con i RFD che, al contrario, sono stati concepiti come nodi a ridotta complessità, non possono comunicare tra di loro e vengono utilizzati per applicazioni molto semplici dove è necessario inviare poche quantità di byte.

La struttura geometrica di una LR-WPAN può essere configurata in due diverse topologie<sup>5</sup>: a stella o a “peer to peer”. Nella topologia a stella, concepita a supporto di contesti operativi semplici, ciascun dispositivo può comunicare esclusivamente con il PAN Coordinator, il quale gestisce tutte le comunicazioni all’interno dell’area; generalmente esso ha un’alimentazione fissa mentre gli altri dispositivi sono alimentati tramite una batteria.

In una topologia peer to peer invece, ciascun dispositivo può comunicare con un altro all’interno della rete, a patto che non debba ricorrere alla mediazione del

---

<sup>4</sup> Rappresenta il numero di simboli che viene trasmesso in un secondo.

<sup>5</sup> In telecomunicazioni una topologia di rete è la rappresentazione della struttura geometrica di una rete di telecomunicazioni che fa parte della più generale architettura di rete.

coordinatore (comunicazione multi-hop): è tipicamente rivolta ad applicazioni decisamente più complesse come ad esempio il controllo e il monitoraggio industriale. In una rete a stella ogni dispositivo FFD può creare una sottorete diventandone il coordinatore e tutte le reti operano indipendentemente dalle altre, in quanto sono contraddistinte da una propria PAN che evita conflitti con altre reti, ed è il coordinatore stesso ad ammettere o meno ulteriori dispositivi FFD o RFD; in altre parole continua ad esistere il PAN Coordinator ma è consentito anche agli altri FFD di operare come coordinatori secondari, fornendo funzionalità di accesso alla rete e di sincronizzazione agli altri dispositivi interconnessi.

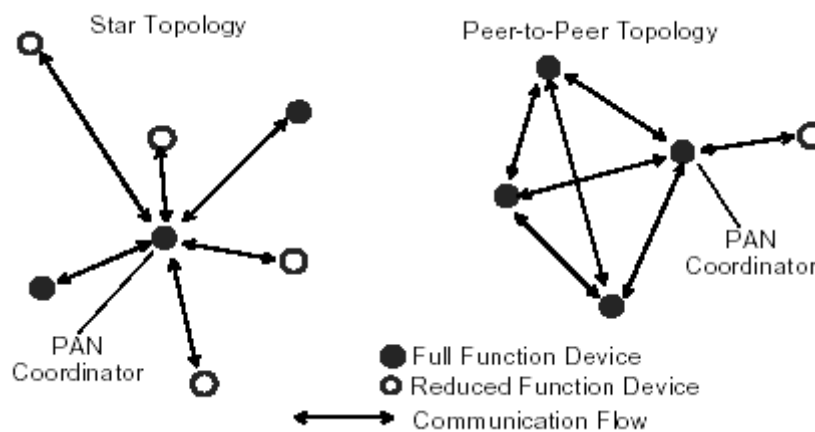


Figura 3 Topologie di rete previste dallo standard IEEE 802.15.4

## 2.2 Definizione di layer

L'architettura LR-WPAN è definita attraverso una struttura a strati (layer) ed ogni strato è responsabile di una parte della standard e offre servizi agli strati superiori. A livello fisico si ha l'attivazione o disattivazione del ricetrasmittitore, la selezione e il controllo del canale, la trasmissione e ricezione dei pacchetti di informazione attraverso il mezzo fisico.

A livello MAC invece viene garantito l'accesso al mezzo fisico e in particolare vengono generati segnali di sincronizzazione (appositi pacchetti di segnalazione chiamati beacons) se il dispositivo è il coordinatore.

## 2.3 Trasferimento dei dati

Per quanto riguarda, invece, le modalità secondo cui i nodi della WPAN si scambiano reciprocamente i messaggi, ovvero il trasferimento dei dati, si devono considerare tre distinte possibili interazioni:

- il dispositivo trasmette dati al coordinatore di rete,
- il dispositivo riceve i dati dal coordinatore di rete,
- la comunicazione “peer to peer”.

### 2.3.1 Trasferimento dei dati al coordinatore

Se la rete supporta i beacon, esso attende prima il network beacon e quando viene identificato, il dispositivo si sincronizza con il “super frame” (frame utilizzato per l’invio dei dati) ed invia il messaggio, il coordinatore deve poi comunicare al dispositivo la ricezione di dati mediante “acknowledgment” (frame di ammissione) con cui completa la transizione. Se la rete non supporta il beacon, il dispositivo trasmette i dati appena può, poiché non c’è sincronizzazione [4].

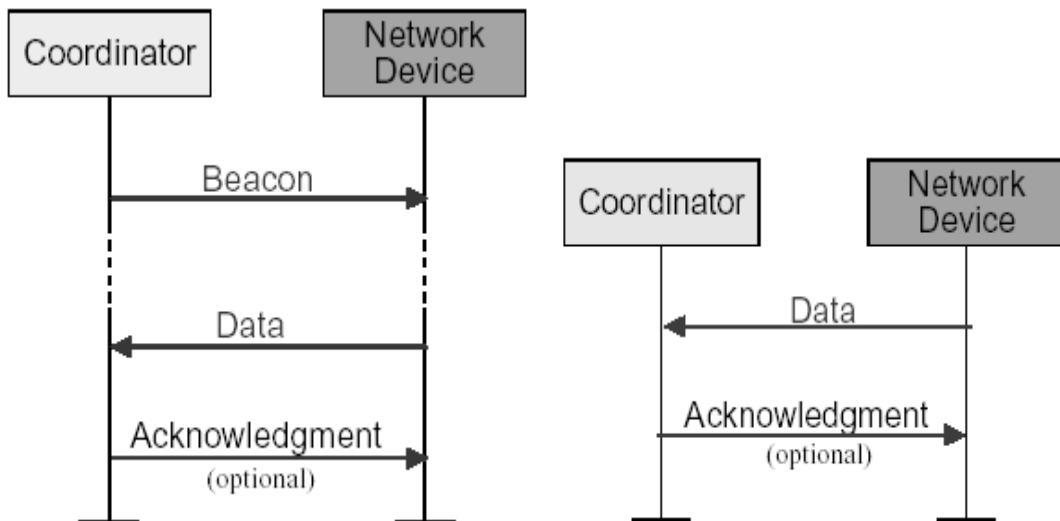


Figura 4 Protocollo per il trasferimento dei dati al coordinatore in modalità “beacon” e no

### 2.3.2 Ricezione dei dati al coordinatore

In una rete che supporta i beacon, viene segnalato dal coordinatore la presenza di un messaggio che viene ascoltato e di conseguenza il coordinatore manda un segnale di avvenuta ricezione mediante il frame di “acknowledgment”, il dispositivo



manda un ulteriore messaggio di conferma di avvenuta ricezione, in modo tale da eliminare il messaggio dalla lista dei dati in attesa. Se la rete non supporta i beacon, il coordinatore mette da parte i dati per uno specifico dispositivo, attendendo il data request.

Se non vi sono messaggi pendenti, il coordinatore trasmette un data frame di lunghezza nulla ad indicare l'assenza di messaggi e si completa la transizione con il frame di acknowledgment del coordinatore [4].

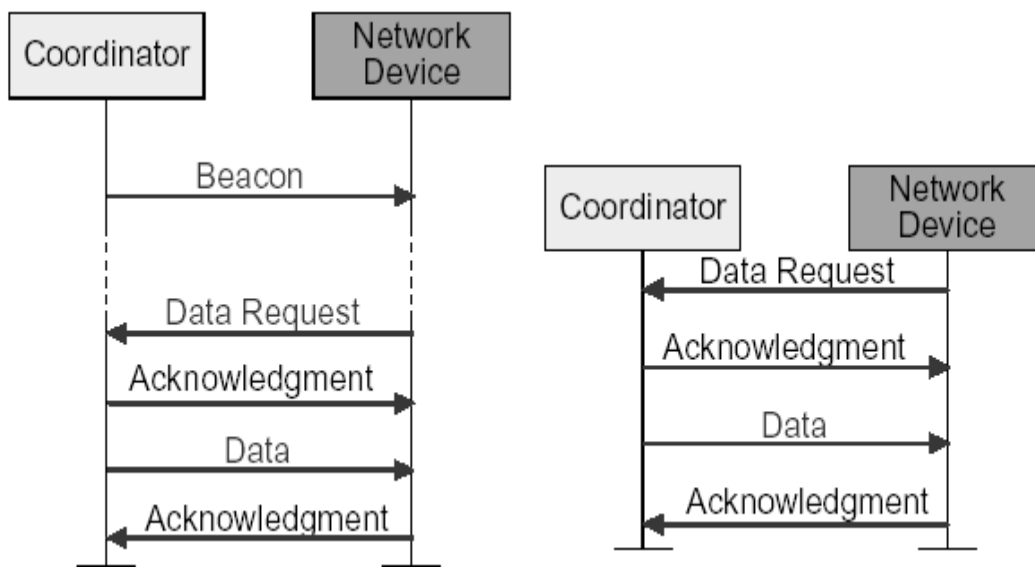


Figura 5 Modalità di trasferimento dei dati dal coordinatore in entrambe le direzioni

### 2.3.3 Peer to Peer

In questo caso il dispositivo deve essere in grado di comunicare con tutti gli altri dispositivi presenti nell'area interessata: per fare ciò gli apparecchi possono sempre rimanere in ascolto oppure sincronizzarsi tra loro.

## 2.4 Robustezza

La robustezza dei dati consiste nella loro capacità di comportamento in situazioni impreviste come possono essere ad esempio gli errori o le eccezioni di varia natura.

Nelle diverse tecniche che assicurino una buona robustezza dei dati, le principali sono:

- **CSMA-CA** (Carrier Sense Medium ACCESS with Collision Avoidance): è un protocollo di accesso multiplo usato nelle reti locali senza fili come

evoluzione del protocollo CSMA con ulteriori accorgimenti per ridurre le collisioni che devono essere il più possibile evitate dato che i sistemi wireless non hanno modo di rilevarle (da cui appunto il nome Collision Avoidance). Per ciò ogni volta che un dispositivo vuole trasmettere, esso identifica il limite dello slot corrente (tempo che impiega un dato a compiere un tragitto tra due nodi) per poi attendere un numero casuale di slot. Se il canale è occupato, attende un altro numero di slot prima di accedere di nuovo al canale, altrimenti la trasmissione avviene dalla slot successiva.

- **Frame acknowledgment (ACK)** : è un messaggio inviato dopo che è avvenuta la ricezione di un'informazione completa: se il dispositivo che ha inviato il messaggio non riceve l'acknowledgment di risposta, vuol dire che la trasmissione non è riuscita e si provvede ad un nuovo invio. Se dopo un certo numero di tentativi il messaggio non è stato ancora ricevuto, il dispositivo può scegliere se riprovare ancora o no. Se l'acknowledgment non è previsto, il dispositivo presume che la trasmissione sia avvenuta appena invia il messaggio.
- **FCS (Frame Check Sequence)** è una struttura di quattro byte all'interno di un frame che contiene un valore CRC<sup>6</sup> per il riconoscimento di eventuali errori di ricezione. Il campo FCS non permette di correggere gli errori, ma solo di rilevarli.

## 2.5 Consumo di potenza

Uno degli obiettivi dello standard riguarda il consumo di potenza: visto che nella maggior parte dei sistemi i dispositivi sono alimentati a batteria, il protocollo è stato studiato per dispositivi portatili, con bassi cicli di lavoro, per ridurre il consumo (i dispositivi rimarranno in modalità "sleep" e ascolteranno il canale in caso di messaggi pendenti solo a determinati intervalli di tempo).

Lo standard IEEE 802.15.4 è stato concepito a supporto di reti wireless a corto raggio (WPAN), basso rate e ridotti consumi energetici. Questi requisiti non possono

---

<sup>6</sup> Il cyclic redundancy check (ovvero Controllo a ridondanza ciclica, il cui acronimo CRC è invece ben più diffuso) è un metodo per il calcolo di una sequenza di bit che viene utilizzata per verificare l'integrità di un dato o di un messaggio che può subire alterazioni.

essere soddisfatti dall'algoritmo CSMA/CA implementato nello standard IEEE 802.11, poiché quest'ultimo non è stato ottimizzato per gli stessi obiettivi. Infatti, nelle reti WiFi, le stazioni interessate ad accedere al canale devono eseguire l'operazione di Carrier Sensing durante l'intervallo di backoff (oltre che nel DIFS) prima di poter trasmettere. Questo comporta un elevato dispendio di energia, visto che le operazioni di sensing causano elevati consumi, paragonabili a quelli che si ottengono in trasmissione. Pertanto, nello standard IEEE 802.15.4, non è prevista alcuna operazione di sensing del canale durante l'intervallo backoff, lasciando che, al suo scadere, i nodi interessati a trasmettere eseguano il carrier sensing per un breve intervallo di tempo pari a due time-slot (procedura di Clear Channel Assessment, CCA).

## 2.6 Specifiche a livello fisico

Un dispositivo che supporta questo standard deve rispettare le seguenti bande ed attenersi ai seguenti dati:

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868/915	868–868.6	300	BPSK	20	20	Binary
	902–928	600	BPSK	40	40	Binary
2450	2400–2483.5	2000	O-QPSK	250	62.5	16-ary Orthogonal

Figura 6 Specifiche a livello fisico nelle diverse bande frequenziali previste

La lunghezza massima del frame a livello fisico è di 127 byte suddivisi in questo modo:

Octets: 4	1	1		variable
Preamble	SFD	Frame length (7 bits)	Reserved (1 bit)	PSDU
SHR		PHR		PHY payload

Figura 7 Suddivisione dei frame a livello fisico

### 2.6.1 Specifiche nella banda a 2.4 GHz

A queste frequenze viene effettuata una modulazione O-QPSK<sup>7</sup> (Offset Quadrature Phase Shift Keying) sulla codifica dei simboli: 4 bit rappresentano il simbolo (16 possibili valori) ed a ciascuno di questi viene associato uno delle 16 possibili sequenze di bit che modulano il segnale portante.

Nella classica modulazione QPSK segnali I e Q possono variare nello stesso istante per passare da un simbolo al successivo, il che fa sì che il vettore della modulazione possa attraversare lo zero qualora tra un simbolo e il successivo la fase vari di 180°, obbligando a utilizzare amplificatori RF lineari e precisi.

Al contrario, nella modulazione OQPSK, i momenti di transizione dei segnali I e Q sono sfalsati di mezzo periodo di simbolo, per cui ogni transizione di fase tra un simbolo e il successivo può essere al massimo di 90 gradi, il che permette di avere un involuppo della portante pressoché costante e, quindi, la possibilità di usare amplificatori RF molto più semplici.

Ogni chip sequences si compone di 32 bit, il che vuol dire che da 4 bit di informazione si passa a 32 bit.

Le sequenze associate a ciascun simbolo sono correlate attraverso spostamenti o coniugazioni cicliche dei bit [4].

## 2.7 Specifiche generali

La tolleranza sulle frequenze generali trasmesse deve essere al massimo 40ppm e la potenza trasmessa deve essere al minimo -3dBm. La trasmissione sui bassi livelli di potenza è fatta in modo tale da ridurre al minimo l'interferenza recata ad altri dispositivi che si trovano nelle vicinanze, mentre per i segnali ad alta potenza il limite è fissato dai regolamenti locali.

Il livello tipico di potenza trasmessa è di 1 mW (0 dBm), in ricezione il massimo livello di potenza del segnale invece è di -20 dBm.

---

<sup>7</sup> Si tratta di un tipo di modulazione digitale dal classico sistema QPSK, ma che richiede meno potenza per la trasmissione a parità di velocità ottenibile.

## **2.8 Direttiva R&TTE 1995/5/CE**

Le soluzioni di seguito discusse e in generale i dispositivi wireless che verranno utilizzati rientrano nella direttiva R&TTE 1995/5/CE [5], riguardante le apparecchiature radio e il reciproco riconoscimento della loro conformità, in particolare devono essere rispettati i seguenti articoli:

Art. 3.1a L'apparecchio deve salvaguardare la protezione della salute e della sicurezza dell'utente o di qualsiasi persona, compresi gli obiettivi per quanto riguarda i requisiti di sicurezza previsti dalla direttiva 73/23/CEE (riguardo la legislazione al materiale elettrico destinato ad essere adoperato) [6], ma senza l'applicazione dei limiti minimi di tensione.

Art. 3.1b I requisiti in materia di protezione per quanto riguarda la compatibilità elettromagnetica prevista dalla direttiva 89/336/CEE [7] (per il ravvicinamento delle legislazioni degli Stati membri, relativi alla compatibilità elettromagnetica).

Art. 3.2 Le apparecchiature radio sono costituite in modo da utilizzare efficacemente lo spettro attribuito alle radiocomunicazioni terrestri/spaziali, evitando interferenze dannose.

Per la sicurezza si può considerare valida l'approvazione ottenuta con lo standard EN 60601-1, mentre per quanto riguarda la compatibilità elettromagnetica si deve prendere nota delle norme ETSI EN 301489-3 [5].

Per quanto riguarda i requisiti radio, bisogna disporre di un'approvazione da parte di organismo notificato (ad esempio la marcatura CExxxx), oppure di moduli radio impostabili in frequenza (limite inferiore, centro banda, limite superiore) e la potenza massima.

Visto il modesto valore di potenza utilizzato, per la valutazione dell'esposizione umana, è sufficiente calcolare il limite di distanza minima di utilizzo e verificare se è compatibile con la posizione del dispositivo.



## Capitolo 3

### 3.1 Analisi del Chipset commerciale: Xbee 802.15.4 OEM RF

Gli Xbee802.15.4 OEM RF sono dei moduli radiocomandati realizzati secondo le norme dello Standard IEEE 802.15.4 e sono particolarmente efficaci per la comunicazione a raggio breve e per il basso consumo.

Si tratta di una soluzione compatibile con lo standard ZigBee/IEEE 802.15.4 che soddisfa la necessità di una rete a basso costo e a basso consumo, pensata soprattutto l'uso in applicazioni embedded che richiedano un basso transfer rate e bassi consumi tra i quali possono rientrare appunto gli attuatori a pedale oggetto del presente elaborato.

Uno dei vantaggi principali che offrono i moduli Xbee è la comunicazione punto-punto o punto-multi punto, con tempi relativamente brevi; apposite interfacce Xbee USB consentono di collegare i suddetti moduli al PC e la comunicazione viene gestita attraverso una comune comunicazione seriale.

Il modulo Xbee opera sulla banda non licenziata ISM (Industrial, Scientific and Medical) alla frequenza di 2.4 GHz, 915 MHz e 868 MHz e non serve nessuna configurazione iniziale [8] nella Nella banda 2,4 GHz ci sono 16 canali ZigBee, da 5 MHz ciascuno.

#### 3.1.1 Caratteristiche tecniche

- frequenza operativa: 2.4 GHz;
- potenza a RF: 1 mW;
- potenza massima: 2 mW;
- range di temperatura da -40 °C a 85 °C;
- sensibilità in ricezione: -96 dBm;
- indoor range: 30 m
- outdoor range: 100 m;



Figura 8 Scheda Xbee

## Vantaggi

- bidirezionalità: i moduli sono capaci sia di trasmettere che di ricevere dati in modo tale da poter sempre sapere se i dati sono stati ricevuti o meno;
- indirizzamento dei moduli: ogni Xbee ha un numero seriale unico, in pratica 2 moduli Xbee possono anche comunicare solo tra di loro senza coinvolgere altre schede;
- implementazione: nei moduli Xbee sono già implementati tutti i controlli tipici di una trasmissione wireless (error checking)
- minimizzare le interferenze: in caso di interferenze o disturbi che non permettono la corretta trasmissione del segnale, i moduli si spostano automaticamente in un nuovo canale (16 in tutto) per permettere il trasferimento dei dati.

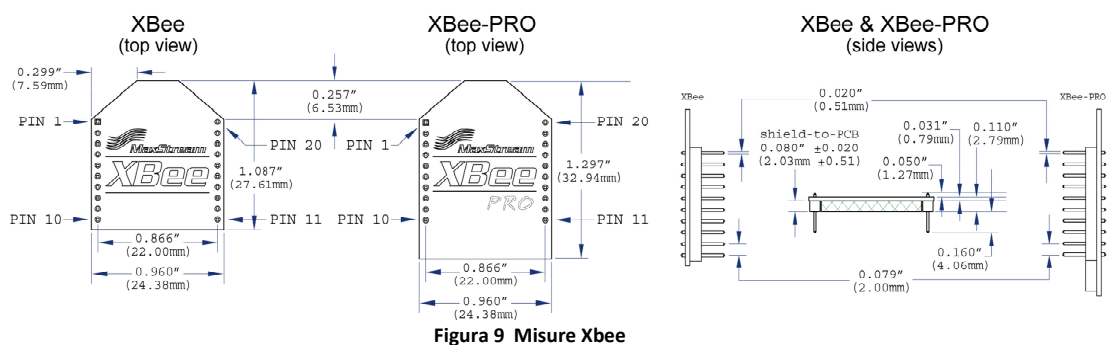


Figura 9 Misure Xbee



Platform	XBee® 802.15.4 (Series 1)	XBee-PRO® 802.15.4 (Series 1)	XBee-PRO® XSC
<b>Performance</b>			
RF Data Rate	250 kbps	250 kbps	10 kbps / 9.6 kbps
Indor/Urban Range	100 ft (30 m)	300 ft (100 m)	Up to 1200 ft (370 m)
Outdoor/RF Line-of-Sight Range	300 ft (100 m)	1 mi (1.6 km)	Up to 6 mi (9.6 km)
Transmit Power	1 mW (+0 dBm)	60 mW (+18 dBm)*	100 mW (+20 dBm)
Receiver Sensitivity (1% PER)	-92 dBm	-100 dBm	-106 dBm
<b>Features</b>			
Serial Data Interface	3.3V CMOS UART	3.3V CMOS UART	3.3V CMOS UART (5V Tolerant)
Configuration Method	API or AT Commands, local or over-the-air	API or AT Commands, local or over-the-air	AT Commands
Frequency Band	2.4 GHz	2.4 GHz	902 MHz to 928 MHz
Interference Immunity	DSSS (Direct Sequence Spread Spectrum)	DSSS (Direct Sequence Spread Spectrum)	FHSS (Frequency Hopping Spread Spectrum)
Serial Data Rate	1200 bps - 250 kbps	1200 bps - 250 kbps	1200 bps - 57.6 kbps
ADC Inputs	(6) 10-bit ADC inputs	(6) 10-bit ADC inputs	None
Digital I/O	8	8	None
Antenna Options	Chip, Wire Whip, U.FL, & RPSMA	Chip, Wire Whip, U.FL, & RPSMA	Wire Whip, U.FL, RPSMA
<b>Networking &amp; Security</b>			
Encryption	128-bit AES	128-bit AES	No
Reliable Packet Delivery	Retries/Acknowledgments	Retries/Acknowledgments	Retries/Acknowledgements
IDs and Channels	PAN ID, 64-bit IEEE MAC, 16 Channels	PAN ID, 64-bit IEEE MAC, 12 Channels	PAN ID, 32-bit Address, 7 Channels
<b>Power Requirements</b>			
Supply Voltage	2.8 - 3.4VDC	2.8 - 3.4VDC	3.0 - 3.6VDC
Transmit Current	45 mA @ 3.3VDC	215 mA @ 3.3VDC	265 mA typical
Receive Current	50 mA @ 3.3VDC	55 mA @ 3.3VDC	65 mA typical
Power-Down Current	<10 uA @ 25° C	<10 uA @ 25° C	45 uA pin Sleep
<b>Regulatory Approvals</b>			
FCC (USA)	OUR-XBEE	OUR-XBEEPRO	MCQ-XBEE XSC
IC (Canada)	4214A-XBEE	4214A-XBEEPRO	1846A-XBEE XSC
ETSI (Europe)	Yes	Yes* Max TX 10 mW	No
C-TICK Australia	Yes	Yes	No
Telec (Japan)	Yes	Yes*	No

Figura 10 Scheda tecnica per Xbee e Xbee Pro

## 3.2 Software X-CTU

X-CTU è un programma della Digi<sup>8</sup> per il sistema operativo Windows; questo programma serve per interagire con i prodotti della Digi (come le schede Xbee ad esempio) e per fornire un'interfaccia grafica semplice da usare.

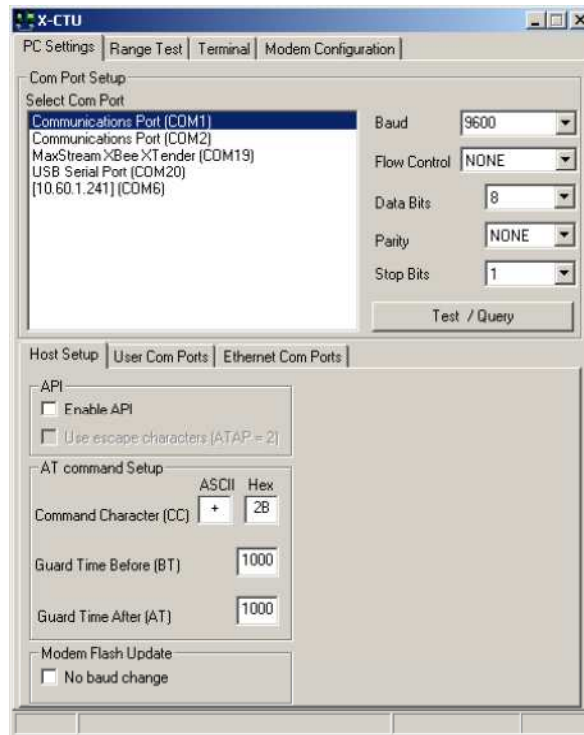


Figura 11 Scheda PC Setting

All'avvio del programma troveremo quattro schede, ognuna con diverse funzioni:

**pc setting:** consente di selezionare la porta COM del computer e di configurarla per i vari settaggi, tra cui la velocità di trasmissione (baud rate), controllo del flusso (flow control), data bits, parità, stop bit. Nella parte bassa della finestra troviamo altre 3 schede che rappresentano l' utilizzo della modalità API, oppure l' aggiunta o eliminazione di una porta COM creata. Infine il tasto Test/Query è utilizzato per verificare che la porta COM sia stata selezionata e che le modifiche siano state apportate;

---

<sup>8</sup> Digi International è leader mondiale nei dispositivi networking di tipo commerciale e nei sistemi wireless machine-to-machine (prodotti M2M).

**range test:** serve a testare i moduli Xbee, indicando la qualità del segnale (RSSI), la ricezione o meno del segnale, e la scelta dei tempi per l'invio dei dati. Il test può essere utilizzato anche in modalità API;

**terminal:** la scheda terminal contiene 3 funzioni principali, ovvero emulazione del terminale, capacità di mandare e ricevere pacchetti di dati predefiniti (Assemble Packet) e capacità di mandare e ricevere dati in formato Hex (esadecimale) e ASCII (Show/ hide Hex).

La componente fondamentale di questa scheda è la schermata bianca dove la maggior parte delle informazioni di comunicazione si verificano mentre si usa X-CTU come emulatore del terminale: il testo blu si riferisce ai codici scritti e diretti fuori dalla porta seriale, mentre quelli in rosso sono i dati in arrivo dalla porta seriale;

**modem configuration:** le funzioni base in questo caso sono quattro, fornire l'interfaccia per le impostazioni di comunicazione tra i due modem, lettura e scrittura del firmware per i microcontrollori delle radio, download per l'aggiornamento del firmware, lettura, scrittura e ripristino delle impostazioni per la comunicazione [9].

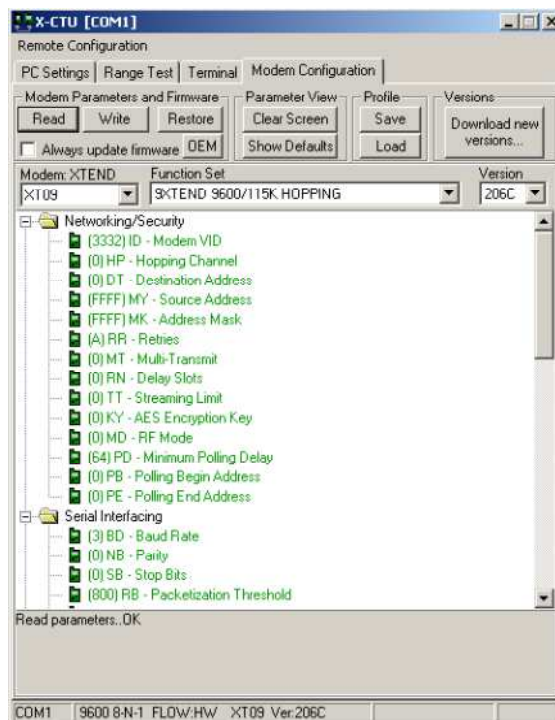


Figura 12 Scheda Modem Configuration



## Capitolo 4

### 4.1 Linguaggio di programmazione C

Il linguaggio C è un linguaggio di programmazione orientato agli oggetti ad alto livello, progettato da Dennis Ritchie nel 1972 presso i Bell Laboratories come evoluzione dal linguaggio B ed usato per la scrittura dei primi sistemi operativi UNIX. Utilizzato ancora tutt'oggi come linguaggio di riferimento per la realizzazione di software di sistema su gran parte delle piattaforme hardware, ma anche in altri campi come le telecomunicazioni, controlli dei processi industriali e software real-time, il C ebbe un enorme successo tra gli anni '70 e '80 con lo sviluppo di molti compilatori indipendenti che ne permisero la diffusione, tanto che l'ANSI (American National Standards Institute) decise di standardizzarlo tra il 1983 e il 1989 permettendo la portabilità dei programmi su qualsiasi piattaforma.

Nonostante lo sviluppo di molti altri linguaggi di programmazione orientati agli oggetti, (Java, Python, Visual Basic), il linguaggio C risulta ancora uno dei più utilizzati e più richiesti a livello aziendale, anche se da alcuni anni, nelle scuole superiori e nelle università è stato rimpiazzato da linguaggi più moderni, ma meno adoperati nel mondo del lavoro.

Il linguaggio C può essere considerato a tutti gli effetti un linguaggio ad alto livello, ma risulta meno astratto rispetto a linguaggi simili e con meno parole chiave (circa 30) che lo accostano all'assembler [10].

Il processo di sviluppo dei programmi realizzati in C passa attraverso 6 fasi: editare, preelaborare, compilare, linkare, caricare, eseguire.

- 1) Editare: è la scrittura del codice in un file che si esegue con un programma chiamato dall'editor.
- 2) Preelaborare: dopo aver eseguito le eventuali correzioni, il programma viene immagazzinato in un dispositivo di memoria di massa (hard disk); il nome del file del programma deve terminare con l'estensione **.c**.
- 3) Compilazione: traduce il programma nel linguaggio macchina; prima che inizi la traduzione viene eseguito il programma preprocessore, il quale obbedisce

a comandi speciali (direttive) dove indica le variazioni da eseguire prima della compilazione vera e propria. Tali manipolazioni consistono nell'inclusione di altri file in quello da compilare .

- 4) Linkare: ha il compito di collegare il codice oggetto con le funzioni mancanti che vengono definite altrove (ad esempio le librerie standard).
- 5) Caricare: un programma prima di essere eseguito dovrà essere caricato nella memoria; il caricatore prende l'immagine eseguibile dal disco e la trasferisce nella memoria.
- 6) Esecuzione: il programma esegue un'istruzione alla volta sotto il controllo della CPU [11].

#### **4.1.1 Vantaggi del linguaggio C**

- 1) dimensioni ridotte del codice e dell'eseguibile;
- 2) portabilità dei programmi;
- 3) buona efficienza;
- 4) largo uso di puntatori;

## **4.2 Elementi fondamentali di un programma in C**

### **4.2.1 Variabili**

Una variabile identifica una porzione di memoria destinata a contenere dati che, al contrario delle costanti, possono essere modificate durante l'esecuzione del programma. Le variabili devono essere dichiarate prima di essere adoperate, e sono caratterizzate da un "tipo" che delimita le operazioni possibili e il range di dati di cui può disporre. Generalmente le variabili sono considerate come L-value (l'operando di sinistra che rappresenta il nome dell'indirizzo di memoria).

TIPO	BIT	VALORI AMMESSI	Precisione
character	8	da -128 a 127	
unsigned character	8	da 0 a 255	
short integer	16	da -32768 a 32767	
unsigned short integer	16	da 0 a 65535	
integer	16	da -32768 a 32767	
unsigned integer	16	da 0 a 65535	
long integer	32	da -2147483648 a 2147483647	
unsigned long integer	32	da 0 a 4294967295	
floating point	32	da $3.4 \cdot 10^{-38}$ a $3.4 \cdot 10^{38}$	7 cifre
double precision	64	da $1.7 \cdot 10^{-308}$ a $1.7 \cdot 10^{308}$	15 cifre
long double precision	80	da $3.4 \cdot 10^{-4932}$ a $1.1 \cdot 10^{4932}$	19 cifre

Figura 13 Tabella tipo di dati

## 4.2.2 Operatori

Gli operatori che vengono utilizzati nel linguaggio C sono :

- aritmetici: somma (+), differenza (-), prodotto (\*), quoziente (/), resto (%)  
incremento (++) e decremento (--);
- di confronto: maggiore (>), maggiore o uguale (>=), minore (<), minore o uguale (<=), uguale (==) e diverso (!=);
- logici: AND logico (&&), OR logico ( || ).

E vengono rispettate le seguenti proprietà:

- arietà: è il numero di argomenti che un operatore può accettare, ad esempio, il simbolo + ha arietà 2 (A + B) mentre il simbolo ++ ha arietà 1 (A++);
- precedenza: valore che identifica quali sono gli operatori più importanti in un'espressione; più è importante l'operatore e minore è il valore che lo identifica;
- associatività: a parità della stessa arietà decide a chi viene data la precedenza, a seconda che l'associatività sia a sinistra o a destra.

## 4.3 Controlli

### 4.3.1 Controlli condizionali

Sono istruzioni che verificano una condizione, e a seconda che il risultato di tale condizione sia vero o falso, viene eseguita o saltata una parte di programma.

Le parole chiave relative ai controlli condizionali sono:

-if\else: se la condizione viene verificata, esegue l'istruzione immediatamente successiva, altrimenti salta il blocco di istruzioni e procede con il successivo; la chiave else viene usata in alternativa ad un blocco di istruzioni if che non viene verificato.

#### Esempio 1

```
if (condizione)
{
/* istruzione da eseguire se la condizione è Vera */
}
else
{
/* istruzione da eseguire se la condizione è Falsa */
}
```

-switch\case: valuta il valore di un espressione ed esegue le istruzioni il cui case coincide con tale valore. La differenza principale tra le istruzioni switch\case e if\else è che il primo può eseguire solo confronti di uguaglianza, mentre il secondo può valutare diversi operatori logici.

#### Esempio 2

```
switch (variabile)
{
case valore1:
// istruzione1
break;

case valore2:
// istruzione2
break;

case valoreN:
```



```
        // istruzioneN
break;
}
```

### 4.3.2 Controlli iterativi

Sono comandi che consentono di ripetere una sequenza di istruzioni fino a quando non si avvera una certa condizione: uno dei principali errori che si possono avere con questi cicli è la creazione di condizioni irrealizzabili che porta conseguentemente a dei cicli infiniti.

Le parole chiave relative a questi controlli sono:

-while: continua a produrre un ciclo di istruzioni fino a quando l' espressione iniziale non risulta falsa.

Esempio 3

```
while( condizione ) {
    istruzione1;
    istruzione2;
}
```

-do\while: diversamente dal while, valuta l'espressione condizionale alla fine del ciclo e non all'inizio; in questo modo il ciclo di istruzioni viene eseguito almeno una volta, ovvero quando la condizione iniziale risulta subito falsa.

Esempio 4

```
do {
    [istruzioni]
}
while (condizione);
```

-for: molto più potente del while e do\while, l'istruzione for viene impiegata per cicli che si devono ripetere un certo numero di volte; la variabile che deve tenere conto dei cicli effettuati è un numero intero ed è definita variabile di controllo del ciclo.

## Esempio 5

```
for(<espressione1>; <espressione2>; <espressione3>)  
<istruzione>;
```

### 4.3.3 Array

Possiamo considerare gli array come un insieme ordinato di variabili dello stesso tipo, alle quali si fa riferimento tramite un unico nome e, l'accesso al singolo valore (o elemento) del vettore si ottiene mediante un indice. Il primo elemento di un array ha come indice il valore zero, mentre l'ultimo ha come indice il valore della sua dimensione meno 1 (esempio: `int mesi[11]`).

## 4.4 Funzioni e puntatori

### 4.4.1 Funzioni

Le funzioni vengono considerate come un insieme di dati e istruzioni a cui è associato un nome e che consentono un risparmio ed un riutilizzo del codice.

Il linguaggio C assume che ogni funzione ritorni un valore, utilizzando l'istruzione `return` (che può essere seguita anche da un valore); se non si utilizza il `return`, nessun parametro deve essere impostato quando viene chiamata la funzione: ogni funzione è costituita da un'intestazione in cui è indicato il nome e un corpo dove sono racchiuse le istruzioni da eseguire [10].

### 4.4.2 Puntatori

I puntatori sono variabili il cui valore è un indirizzo di memoria di una variabile, di una funzione o di un'area di memoria. L'esempio `int *indice`, mostra che il contenuto della locazione di memoria puntata alla variabile (`indice`) è un `int`. Una variabile puntatore può contenere solo indirizzi di memoria [10].

# Capitolo 5

## 5.1 Sviluppo del codice realizzato

Al fine di realizzare un codice di questo tipo, bisogna tenere presente i compiti e le specifiche del trasmettitore e del ricevitore durante l'invio dei dati, in particolare bisogna tenere in considerazione i seguenti aspetti:

- **Robustezza:** è la capacità di comportamento di un programma in presenza di errori o di situazioni impreviste, più un programma è robusto e più sarà alta la probabilità di riuscita anche in situazioni non idonee;
- **Sicurezza:** l'apparecchio che riceve il segnale deve saper riconoscere chi ha inviato il segnale, e di conseguenza impostare i valori desiderati per il tipo di comando che è stato inviato;
- **Tempo di attesa:** tempo entro il quale deve arrivare il segnale al dispositivo ricevente;
- **Codice di ritorno:** come già spiegato nei capitoli precedenti, è un segnale (acknowledgment) di ricevuta trasmissione da parte dell'apparecchio e che quindi permette di terminare la trasmissione;
- **Interferenze:** nelle connessioni wireless, le interferenze sono molto comuni e problematiche; in caso di disturbi la comunicazione tra il pedale e la macchina viene spostata automaticamente in un nuovo canale, fino all'invio del messaggio.

Le ultime tre caratteristiche sono già controllate in automatico dalle schede Xbee che andremo ad utilizzare, le quali, come già ripetuto, rispettano lo standard IEEE 802.15.4.

Lo sviluppo del codice quindi si concentrerà sulla trasmissione del segnale, sulla sua robustezza e sulla rilevazione e correzione degli errori, definendo l'uso dei seguenti programmi:

- CRC
- Distanza di Hamming
- Trasmissione Tx/Rx

I tre programmi utilizzati sono stati sviluppati, eseguiti e verificati singolarmente: come verrà spiegato in seguito, le conclusioni riportate sono ancora parziali e quindi non si ha ancora la conferma dell'efficacia dei codici utilizzati. Tuttavia, si è deciso comunque di adoperarli sia per la semplicità dei loro algoritmi, che per la loro diffusione e conoscenza; questo ha consentito un controllo e una verifica a livello logico e a livello di programmazione più rapida e funzionale.

### 5.1.1 CRC

Il CRC (Cyclic Redundancy Check ovvero controllo a ridondanza ciclica) è un codice molto diffuso per la sua semplicità (richiede poche conoscenze matematiche), per la sua implementazione binaria e perché è un ottimo rivelatore (ma non correttore) di errori di trasmissione in caso di rumori indesiderati [12].

Il controllo consiste nel prendere i dati in ingresso e farli scorrere ciclicamente in una rete logica che contiene elementi ridondanti che permettano la rilevazione dell'errore.

Il principio del CRC consiste nel trattare delle sequenze binarie come dei polinomi lineari in modo tale che il bit meno significativo rappresenti il termine noto del polinomio, mentre il bit più significativo rappresenterà il monomio di grado massimo (es. La sequenza binaria 1101 diventa  $x^3 + x^2 + x^0$ ).

Supponendo che il CRC abbia un polinomio di grado  $g$  ed il messaggio inviato ha grado  $n-1$ , con  $n$  pari al numero di bit del messaggio stesso, si ottiene un polinomio  $p(x)$  di grado  $h$  con  $h = n-1+g$  (es. Se abbiamo un messaggio di 8 bit 10100100, e quindi un polinomio  $b(x) = x^7 + x^5 + x^2$ , ed un CRC con polinomio di grado  $g = 4$ , otterremo un polinomio di grado  $h = 11$  e  $p(x) = x^{11} + x^9 + x^6$ ).

Si esegue quindi la divisione polinomiale  $p(x)/g(x)$ , ottenendo un quoziente  $q(x)$  e un resto  $r(x)$ : il messaggio finale sarà dato dalla somma  $m(x) = p(x) + r(x)$  con  $r(x)$  di grado inferiore a  $g(x)$ , inoltre essendo  $p(x)$  una traslazione di  $g$  posizioni di  $b(x)$ , si ottiene che nella somma dei due polinomi non ci sarà mai una sovrapposizione dei termini di ugual grado.

Il polinomio  $p(x)$  è definito anche come  $p(x) = q(x) * g(x) + r(x)$ , che sostituito alla definizione precedente, si ottiene  $m(x) = q(x) * g(x) + r(x) + r(x)$ : tenendo conto che i calcoli sono svolti in modulo 2 e che quindi la somma e la differenza tra i due polinomi viene eseguita usando la funzione logica XOR, abbiamo che  $r(x) + r(x) = 0$  e  $m(x) = q(x) * g(x)$ , quindi se la divisione non produce resto, il messaggio ricevuto risulta effettivamente quello che è stato inviato, in caso contrario si sono verificati degli errori. Il CRC può tuttavia riscontrare dei problemi in caso di errori multipli nel caso in cui il resto della divisione fosse comunque zero [12].

Il codice CRC utilizzato nel linguaggio di programmazione C è il seguente:

```
#include <stdio.h>
#define PG 0x11021 /* Polinomio generatore */

int crc(int oldcrc, unsigned char toadd); /* Aggiunge toadd al CRC calcolato */

main()
{
    int r=0; /* CRC
    int c; /* Carattere letto
    int n = 0; /* Contatore dei caratteri

    printf("Inserire una sequenza di caratteri: ");
    while((c=getchar())!=EOF)
    {
        r=crc(r,c); /* Aggiunge il carattere letto al CRC dei precedenti
        n++; /* Conteggio dei caratteri letti
    }
    r=crc(r,0); /* Aggiunge al CRC i primi otto zeri
    r=crc(r,0); /* Aggiunge al CRC i secondi otto zeri

    printf("\nLetti %d caratteri, CRC calcolato: %x\n", n, r);
}

/* Aggiunge i bit del codice oldcrc al dividendo, per
 * calcolare il nuovo CRC
 */
int crc(int oldcrc, unsigned char toadd)
{
    int i; /* Indice del bit

    for(i=0;i<8;i++)
    {
        oldcrc= (oldcrc<<1) | (toadd>>7); /* Aggiunge il bit piu' significativo
        /* del codice in fondo al CRC
        toadd <<= 1; /* Toglie il bit gestito dal codice
        if((oldcrc & 0x10000) != 0)
        {
            oldcrc=oldcrc^PG;
        }
    }
    return oldcrc; /* Restituisce il CRC ricalcolato
}
```

### 5.1.2 Codice di Hamming

Il codice di Hamming è un codice correttore di singoli bit che si basa sulla distanza di Hamming, ovvero sul conteggio del numero di sostituzioni necessarie per uguagliare due stringhe aventi la stessa lunghezza. I parametri utilizzati nel codice sono  $\left[ \frac{q^m-1}{q-1}, \frac{q^m-1}{q-1} - m, 3 \right]$  dove  $q$  è la grandezza dell'alfabeto utilizzato, mentre  $m$  è il numero di bit adoperati.

Condizione necessaria e sufficiente affinché un codice sia in grado di rilevare  $E$  errori è che la distanza minima tra tutte le coppie sia almeno  $E + 1$ , mentre la condizione necessaria e sufficiente affinché un codice sia in grado di correggere  $E$  errori è che la distanza minima tra tutte le coppie  $d$  sia  $2E + 1$ , ovvero, se il codice ha distanza 2, vengono rilevati esclusivamente gli errori singoli, mentre se ha distanza 3 l'errore viene anche corretto; in generale per correggere un singolo errore sono necessari  $\log_2 m$  bit, con  $m$  pari al numero di bit che costituiscono il messaggio vero e proprio [13].

### 5.1.3 Costruzione del codice

Il codice è formato da un numero di bit  $n$  che corrispondono al messaggio originale e da un numero di bit  $r$  che corrispondono ai bit di controllo, secondo la formula  $m+1 \leq 2^r - r$ : i bit della parola di codice vengono numerati da sinistra a destra a partire dall'indice 1 e i bit di controllo, che rappresenteranno i bit di parità dei bit del messaggio, saranno posizionati negli indici aventi una potenza di due. In ricezione se tutti i valori dei bit di controllo sono corretti, la sequenza viene accettata, altrimenti la somma degli indici dei bit di controllo con valore sbagliato indicherà l'indice dove si verificato l'errore [13].

Il codice di Hamming utilizzato nel linguaggio di programmazione C è il seguente:

```
#include <stdio.h>

#define DIM 8      /* conversione in binario */

#define DIM1 256  // vettore in cui si memorizzano i valori aventi distanza
                  // di Hamming desiderata

int disth(int n1,int n2); // calcola la distanza di Hamming tra due vettori
int codici(int hamm[],int dist); // genera i numeri aventi la distanza di
                                // Hamming in ingresso

main()
{
int i;      // Indice per i cicli
int n;      // Numero di codici nella codifica
int hamm;   // Distanza di Hamming richiesta dall'utente
int flag;   // Interrompe il ciclo
int c;      // Codice inserito
int d=10;   // Distanza di hamming minima
int corr;   // Codice con distanza di Hamming minima
int l;      // Numero di codici che hanno distanza uguale al minimo
int vet[DIM1]; // Codice di riferimento calcolato

// input per la distanza di Hamming
do {
printf("Distanza di Hamming = ");
scanf("%d",&hamm);
} while (hamm<1 || hamm>DIM);

n = codici(vet,hamm); // Calcolo della codifica

// Stampa del codice prodotto
for(i=0;i<n;i++)
{
printf("%4d",vet[i]);
}
printf ("\n");

// prende in input il numero da controllare ed eventualmente da correggere
do {
printf("\nCodice: ");
scanf("%d",&c);
}while (c<0 || c>255);
```

```

// controlla che il numero non si già presente tra quelli generati
for(i=0, flag=0; i<n && flag==0;i++)
{
    if(vet[i]==c)
    {
        printf("%d appartiene alla codifica\n",c);
        flag=1;
    }
}

// correzione dell' errore se il numero è diverso
if(flag==0)
{
    printf("%d non appartiene alla codifica\n",c);
    for (i=0;i < n;i++) // Analizza tutti i codici della codifica
    {
        if(d==dsth(c,vet[i])) // Se la distanza e' uguale al minimo calcolato
        {
            l++; // Conta quanti codici hanno la stessa distanza
        }
        if(d>dsth(c,vet[i])) // Se la distanza e' inferiore al vecchio minimo
        {
            d=dsth(c,vet[i]); // La distanza corrente e' il nuovo minimo
            corr=vet[i]; // Il codice corrispondente e' il relativo minimo
            l = 1; // Caso in cui trova solo un minimo
        }
    }
    if(l > 1) // Se ho piu' di un minimo
    {
        printf("Non e' possibile correggere %d\n",c);
    } else { // Se invece il minimo e' uno solo
        printf("Correzione di %d: %d\n",c,corr);
    }
}
}

// calcolo della distanza di Hamming tra due codici usando operazioni binarie

int dsth(int a,int b)
{
    int i; // Codice contenete i bit diversi tra a e b
    int dist; // Distanza calcoalta

    dist=0; // Distanza iniziale
    i = (a ^ b); // Uso dell'exclusive or, discrimina i bit diversi

    if (i < 0)
    while (i != 0)// Finche' ci sono bit a 1 (bit diversi)
    {
        if(i & 1) // Se il bit meno significativo e' diverso da 0 (bit diversi)
        {
            dist++; // Conta la differenza
        }
        i = i >> 1; // Viene scartato il bit meno significativo
    }

    return dist; // Restituisce la distanza
}

```



```

/* genera i numeri aventi la distanza di Hamming passata in ingresso e
 * ritorna il numero di codici che appartengono alla codifica
 */
int codici(int hamm[],int dist)
{
int i;      // Codice corrente
int cont;
int j;
int n;      // Numero di codici inseriti nella codifica

    hamm[0]=0; // Inizializza il primo codice
    n=1;       // e conta i codici introdotti

    // Per ogni codice possibile oltre il primo
    for(i=1;i<DIM1;i++)
    {
        // E per ogni codice gia' inserito, se la distanza sbaglia, termina
        for (j=0;j<n && disth(i,hamm[j])>=dist ; j++)
            ;
        if(j==n) // Se passa tutti i test
        {
            hamm[n]=i; // Inserisce il codice
            n++;       // E lo conta
        }
    }
    return n; // Ritorna il numero di codici trovati
}

```

### 5.1.4 Codice Tx/Rx

Il seguente codice permette la trasmissione radio tra il pedale, che invia il segnale, e il dispositivo, che lo riceve: anche se i moduli Xbee utilizzati ricevono e trasmettono il segnale in modalità RF, il programma consente di effettuare un collegamento anche in modalità API (Application Programming Interface):

```
#define _CTS  PORTC.F0
#define _RTS  LATC.F1

#define Delimiter  0x7E  // Limitatore
#define DH  00  // MSB destination id
#define DL  01  // LSB destination id
#define TX_request_16  1  // identificatore API
#define Frame_id  0  // frame id

void API_frame_send(char *str){
    char checksum = 0;
    char count = 0;
    while(*str != 0){
        checksum += *str;  //somma totale del codice
        count++;
        str++;
    }
    checksum += TX_request_16;  //identificatore API
    checksum += Frame_id;  //frame id
    checksum += DH;  //indirizzo MSB destination
    checksum += DL;  //indirizzo LSB destination
    checksum += 1;
    checksum = 0xFF - checksum;

    str -= count;  //resetta il puntatore
    count += 5;  //aggiunta di 5 byte al contatore
}
```

```

    UART1_write(Delimiter); // Limitatore
    UART1_write(0);         // lunghezza byte del MSB
    UART1_write(count);     // lunghezza byte del LSB
    UART1_write(TX_request_16);
    UART1_write(Frame_id);
    UART1_write(DH);
    UART1_write(DL);
    UART1_write(1);
    while(*str != 0){
        UART1_write(*str);
        str++;
    }
    UART1_write(checksum);
}

main() {
    int TRISB = 0;
    int TRISC = 1;
    int ADCON1 = 0x0F;
    int RTS = 0;
    UART1_init(9600);

    while(1){
        API_frame_send("API structure test");
        delay_ms(1000);
    }
}

```



## Capitolo 6

### 6.1 Conclusioni e prospettive future

In questo lavoro di tesi sono state mostrate le caratteristiche principali dei dispositivi utilizzati e in particolar modo sulle schede utilizzate per la trasmissione dei dati, sul protocollo di trasmissione stesso con le relative norme e sul dispositivo dove verranno impiantati i moduli.

Nella fase di progettazione invece si è svolto un codice che permetta la trasmissione del segnale, cercando di evitare possibili errori che creano disturbi o interferenze. In particolare si è cercato di garantire un buon livello di robustezza, cioè che sia in grado di proteggere il sistema in caso di malfunzionamenti, con un accurato controllo degli errori, e in modo particolare, gli errori a singolo bit, che vengono rilevati e quindi corretti in automatico.

I risultati di questa tesi sono tuttavia ancora parziali, poiché non sono ancora state compiute delle prove sul dispositivo medico: in futuro è prevista la costruzione di un nuovo prototipo di Vesalius<sup>®</sup>, e i test effettuati su tali macchine, applicando le norme armonizzate, dovranno garantire il rispetto dei requisiti essenziali riportati nelle direttive.

Le prove saranno dedicate principalmente al tempo di durata della batteria (deve essere garantito il funzionamento dei pedali anche in caso di operazioni di lunga durata) ai problemi di interferenza tra la macchina e i pedali (il Vesalius<sup>®</sup> lavora a frequenze nell'ordine dei MHz, mentre le schede Xbee lavorano a 2.4 GHz) ed alla verifica dell'indirizzamento univoco del pedale (non devono verificarsi delle attivazioni involontarie). In generale sarà effettuata una simulazione del sito di funzionamento della macchina (sala operatoria) e tutti i problemi ad esso associati (durata dell'operazione, interferenze tra i dispositivi) che dovranno essere superati per certificare la validità del dispositivo.

Le interfacce wireless, come spiegato nei capitoli precedenti, rientrano nello scopo della direttiva R&TTE 1995/5/CE ed in particolare devono rispettare gli articoli 3.1a e b che riguardano la sicurezza dei pazienti o chiunque in vicinanza della macchina, e l'articolo 3.2 riguardo allo spettro delle frequenze terrestri/spaziali.

Per quanto riguarda la sicurezza bisogna considerare valida l'approvazione ottenuta con lo standard di protocollo EN 60601-1 [14], considerato il punto di riferimento per i fabbricanti e gli utilizzatori dei dispositivi medici, dove, con la pubblicazione della terza edizione nel 2006, viene amplificato il concetto di sicurezza anche in riferimento delle prestazioni essenziali della macchina. La normativa introduce anche delle prescrizioni generali per la gestione del processo di rischi di un dispositivo, allo scopo di garantire la sicurezza dell'apparecchio, allineandosi a concetti già riportati nella direttiva 93/42, riguardante i dispositivi medici.

La norma armonizzata per la Compatibilità Elettromagnetica (ECM) invece da indicazioni di come le perturbazioni elettromagnetiche provocate dalle apparecchiature non impediscano il corretto funzionamento di altri dispositivi e delle reti in generale e che abbiano un adeguato livello d'immunità contro le perturbazioni elettromagnetiche stesse [15].

Infine è importante segnalare che è in avanzata fase di studio una nuova versione di questo standard, denominata IEEE 802.15.4e, basata su meccanismi di accesso multiplo di tipo Time Slotted Channel Hopping (TSCH), in grado di garantire una maggiore robustezza alle interferenze elettromagnetiche esterne e ridotte probabilità di collisione dei dati.

## Appendice A

Esempio di codice di Hamming [14]

Consideriamo il seguente messaggio di 9 bit:

0 1 0 0 1 1 0 1 0

Ricordando la formula  $m+1 \leq 2^r - r$ , si ha che il messaggio avrà 4 bit di controllo, divisi in questo modo :

$r_1$	$r_2$	0	$r_3$	1	0	0	$r_4$	1	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13

Il bit di parità  $r_1$  controlla i bit di indice dispari, quindi si avrà:

0	$r_2$	0	$r_3$	1	0	0	$r_4$	1	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13

Il bit di parità  $r_2$  controlla i bit con indice la cui scomposizione in binario include  $2^1$ :

0	1	0	$r_3$	1	0	0	$r_4$	1	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13

Il bit di parità  $r_3$  controlla i bit con indice la cui scomposizione in binario include  $2^2$ :

0	1	0	0	1	0	0	$r_4$	1	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13

Il bit di parità  $r_4$  controlla i bit con indice la cui scomposizione in binario include  $2^3$ :

0	1	0	0	1	0	0	$r_4$	1	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13

Ottenendo così il messaggio codificato.

Se si verifica un errore:

0	1	0	0	1	0	<b>1</b>	1	1	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13

La somma degli indici dei bit di controllo discordanti indicherà dove si trova l'errore:

1	0	0	1	1	0	<b>1</b>	1	1	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13

Infatti i bit di controllo discordanti sono 1, 2, 4:  $1 + 2 + 4 = 7$ .



## **Bibliografia**

- [1] Dispositivi medici, aspetti regolatori e operativi, conferenza nazionale sui dispositivi biomedici;
- [2] Atlante di Chirurgia a Risonanza Quantica Molecolare, raccolta di testi curata da Aldo Bolzoni;
- [3] “ Vesalius<sup>®</sup> ”, dott. Gianantonio Pozzato, dott. Giuseppe Vignato, Quintessence International 05/06/2003;
- [4] IEEE std. 802.15.4, 01/10/03;
- [5] R&TTE 1995/05/CE, consiglio del 09/03/99;
- [6] Direttiva 73/23 – CEE;
- [7] Direttiva 89 – 336 – CEE;
- [8] Xbee/Xbee pro Modules, Digi International Inc. 2009;
- [9] Configuration & Test Utility, Software X-CTU, Digi International Inc. 2009;
- [10] Linguaggio C e C++, M. Romagnoli, P. Ventura, Petrini editore, 2005;
- [11] Linguaggio di programmazione C, A. Scaringella e S. Bochicchio;
- [12] Calcolo CRC di una sequenza di caratteri, S. Salvi;
- [13] Codici di Hamming, prof. A. di Santo, Dipartimento di Informatica e Sistemistica di Roma [pag. 7 – 22];
- [14] Direttiva CEI EN 60601-1, terza edizione;
- [15] Direttiva Compatibilità Elettromagnetica 2004 – 108 CE;