

POLITECNICO DI TORINO

III Facoltà di Ingegneria dell'Informazione  
Corso di Laurea in Ingegneria Elettronica

Tesi di Laurea Magistrale

**Progetto di una scheda con  
funzione di ricetrasmittitore UHF  
per un satellite universitario**



**Relatore:**

prof. Claudio Sansoè

**Candidato:**

Rosanna GENOVESE

Luglio 2009

*Alla mia famiglia e a tutti i miei Angeli.....*

# Sommario

*Il progetto AraMiS (Architettura Modulare per Satelliti) ha inizio nell'autunno del 2006 sulla base dell'esperienza appurata dal progetto PiCPoT (Piccolo Cubo del Politecnico di Torino), il primo nanosatellite realizzato dal Politecnico di Torino, che ha così aderito ad un'iniziativa internazionale, dall'elevata valenza didattica, che concerne la progettazione di un satellite in ambito universitario.*

*La filosofia progettuale standard adottata da diverse università in tutto il mondo si basa sul concetto di Cubesat, un satellite di forma cubica con un lato di circa 10 cm e una massa di un 1 Kg al massimo.*

*PiCPoT è un piccolo cubo di 13 cm di lato contenente, all'interno, sottosistemi elettronici sviluppati ad hoc per svolgere le funzioni di trasmettere dati (per esempio, le misure dei sensori di bordo alla Stazione di Terra), scattare fotografie, valutare il funzionamento del GPS in orbita LEO e quello dei componenti COTS (Components Off The Shelf) nello spazio.*

*Per proseguire l'attività iniziata col progetto precedente, è nata l'idea di base del progetto AraMiS che si sviluppa seguendo il concetto di modularità. Quest'ultimo si compone di un discreto numero di moduli, detti tiles, preassemblati e precollaudati che hanno la caratteristica di poter essere riutilizzati a seconda delle esigenze di ciascuna missione.*

*Ciò permette una realizzazione low cost e tempi di sviluppo molto ristretti per il progetto.*

*I moduli, comunicanti tra di loro tramite un bus seriale, hanno dimensione standard; alcuni saranno collocati sulla superficie esterna del satellite, altri al suo interno.*

*Questo lavoro di tesi intende occuparsi del modulo di comunicazione Telecommunication Tile che lavora alla frequenza di 2.4 GHz e si occupa di gestire la comunicazione relativa alla trasmissione e ricezione di dati e comandi con la Stazione di Terra.*

*Il processore di bordo controlla, mediante un opportuno protocollo, il transceiver: il CC2510, un dispositivo della Chipcon TI dalle alte prestazioni RF con un microcontrollore basato sull'architettura dell'8051 enhanced.*

*Il lavoro è cominciato con lo sviluppo del programmatore del transceiver, allora non ancora reso disponibile dall'azienda produttrice, ed è proseguito con l'implementazione del software di trasmissione e ricezione di pacchetti di dati, con la conseguente analisi delle casistiche di funzionamento e valutazione delle problematiche derivanti dalle tipologie di eventuali errori.*

*Alla base della comunicazione sono stati utilizzati due protocolli: il protocollo wireless e il protocollo wired.*

*La ritrasmissione avviene attraverso l'invio di un comando al microcontrollore il quale, una volta ricevuto correttamente, dà l'avvio alla trasmissione e ricezione di pacchetti via radio.*

*A tal proposito, è stato determinato un certo tempo di attesa massimo tra l'invio di un pacchetto e la conferma di avvenuta ricezione: il Timeout.*

*La trasmissione wireless è soggetta a fenomeni di interferenza che abbassano la qualità della comunicazione.*

*In particolare, nel progetto sono riportati i risultati sperimentali di prove eseguite sia a livello fisico, in termini di misurazioni di potenza del segnale utile, sia a livello di canale attraverso una stima della probabilità d'errore, il BER (bit error rate) usando un adeguato modello probabilistico.*

*Inoltre è stato preso in considerazione ed analizzato sperimentalmente il fenomeno fisico dell'Effetto Doppler che potrebbe influenzare la comunicazione tra il satellite e la Stazione di Terra, rispetto alla quale esso è in moto relativo.*

*La tesi procede con i calcoli dei parametri di tratta, il link budget, dimensionando così la potenza utile in trasmissione nel collegamento downlink (Satellite-Stazione di Terra) e la potenza utile in ricezione nel collegamento uplink (Stazione di Terra-Satellite).*

*Il transceiver, progettato per applicazioni a corto raggio, eroga in uscita una potenza non sufficiente a supportare la trasmissione di una comunicazione che avviene nello spazio, rendendo necessaria la ricerca di un amplificatore di potenza: l'SZM-2166Z, un dispositivo della Sirenza Microdevices in grado di fornire i 4W di potenza in uscita necessari.*

*La suite di tool del software Mentor Graphics ha conseguentemente permesso la realizzazione del circuito stampato dell'amplificatore di potenza che è stato sottoposto a test di prova, prima utilizzando la strumentazione del Dipartimento di Elettronica e poi quella della Spin Electronics, un'azienda che si occupa di telecomunicazioni, verificandone così il corretto funzionamento.*

*Per quanto riguarda il comportamento del transceiver in ricezione, ci si è serviti di un amplificatore a basso rumore per migliorare la sensitivity e garantirne un valore tale da consentire il massimo data rate di 500 kbps.*

*Si tratta del MAX2644, un dispositivo della MAXIM già collaudato durante lo svolgimento del progetto PiCPoT e dotato di un alto guadagno e una bassa cifra di rumore.*

*Da queste realizzazioni progettuali si delinea, a conclusione del lavoro di tesi, l'intera architettura del sistema di comunicazione: la schedaTxRx.*

*La scheda sarà connessa al processore di bordo tramite una sola interfaccia servendosi di un transceiver master e un transceiver slave comunicanti tra loro tramite una seriale asincrona.*

*Un transceiver lavorerà in trasmissione con il supporto dell'amplificatore di potenza e l'altro in ricezione con l'appoggio dell'amplificatore a basso rumore.*

*Il sistema si chiude su due switch identici, gli FMS2031-001 della Filtronic, pilotati in opposizione di fase tra di loro (quando uno è chiuso l'altro è aperto) garantendo un alto isolamento tra la trasmissione e la ricezione, eliminando così l'anello di reazione che renderebbe instabile il sistema.*

*Sulla base degli assorbimenti dei sottosistemi da alimentare è stato progettato, in particolare, l'alimentatore switching LM2596 della National Semiconductor a tensione di uscita variabile.*

*Scelti i componenti da adottare per la scheda, la fase successiva è consistita nel disegno dello schema elettrico e del circuito stampato del sistema.*

*Si auspica che la scheda in oggetto possa essere realizzata in tempi brevi per dare attuazione al discorso progettuale che è stato il filo conduttore di questo lavoro di tesi il quale mi ha dato l'opportunità di lavorare su un percorso sperimentale seguendolo nel concreto e, quindi, mi ha permesso di arricchirmi dell'esperienza diretta della problematica oggetto di studio e del lavoro di gruppo di cui necessitano progetti di tale portata.*

# Indice

<b>Sommario</b>	<b>III</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Il progetto PiCPoT . . . . .	2
1.2 Il progetto AraMiS . . . . .	4
<b>2 I sistemi di comunicazione</b>	<b>7</b>
2.1 Bande di frequenza . . . . .	9
2.2 Caratteristiche della propagazione delle onde a radiofrequenza . . . . .	11
2.3 Sistemi di comunicazione satellitare . . . . .	15
<b>3 Il transceiver CC2510</b>	<b>18</b>
3.1 L'8051 CPU . . . . .	19
3.2 La radio . . . . .	21
3.3 Evaluation module del CC2510 . . . . .	22
3.4 Il protocollo xmodem . . . . .	26
3.5 La debug interface . . . . .	27
3.6 Il file.hex . . . . .	32
3.7 Programmazione della memoria Flash . . . . .	33
3.8 Programmatore del CC2510 . . . . .	35
3.9 Interfaccia Utente . . . . .	36
<b>4 Software di trasmissione (Tx) e ricezione (Rx)</b>	<b>40</b>
4.1 Struttura dei pacchetti . . . . .	41
4.2 Gestione delle funzionalità della Radio . . . . .	42
4.3 Gestione della comunicazione via seriale Rs-232, libreria uart.h . . . . .	54
4.4 Principio di funzionamento della Ricetrasmisione: main.c . . . . .	56
4.5 Interfaccia Utente . . . . .	64
4.6 Analisi dell'effetto Doppler . . . . .	67
4.7 Probabilità d'errore per un sistema di comunicazione binario . . . . .	71
4.8 Test del CC2510 . . . . .	74

<b>5</b>	<b>Progetto dell'amplificatore di potenza: SZM-2166Z</b>	<b>78</b>
5.1	Dimensionamento della potenza in trasmissione . . . . .	78
5.2	L'amplificatore di potenza SZM-2166Z . . . . .	81
5.3	I componenti passivi . . . . .	82
5.4	Realizzazione della libreria centrale . . . . .	83
5.5	Realizzazione dello schema elettrico . . . . .	89
5.6	Realizzazione del circuito stampato: <i>PCB, Printed circuit board</i> . . .	91
5.7	Test di funzionamento . . . . .	97
<b>6</b>	<b>Progetto del sistema di comunicazione a 2.4 GHz: la scheda TxRx100</b>	
6.1	Dimensionamento della potenza in ricezione . . . . .	100
6.2	Progetto del sistema di comunicazione . . . . .	101
6.3	L'alimentatore switching LM2596 . . . . .	104
6.4	Dimensionamento dei componenti dell'LM2596 . . . . .	104
6.5	Inserimento dello switching nella Libreria Centrale di Mentor Graphics	108
6.6	Lo switch FMS2031-001 . . . . .	110
6.7	Inserimento dello switch nella libreria Centrale di Mentor Graphics .	111
6.8	Analisi e revisione dei restanti componenti della scheda . . . . .	112
6.9	Realizzazione dello schema elettrico del sistema di comunicazione e relativo PCB . . . . .	113
<b>7</b>	<b>Conclusioni</b>	<b>123</b>
<b>A</b>	<b>Il CD-ROM allegato</b>	<b>125</b>
	<b>Bibliografia</b>	<b>127</b>

# Capitolo 1

## Introduzione

Negli ultimi anni c'è stato un forte aumento nella realizzazione di satelliti di piccole dimensioni, detti picosatelliti o nanosatelliti, da parte di Università di tutto il mondo, dovuto a costi e tempi di sviluppo significativamente contenuti.

I picosatelliti hanno un peso inferiore ad 1 Kg, mentre i nanosatelliti un peso che può variare da 1 Kg a 10 Kg.

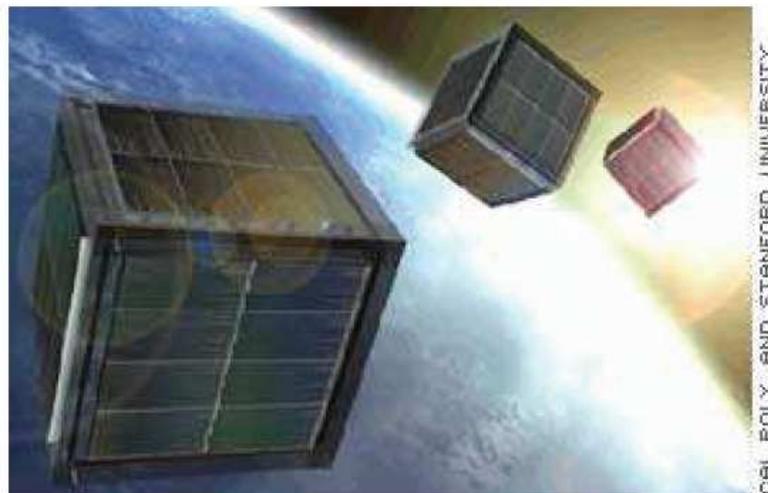


Figura 1.1. Prototipo di Cubesat ideato dalla CalPoly e dalla Stanford University

La filosofia progettuale standard seguita si basa sul concetto di Cubesat, un satellite di forma cubica di circa 10 cm di lato e 1 Kg di massa al massimo.

Questo progetto è stato sviluppato nel 2001 dal professore Robert Twiggs, docente alla Stanford University, in collaborazione con la California Polytechnic State University, con l'obiettivo appunto di realizzare uno standard per lo sviluppo di picosatelliti che ne facilitasse la progettazione.

L'idea di progettare un satellite è stata adottata da molte Università italiane e straniere, tra le quali citiamo l'Università di Trieste con Atmocube (I), l'Università della Sapienza con Unisat, l'Università Wurzburg (D) con AAU-Cubesat, la Norwegian University of Science and Technology (NO) con NCube.

Anche il Politecnico di Torino ha aderito all'iniziativa internazionale di progettare un satellite universitario, prima con il progetto PicPot e attualmente con il progetto AraMiS.

## 1.1 Il progetto PiCPoT

Il progetto PiCPoT nasce nel Gennaio del 2004 in collaborazione tra alcuni dipartimenti dell'Ateneo, in particolare il Dipartimento di Ingegneria Elettronica e il Dipartimento di Ingegneria Areospaziale.

PiCPoT, acronimo di *Piccolo Cubo del Politecnico di Torino*, è un satellite di forma cubica di 13 cm di lato, una massa di circa 2,5 Kg e un tempo di vita previsto di 90 giorni.

Gli obiettivi principali del progetto sono stati:

- Verificare il funzionamento dei componenti COTS (Components Off The Shelf) nello spazio
- Trasmettere dei dati, quali le letture dei sensori di bordo alla Stazione di Terra
- Scattare fotografie a bassa risoluzione dalla superficie terrestre attraverso tre fotocamere.

La struttura esterna del satellite è un cubo composto da sei facce quadrate e ortogonali tra di loro in lega di alluminio tipo 5000 AlMn (Fig. 1.2).

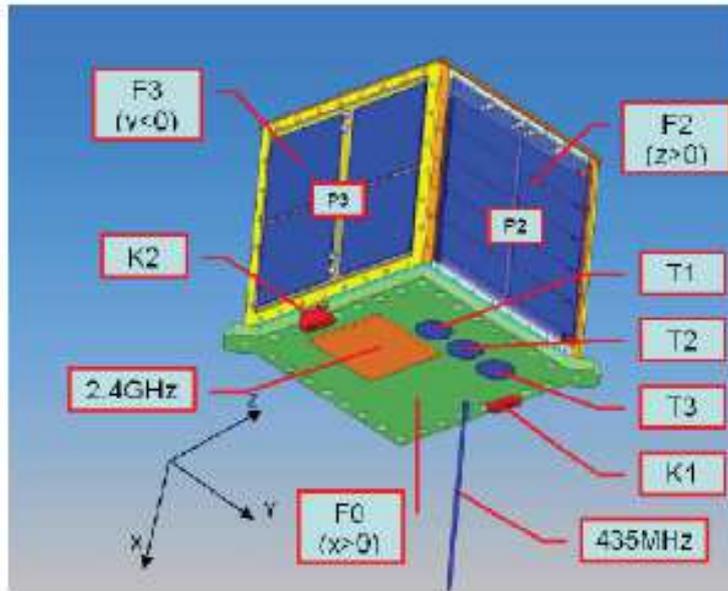


Figura 1.2. Vista esterna PiCPoT

Su di esse sono presenti cinque pannelli solari (P1, P2, P3, P4, P5), due antenne (alle frequenze di 437 MHz e 2.4GHz), tre fotocamere (T1, T2, T3), due kill-switch (K1, K2), e un connettore di test per testare la parte elettronica del satellite dopo essere stato montato.

Internamente è costituito da sei schede elettroniche (Fig. 1.3):

- Power Supply, che mantiene cariche le batterie ricaricabili del satellite e controlla lo stato elettrico e termico dei pannelli solari, dei caricabatterie e delle stesse batterie
- Power Switch, che genera, a partire dalle tensioni delle batterie, le tensioni per le altre schede del satellite
- TxRx, che trasmette e riceve dati tra il satellite e la Stazione di Terra, utilizzando due canali di comunicazione, uno alla frequenza di 437 MHz e l'altro a 2.4 GHz
- ProcA e ProcB, i processori di bordo che eseguono i comandi ricevuti dalla Stazione di Terra e trasmettono alle due diverse frequenze le fotografie e la telemetria del satellite
- Payload, che acquisisce le immagini dalle tre fotocamere, le comprime in formato JPEG e le trasmette su richiesta a uno dei due processori.

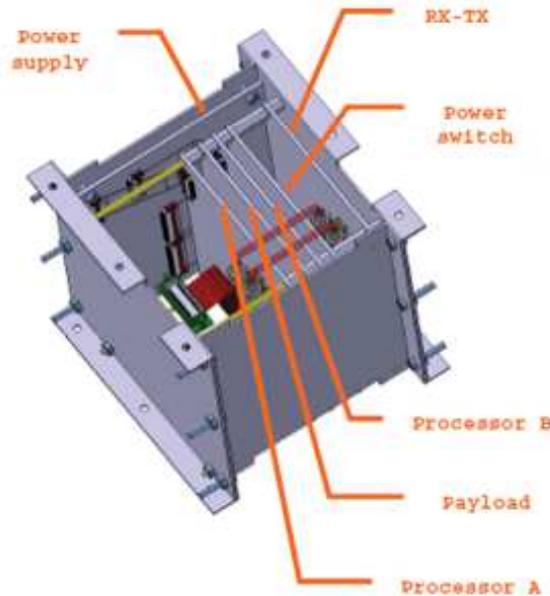


Figura 1.3. Elettronica di bordo PiCPoT

Come si può notare, già solo dall'utilizzo di due canali di comunicazione, il sistema elettronico di bordo è stato sviluppato secondo il criterio di ridondanza in modo da garantirne il funzionamento anche in caso di un guasto.

Il progetto PiCPoT si è concluso con il lancio del 26 Luglio 2006 dalla base russa di Baikonur (KAZ) su un razzo vettore Dnepr-LV, di derivazione militare.

Purtroppo a causa di un problema idraulico del razzo vettore il lancio è fallito.

## 1.2 Il progetto AraMiS

Dopo l'esperienza di PiCPoT, nell'autunno del 2006, ha inizio un nuovo progetto: il progetto AraMiS, assonanza che sta per *Architettura Modulare per Satelliti*.

Alla base del progetto c'è l'utilizzo di un'architettura modulare, basata su un discreto numero di moduli che devono poter essere riutilizzati a seconda delle esigenze delle diverse missioni.

Il riutilizzo degli stessi moduli permette una diminuzione dei costi di realizzazione e allo stesso modo una riduzione dei tempi di sviluppo, che sono gli obiettivi principali.

Seguendo questa architettura, il satellite è costituito sulla superficie esterna da mattonelle standard di due tipi (Fig. 1.4):

- Power Management tile, che si occupa della gestione della potenza e del controllo d'assetto attivo.

Essa è costituita da un pannello solare esterno che genera l'energia da trasferire nella batteria, collocata all'interno della mattonella, per essere immagazzinata.

Per realizzare il controllo d'assetto vi sono un solenoide e una ruota d'inerzia che permettono di orientare il satellite intorno ai tre assi x, y, z.

Tutti i processi sono gestiti da un microcontrollore.

- Telecommunication tile, che si occupa di trasmettere e ricevere dati e comandi con la Stazione di Terra, realizzata a tale proposito, alle frequenze di 437 MHz e 2.4 GHz che si collocano nella banda dedicata alle comunicazioni satellitari amatoriali.

Principalmente è costituita da un transceiver controllato da un processore, un amplificatore di potenza per la trasmissione, un amplificatore a basso rumore per la ricezione e un sistema d'antenna.

All'interno (Fig. 1.5) le mattonelle possono essere di più tipi e per il momento quella in fase di sviluppo è la On-Board Processor and Payload Support.

L'On-Board Processor gestisce il funzionamento dell'intero sistema, controllando la comunicazione e i comandi relativi all'assetto del satellite, mentre il Payload potrà essere variato relativamente alle diverse missioni.

Gli obiettivi del progetto AraMiS rimangono gli stessi del progetto PiCPoT.

Questa tesi si occupa del progetto del sottosistema di comunicazione tra il Satellite e la Stazione di Terra alla frequenza di 2.4 GHz.

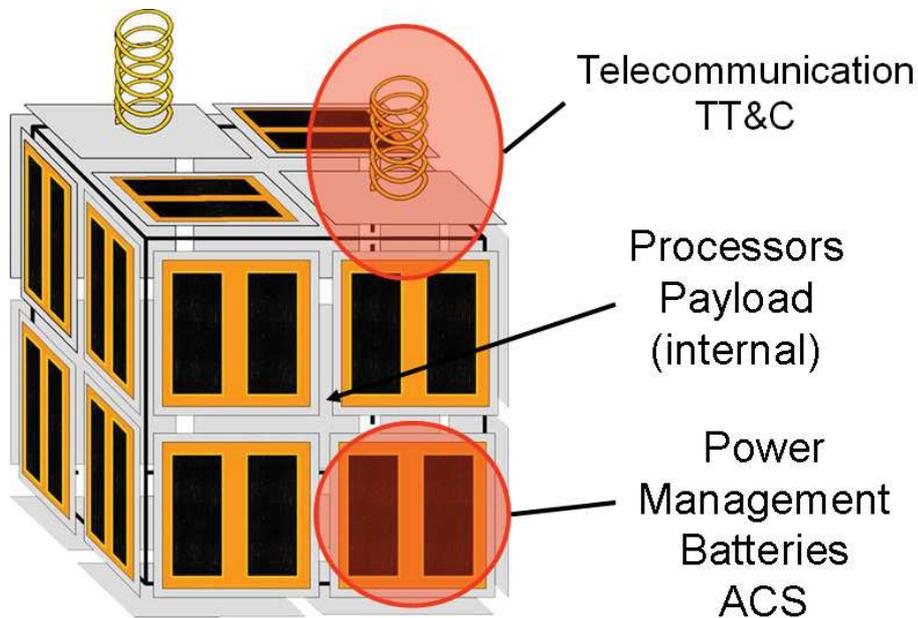


Figura 1.4. Vista esterna AraMiS



Figura 1.5. Vista interna AraMiS

# Capitolo 2

## I sistemi di comunicazione

L'argomento riguardante i sistemi di comunicazione è immenso, basta pensare al suo sviluppo storico che parte dal 3000 a.c., quando gli Egizi svilupparono il linguaggio grafico detto *Hieroglyphics*, ad oggi con l'era della trasmissione digitale.

I sistemi di comunicazione possono essere descritti con il diagramma a blocchi mostrato in figura Fig. 2.1.

Indipendentemente dalle particolari applicazioni in cui vengono usati, i sistemi di comunicazione si basano su tre sottosistemi importanti, che sono: *il Transmitter, lo Channel e il Receiver*.

Il segnale d'ingresso  $m(t)$ , che trasporta l'informazione, passa nel transmitter attraverso il blocco signal processing attraverso il quale viene modificato per avere una trasmissione più efficiente.

Per esempio, in un sistema analogico, il signal processing potrebbe essere un filtro passa-basso per restringere la larghezza di banda del segnale.

In un sistema ibrido, invece, potrebbe essere un convertitore analogico-digitale ADC che converte il segnale in una digital-word.

In questo caso il convertitore ADC fornisce la codifica di sorgente e in più il signal processing fornisce il controllo di parità sui bit, consentendo al ricevitore di rilevare eventuali errori.

All'uscita del signal processing il segnale è un segnale in banda base.

Passando poi attraverso il carrier circuit, il segnale viene convertito in una banda di frequenza appropriata per la trasmissione attraverso il canale.

Il carrier circuit è necessario se il canale di trasmissione è collocato in una banda di frequenza maggiore di zero.

In questo caso il segnale all'uscita del canale è un segnale in banda passante  $s(t) = R(t) \cos[\omega_c(t) + \theta(t)]$  e quindi un segnale *modulato*.

I canali possono essere classificati in due categorie: *wire e wireless*.

Alcuni esempi di canali wire sono: i cavi coassiali, le guide d'onda, i cavi in fibra ottica.

Invece alcuni tipici canali wireless sono: l'aria, il vuoto, il mare.

I principi generali della modulazione digitale e analogica si applicano a tutti i tipi di canali, sebbene le caratteristiche dei vari canali impongano costrizioni che favoriscono un particolare tipo di elaborazione del segnale.

In generale il canale attenua il segnale cosicchè il segnale d'ingresso non viene affetto dal rumore generato dal canale stesso o da un difetto del ricevitore.

Il canale può anche dare origine a *cammini multipli*. La propagazione per cammini multipli crea una diffusione ritardata del segnale ricevuto, il *delay spread*, e una modulazione di frequenza casuale, dovuta a diversi spostamenti, il fenomeno dello *effetto Doppler*.

Infine, il ricevitore converte il segnale d'uscita dal canale da banda passante in banda base, ossia lo ripulisce e fornisce una stima della sorgente d'informazione  $m(t)$  all'uscita del sistema di comunicazione.

Nei sistemi digitali, la misura d'errore del segnale ricevuto è detta *probability of bit error*  $P(e)$  chiamata anche *bit error rate* BER, mentre nei sistemi analogici questa misura è data dal rapporto

$$\frac{\text{segnale d'uscita}}{\text{rumore}}$$

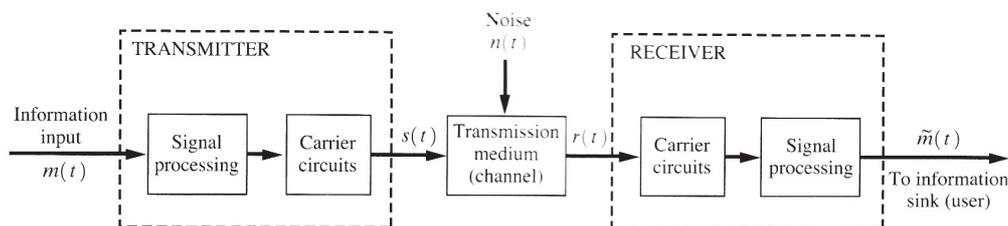


Figura 2.1. Communication system

I dati principali che caratterizzano un Trasmittitore sono:

- Tipo e Proprietà di Modulazione possibili
- Potenza d'uscita della portante (valore di regime e massimo)
- Banda di Frequenza in cui operare e Stabilità di Frequenza riferite alla portante
- Livello di Rumore
- Livello di potenza del segnale di ingresso per ottenere un'efficienza del 100 % sulla modulazione di frequenza, Banda di Frequenza del segnale considerato e Distorsione

- Rendimento.

Le principali caratteristiche di un Ricevitore sono:

- Sensibilità, la capacità di ricevere segnali deboli
- Selettività, la capacità di selezionare portante e bande laterali relative alla stazione desiderata e di attenuare il più possibile tutti gli altri segnali presenti in antenna
- Fedeltà, la capacità di riprodurre l'informazione senza introdurre distorsione
- Sensibilità alle Interferenze, la capacità di consentire l'intelligibilità delle informazioni, anche in presenza di disturbi e interferenze di vario genere.

## 2.1 Bande di frequenza

I sistemi di comunicazione wireless usano lo spazio libero come canale di trasmissione, nel quale l'interferenza e le condizioni di propagazione dipendono fortemente dalla frequenza.

Teoricamente, qualunque tipo di modulazione potrebbe essere usata a qualunque frequenza di trasmissione, ma per ridurre l'interferenza bisogna specificare il tipo di modulazione, la larghezza di banda, la potenza e il tipo d'informazione che può essere trasmessa su determinate bande di frequenza.

La frequenza assegnata e gli standard tecnici sono settati a livello internazionale dall'International Telecommunications Union (ITU), un'agenzia specializzata delle Nazioni Unite.

L'ITU è divisa in tre settori: il Radiocommunication Sector (ITU-R), che si occupa dei valori di frequenza da assegnare e di garantire un uso efficiente della radiofrequenza; la Telecommunications Standardization Section (ITU-T), che gestisce i problemi tecnici ed economici e in più raccomanda gli standard universali per la rete pubblica di telecomunicazione (PTN) e per i relativi sistemi radio; e il Telecommunication Development Sector (ITU-D), che provvede all'assistenza tecnica soprattutto a livello di sviluppo nazionale, consentendo a molti servizi di telecomunicazione di poter essere supportati economicamente e integrati in un sistema di telecomunicazione universale.

I valori delle frequenze internazionali assegnate sono suddivisi in sottobande dalla Federal Communications Commission (FCC), l'agenzia degli Stati Uniti che si occupa dei sistemi radio.

Come si può vedere dalla tabella sottostante in Fig 2.2 e Fig 2.3 per ogni banda di frequenza è riportata la sua denominazione, le sue caratteristiche di propagazione e i rispettivi usi tipici di quella frequenza.

TABLE 1-2 FREQUENCY BANDS

Frequency Band <sup>a</sup>	Designation	Propagation Characteristics	Typical Uses
3–30 kHz	Very low frequency (VLF)	Ground wave; low attenuation day and night; high atmospheric noise level	Long-range navigation; submarine communication
30–300 kHz	Low frequency (LF)	Similar to VLF, slightly less reliable; absorption in daytime	Long-range navigation and marine communication radio beacons
300–3000 kHz	Medium frequency (MF)	Ground wave and night sky wave; attenuation low at night and high in day; atmospheric noise	Maritime radio, direction finding, and AM broadcasting
3–30 Mhz	High frequency (HF)	Ionospheric reflection varies with time of day, season, and frequency; low atmospheric noise at 30 Mhz	Amateur radio; international broadcasting, military communication, long-distance aircraft and ship communication, telephone, telegraph, facsimile
30–300 MHz	Very high frequency (VHF)	Nearly line-of-sight (LOS) propagation, with scattering because of temperature inversions, cosmic noise	VHF television, FM two-way radio, AM aircraft communication, aircraft navigational aids
0.3–3 GHz	Ultrahigh frequency (UHF)	LOS propagation, cosmic noise	UHF television, cellular telephone, navigational aids, radar, GPS, microwave links, personal communication systems
	<i>Letter designation</i>		
1.0–2.0	L		
2.0–4.0	S		
3–30 GHz	Superhigh frequency (SHF)	LOS propagation; rainfall attenuation above 10 GHz, atmospheric attenuation because of oxygen and water vapor, high water vapor absorption at 22.2 GHz	Satellite communication, radar microwave links
	<i>Letter designation</i>		
2.0–4.0	S		
4.0–8.0	C		
8.0–12.0	X		
12.0–18.0	Ku		
18.0–27.0	K		
27.0–40.0	Ka		
26.5–40.0	R		
30–300 GHz	Extremely high frequency (EHF)	Same; high water-vapor absorption at 183 GHz and oxygen absorption at 60 and 119 GHz	Radar, satellite, experimental

<sup>a</sup> kHz = 10<sup>3</sup> Hz; MHz = 10<sup>6</sup> Hz; GHz = 10<sup>9</sup> Hz.

Figura 2.2. Frequency Band

TABLE 1-2 FREQUENCY BANDS (cont.)

Frequency Band <sup>a</sup>	Designation	Propagation Characteristics	Typical Uses
<i>Letter designation</i>			
27.0–40.0	Ka		
26.5–40.0	R		
33.0–50.0	Q		
40.0–75.0	V		
75.0–110.0	W		
110–300	mm (millimeter)		
10 <sup>3</sup> –10 <sup>7</sup> GHz	Infrared, visible light, and ultraviolet	LOS propagation	Optical communications

<sup>a</sup> kHz = 10<sup>3</sup> Hz; MHz = 10<sup>6</sup> Hz; GHz = 10<sup>9</sup> Hz.

Figura 2.3. Frequency Band

## 2.2 Caratteristiche della propagazione delle onde a radiofrequenza

Le caratteristiche di propagazione delle onde elettromagnetiche usate nei canali wireless dipendono fortemente dalla frequenza.

Esse sono il risultato dei cambiamenti di velocità dell'onda radio, funzione dell'altitudine e delle condizioni al contorno.

La velocità dell'onda dipende dalla temperatura, dalla densità e dai livelli di aria ionizzata (la ionosfera).

La ionizzazione è causata dalla radiazione dei raggi ultravioletti generati dal Sole, tanto quanto dai raggi cosmici.

Di conseguenza l'aumento di ionizzazione dipende dall'ora del giorno, dalla stagione e dall'attività del Sole. Questo risulta dalle diverse regioni in cui si suddivide la ionosfera(Fig 2.4):

La *regione D* compresa fra 50 Km e 90 Km rispetto alla densità elettronica, che in questo caso è bassa.

Qui la ionizzazione è causata dal Sole e si presenta soprattutto di giorno.

Questa regione consente la riflessione delle onde comprese fra 30 kHz e 300 kHz, che si propagano come in una guida d'onda fra suolo e regione D, mentre le frequenze più alte attraversano la regione ma sono poco attenuate.

La *regione E* compresa fra circa 90 Km e 140 Km rispetto alla densità elettronica, che in questo caso è alta. Qui la ionizzazione è causata dai raggi ultravioletti e dai raggi X provenienti dal Sole.

Questa regione permette la riflessione delle onde elettromagnetiche di giorno, nella banda compresa tra 3 MHz e 30 MHz, e di notte nella banda compresa tra 30 kHz e 3000 kHz.

All'interno si può presentare uno strato detto *strato E sporadico* che è caratterizzato da una densità elettronica più alta di quella della regione E e che si presenta, appunto, in modo sporadico per vari motivi, come ad esempio le perturbazioni elettromagnetiche.

La regione F va oltre i 140 Km ed è costituita da due strati F1 e F2.

Lo strato F1 è compreso fra circa 180 Km e 230 Km, quindi da un'alta densità elettronica, e qui la ionizzazione è causata maggiormente dal Sole; di notte i due strati F1 e F2 si fondono.

Lo strato F2 è compreso fra circa 230 Km e 400 Km, quindi anch'esso da un'alta densità elettronica causata dall'attività del Sole, dalla località, dall'ora del giorno e dalla stagione.

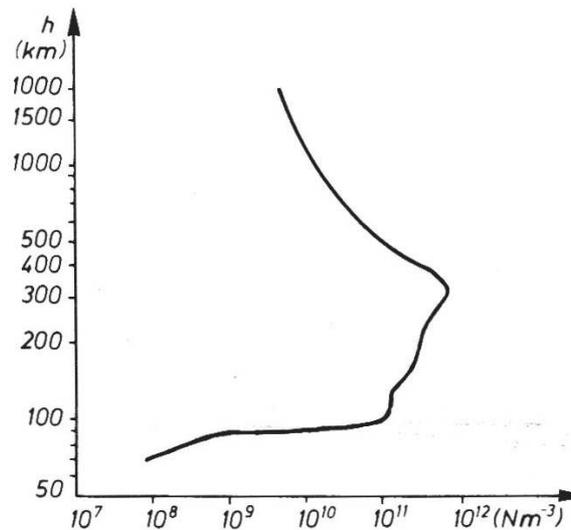


Figura 2.4. Densità di elettroni nella ionosfera

Le principali caratteristiche di propagazione sono: *il ground-wave, lo sky wave e il line of sight(LOS)* (FIG. 2.5).

Il Ground-wave è il modo di propagazione dominante per frequenze sotto i 2 MHz.

Esso comprende la banda VLF (*Very Low Frequency*), la cui frequenza va da 3 kHz a 30 kHz e la banda LF (*Low Frequency*), la cui frequenza va da 30 kHz a 300 kHz.

Qui la propagazione avviene per onda superficiale.

Queste onde trovano applicazioni nelle comunicazioni con i sommergibili, nelle esplorazioni geofisiche e nei sistemi di navigazione iperbolici per le condizioni costanti di propagazione, rese possibili dalla bassa frequenza.

Un limite è l'elevato rumore atmosferico che disturba la ricezione.

Tra il Ground-wave e lo Sky-wave c'è la banda di frequenza MF (*Medium Frequencies*), che va da 300 kHz a 3 MHz all'interno della quale la propagazione avviene per onda superficiale e per onda ionosferica, a seconda delle comunicazioni diurne o notturne.

Lo Sky-wave è il modo di propagazione per frequenze che vanno da 2 MHz a 30 MHz e comprende la banda HF (*High Frequency*), la cui frequenza va da 3 MHz a 30 MHz.

La propagazione avviene per onda ionosferica; le onde attraversano le regioni D ed E, che assorbono parte dell'energia irradiata e sono riflesse nella regione F.

Le riflessioni, semplici o multiple, e la bassa attenuazione ionosferica alle alte frequenze consentono la ricezione in punti molto distanti dal trasmettitore.

Data la distanza tra il trasmettitore e il ricevitore, esiste, a seconda dello stato della ionosfera, una banda di frequenza ottima di collegamento il cui limite superiore è detto MUF, ossia massima frequenza usabile.

Per le comunicazioni a lunga distanza, in genere, si preferiscono frequenze alte di giorno (intorno ai 15 MHz) e frequenze basse di notte (inferiori a 10 MHz).

A queste frequenze non si hanno delle buone caratteristiche di stabilità, in quanto le condizioni di propagazione dipendono fortemente dall'attività solare, quindi sono variabili e ciò richiede cambiamenti di frequenza di trasmissione, inoltre, le onde possono arrivare al ricevitore attraverso vari cammini e ciò può causare affievolimenti (fading).

Il line of sight (LOS) è il modo di propagazione dominante per frequenze oltre i 30 MHz.

Esso si suddivide in due bande di frequenza: la VHF (*Very High Frequencies*), con frequenze comprese tra 30 MHz e 300 MHz, la UHF (*Ultra High Frequencies*), con frequenze comprese tra 300 MHz e 3 GHz.

In questo caso la propagazione avviene per onda troposferica, in quanto la ionosfera alle alte frequenze non consente la riflessione.

Le comunicazioni sono possibili tramite onda diretta e riflessa, per diffusione, per scattering, ma non tramite onda superficiale, perché a queste frequenze si attenuerebbe velocemente.

Questo modo è usato nelle trasmissioni televisive, nei radar, nelle comunicazioni satellitari, nelle comunicazioni tra punti fissi e punti mobili.

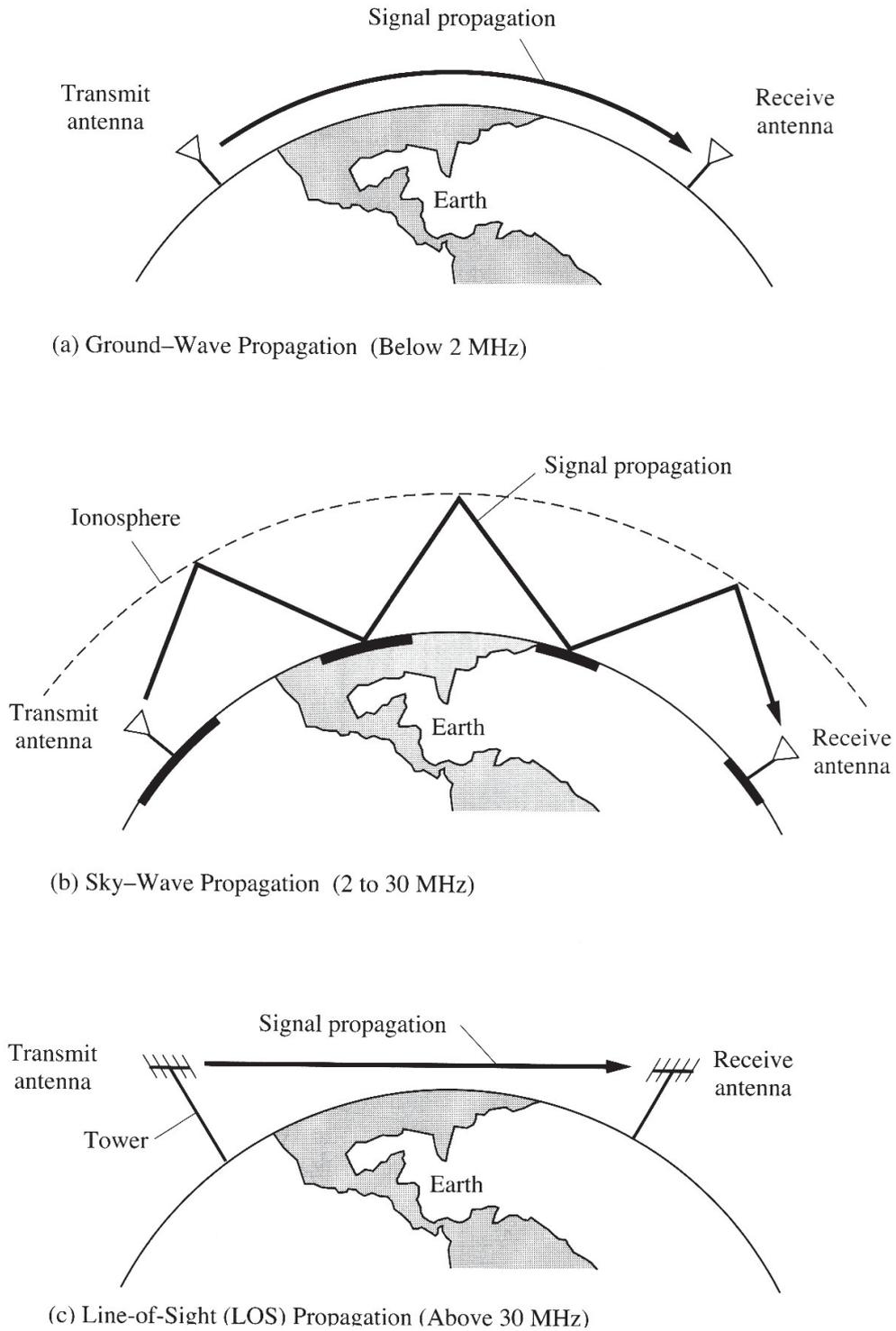


Figura 2.5. Propagation of radio frequencies

## 2.3 Sistemi di comunicazione satellitare

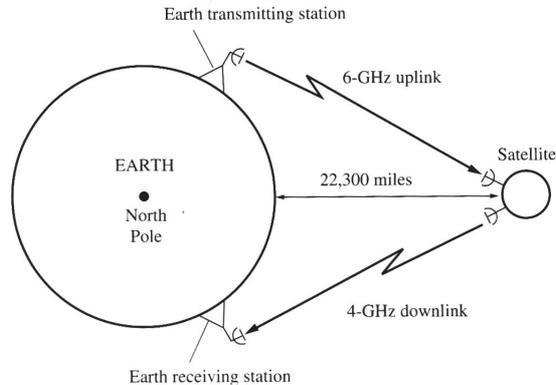


Figura 2.6. Communications satellite in geosynchronous orbit

Le prime comunicazioni satellitari risalgono agli inizi degli anni '60, anni in cui venivano usati i satelliti in orbita LEO, tra cui ricordiamo: i satelliti *Echo* (1960), *Telstar* (1962) e *Relay* (1962).

Il Satellite fungeva da *cavo nel cielo* per connettere due punti sulla superficie della Terra molto distanti tra loro.

La sua potenzialità veniva dimostrata in particolar modo nella trasmissione di segnali televisivi oltre Oceano, quando ancora non esistevano alternative per questo tipo di servizio.

Il progresso dei sistemi di comunicazione è stato largamente supportato dall'utilizzo dei *satelliti geostazionari*, cioè satelliti che descrivono un'orbita circolare equatoriale a 35.786 km dalla superficie terrestre e localizzati in modo da avere lo stesso periodo della Terra per cui appaiono fermi.

Questa caratteristica permette la semplificazione delle antenne della Stazione di Terra, in quanto esse puntano verso una direzione fissa.

I primi satelliti GEO furono: *Syncom* (1963) e *Early Bird* (1965), detto anche *INTELSAT-I*, perchè fu il primo satellite commerciale usato dall'INTELSAT.

L'INTELSAT, l'Internal Telecom Satellite Consortium fu creata nel 1964 come la prima organizzazione intergovernativa mondiale per lo sviluppo e la gestione commerciale delle telecomunicazioni via satellite. La tecnologia spaziale in uso a quei tempi portava alla realizzazione di satelliti semplici e di piccole dimensioni, contrariamente alla Stazione di Terra, che doveva essere di grandi dimensioni, complessa e costosa.

Le Stazioni di Terra che venivano usate per le prime comunicazioni transatlantiche erano costituite da un sistema d'antenne di tipo parabolico, con un diametro compreso tra 25 e 32 m, un LNA, *low-noise amplifier*, per amplificare il segnale così

da non essere influenzato dal rumore e poter essere demodulato, e da un HPA, *high-power amplifier*, per fornire la potenza sufficiente a garantire la comunicazione.

Questo tipo di radiofrequenza veniva standardizzato dall'INTELSAT e chiamato standard A; per diversi anni solo le Stazioni di Terra che rispettavano questo standard venivano usate nel sistema INTELSAT.

Le bande di frequenza maggiormente considerate per i sistemi di comunicazione satellitare sono: 6 GHz per la comunicazione Terra-Satellite (*uplink*) e 4 GHz per la comunicazione Satellite-Terra (*downlink*).

In questo range di frequenza l'attenuazione è bassa così come il rumore, quindi le Stazioni di Terra devono essere localizzate in aree ben protette dalle interferenze terrestri.

Ogni satellite ha un certo numero di *transponders* (Fig. 2.7) a bordo per amplificare il segnale ricevuto dal collegamento uplink e convertirlo ad una frequenza più bassa per il collegamento downlink.

La larghezza di banda è comunemente 500 MHz ed è divisa in 12 bande di 36 MHz, con un intervallo di separazione tra una banda e un'altra; e ogni banda da 36 MHz è divisa in più canali in base alla caratteristica di comunicazione quali i canali telefonici, la trasmissione di dati e la TV.

Tipicamente un satellite ha 24 transponders e usa la stessa frequenza, utilizzando per 12 transponders la polarizzazione orizzontale e per gli altri 12 la polarizzazione verticale (Fig. 2.8).

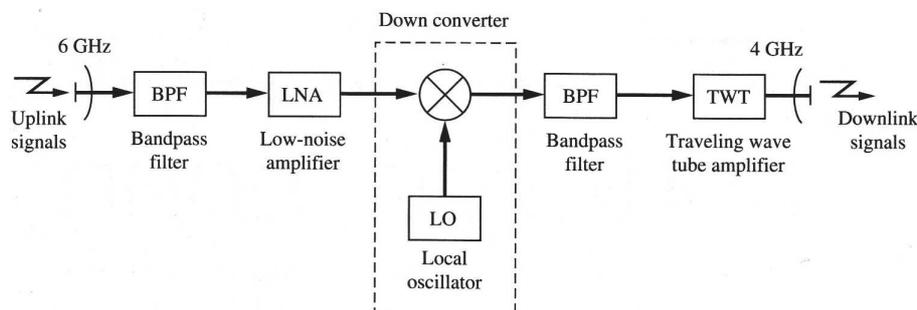


Figura 2.7. Simplified block diagram of a communications satellite transponder

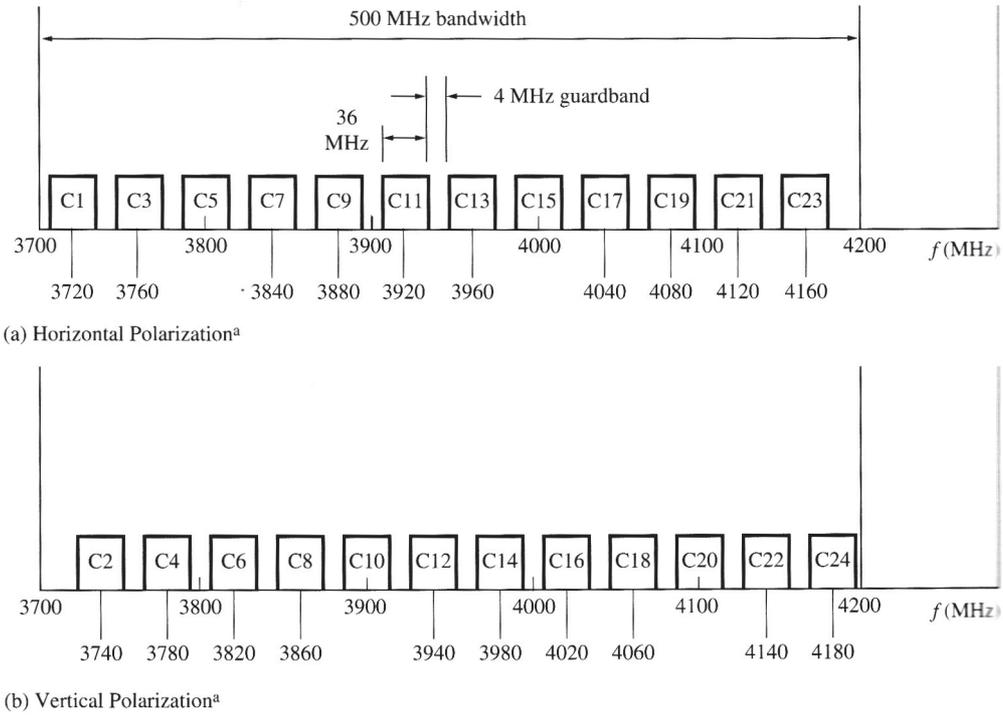


Figura 2.8. Simplified block diagram of communications satellite trasponder

## Capitolo 3

# Il transceiver CC2510

Il progetto del sottosistema di comunicazione a 2.4 GHz del satellite AraMiS parte con la scelta del transceiver da utilizzare: *il CC2510*.

Esso è un dispositivo della *Chipcon, TI* a basso costo, per applicazioni di comunicazioni wireless a bassa potenza e bassa tensione.

Le sue caratteristiche principali sono:

- Sintesi delle alte prestazioni del transceiver a radiofrequenza con il microcontrollore, basato sull'architettura dell'8051 enhanced
- Possibilità di programmare la memoria Flash a blocchi di 8,16,32 kB
- Banda di frequenza compresa tra (2400.0 ÷ 2483.5) MHz
- 1,2,4 kB di memoria Ram
- Tensione di alimentazione compresa tra (2.0 ÷ 3.6) V
- Alta sensitivity (−103 dBm a 2.4 kbps)
- Possibilità di programmare il Data Rate oltre 500 kbps
- Basso consumo di corrente (22 mA in ricezione, 23 mA in trasmissione) con un microcontrollore funzionante a 26 MHz
- Possibilità di programmare la Potenza d'uscita fino a 1 dBm per tutte le frequenze supportate
- 2 Interfacce Usart, 4 Timer
- un ADC da 14 bit e un Sensore di temperatura interno.

Il CC2510 è suddiviso principalmente in tre moduli connessi da un bus interno (SFR): la CPU, la Radio e il modulo relativo alla gestione della potenza e alla distribuzione del clock.

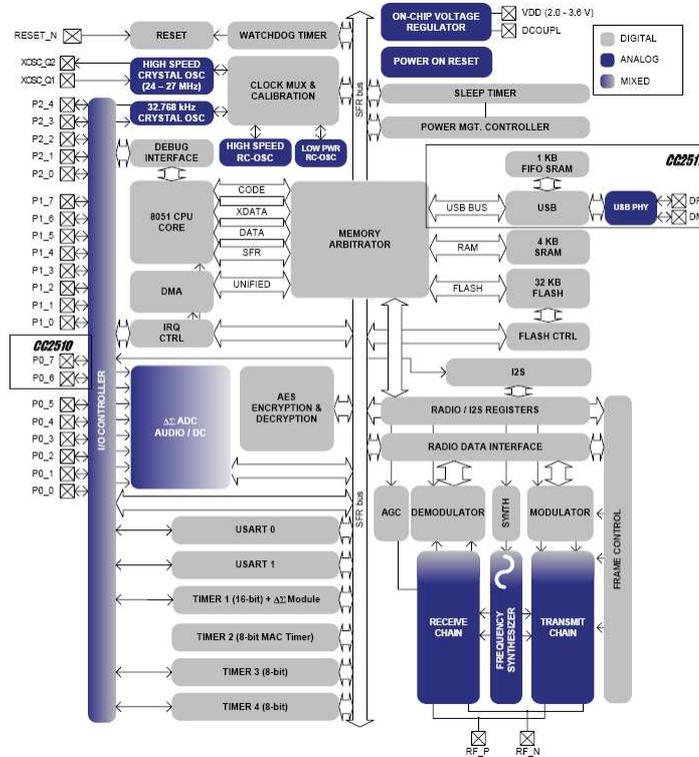


Figura 3.1. Diagramma a blocchi del CC2510

### 3.1 L'8051 CPU

La CPU utilizzata è una CPU a 8 bit compatibile con l'8051, ma con istruzioni che vengono eseguite più velocemente, per i seguenti motivi:

- viene usato un solo ciclo di clock per un ciclo di istruzioni a differenza dello standard 8051 che ne usa 12
- vengono eliminati gli stati ridondanti di bus.

Inoltre, l'architettura 8051 enhanced prevede un doppio puntatore ai dati per una maggiore velocità nello spostamento di questi ultimi da e verso la memoria, ed un ampliamento della gamma dei segnali di interrupt fino a 18.

La memoria si suddivide in 4 spazi:

- Code, uno spazio a 16 bit in sola lettura, in cui viene salvato il codice per programmare la memoria
- Data, uno spazio a 8 bit accessibile in lettura e scrittura da una singola istruzione CPU (i 128 byte meno significativi accessibili direttamente o indirettamente, i 128 byte più significativi accessibili solo indirettamente)
- XData, uno spazio di memoria dati a 16 bit accessibile in lettura e scrittura. L'accesso richiede 4-5 cicli di istruzione CPU, di conseguenza è più lento di quello a Data o Code, in quanto questa memoria condivide un bus con la CPU
- SFR, un registro di memoria a 7 bit a cui può accedere singolarmente un'istruzione CPU.

Questi spazi di memoria sono distinti nell'architettura 8051, ma al contrario nel CC2510 sono parzialmente sovrapposti per facilitare i trasferimenti in DMA e l'operazione di debug in hardware.

Viene riportata la relativa mappatura.

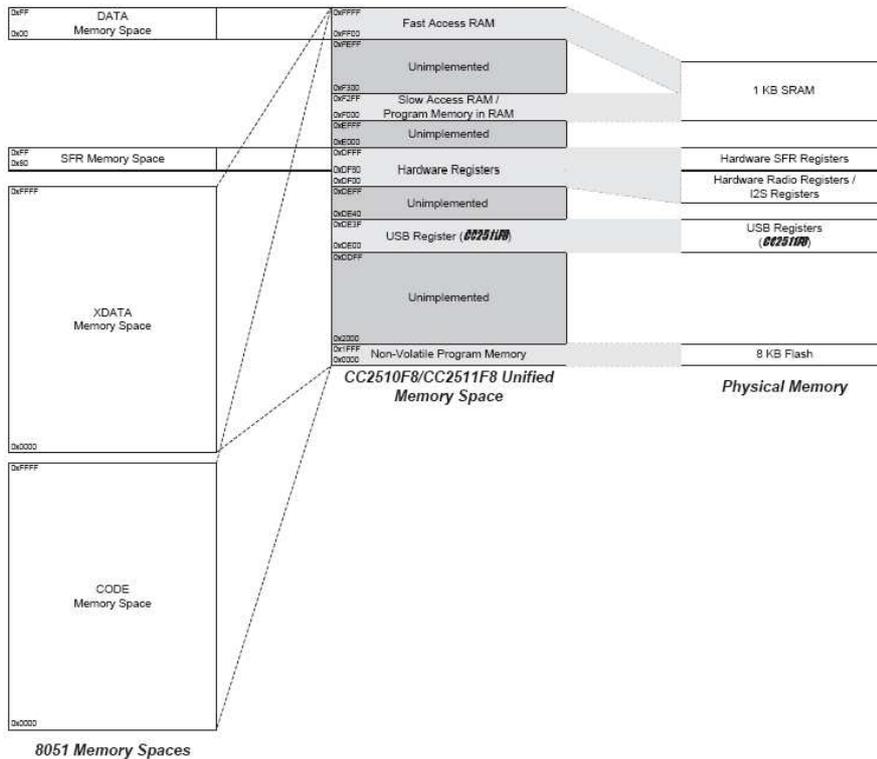


Figura 3.2. Mappa di memoria

## 3.2 La radio

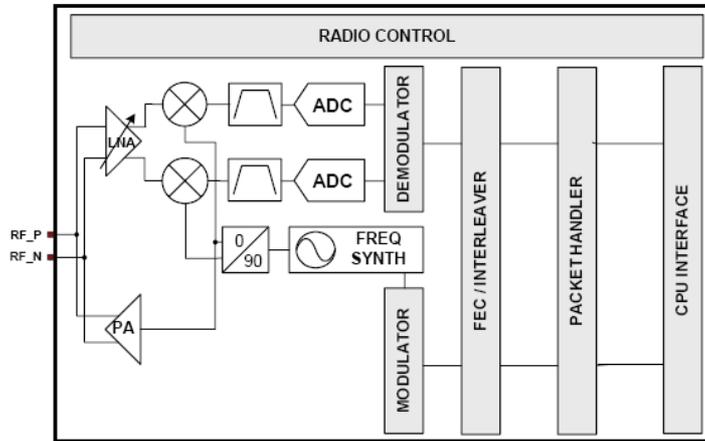


Figura 3.3. La radio

La radio è caratterizzata da un ricevitore *low-if*, detto anche ricevitore a *supereterodina*.

Il segnale radio ricevuto viene amplificato dal preamplificatore, LNA, successivamente la sua frequenza viene convertita in fase ( $I$ ) e in quadratura ( $Q$ ) ad una frequenza fissa detta frequenza intermedia ( $IF$ ).

In questo modo l'amplificazione del segnale avviene ad una frequenza più bassa consentendo un guadagno maggiore e una sensibilità più alta.

Il segnale viene poi digitalizzato attraverso il convertitore ADC e, effettuato il controllo automatico del guadagno (AGC) per garantire una buona demodulazione del segnale, viene demodulato e sincronizzato relativamente a bit o a pacchetto.

Il trasmettitore è basato su una sintesi diretta della frequenza RF e da un amplificatore di potenza, PA, che porta il segnale al livello di potenza desiderato per essere trasmesso.

Vi è anche un sintetizzatore di frequenza, che include a bordo del chip un VCO di tipo LC e un traslatore di fase a 90 gradi per generare i segnali  $I$  e  $Q$  per la conversione in modalità ricezione.

La frequenza di riferimento del sintetizzatore viene generata da un oscillatore a 26 o 48 MHz, il quale genera anche i cicli di clock relativi alla parte digitale per la configurazione del pacchetto e la bufferizzazione.

Esso supporta le modulazioni 2-FSK (*Frequency Shift Keying*), GFSK (*Gaussian Frequency Shift Keying*), MSK (*Minimum Shift Keying*) e non comporta overhead per la CPU.

Per quanto riguarda il modulo relativo alla gestione della potenza e al sistema di clock vengono usati 4 modi flessibili di potenza per ridurre il consumo.

Di seguito(Fig. 2.5) viene riportata la tabella relativa a questi 4 modi, PM0, PM1, PM2, PM3 rispetto all'alimentazione e all'oscillatore utilizzato.

Quando la radio è attiva viene usato l'oscillatore a 26 MHz.

Operating Mode	High-speed Oscillator		Low-speed Oscillator		Digital Voltage Regulator	CPU
Configuration	A	None	A	None		
	B	High speed XOSC	B	Low power RCOSC		
	C	HS RCOSC	C	32.768 kHz XOSC		
Active	B and / or C		B or C		On	Running
PM0	B and / or C		B or C		On	Idle
PM1	A		B or C		On	Idle
PM2	A		B or C		Off	Idle
PM3	A		A		Off	Idle

Figura 3.4. Modi operativi di Potenza

### 3.3 Evaluation module del CC2510

L'evaluation module del CC2510 (Fig. 3.6) è stata realizzata utilizzando lo schema della *Texas Instrument* con opportune modifiche per eliminare i disturbi.

Le modifiche hanno comportato: l'aggiunta di un filtro passa-alto, costituito da una resistenza pari a 10 k $\Omega$  e una capacità da 100 pF sulla *Debug\_Clock*, e l'aggiunta di una resistenza da 10 k $\Omega$  tra il *reset* e l'alimentazione (si veda a proposito lo schema riportato in Fig. 3.7).

Dopo la realizzazione hardware si è passati all'esecuzione del codice di programmazione da scrivere nella *memoria flash* del CC2510 per fornirgli le istruzioni atte al suo funzionamento.

Per lo sviluppo del software si è scelto di utilizzare una piattaforma hardware esterna, il CC1010 già precedentemente testato, che andasse a pilotare l'interfaccia di programmazione del CC2510.

Questa scelta è stata dettata dalla verificata compatibilità a livello di istruzioni dei due transceiver.

L'Evaluation Board del CC1010 è costituita, come si può vedere, principalmente da: due porte seriali, un'interfaccia parallela, LEDs, buttons, un regolatore di tensione, jumpers e connettori di configurazione per permettere al CC1010 di interfacciarsi con altri dispositivi esterni(Fig 3.5).

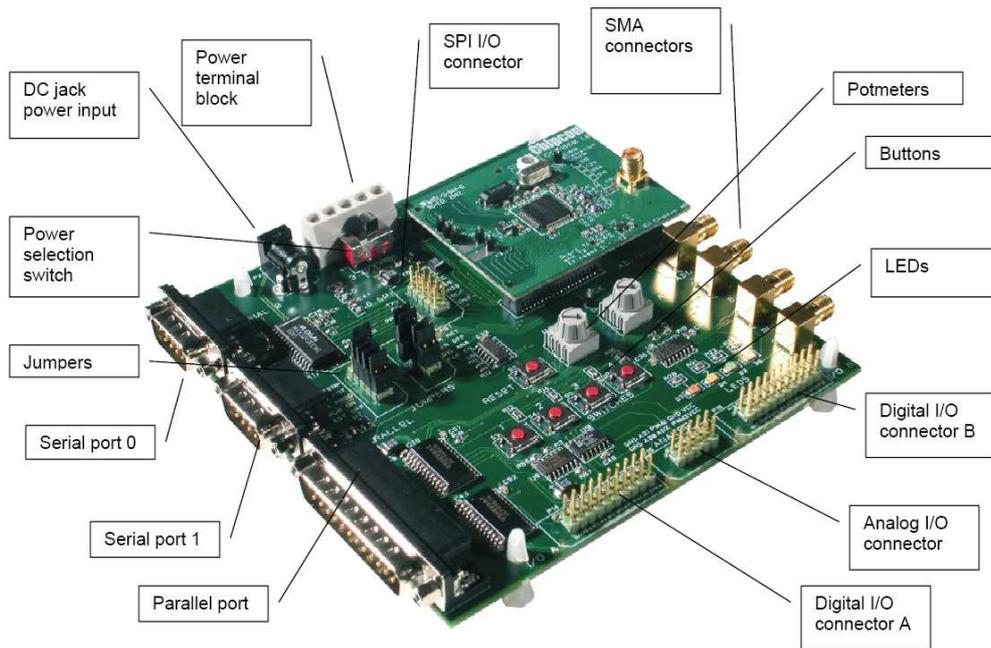


Figura 3.5. CC1010EB evaluation board

In questo modo il firmware presente a bordo del CC1010 consente la comunicazione con il PC attraverso la seriale Rs-232 e la visualizzazione del set di operazioni disponibili tramite un'opportuna interfaccia grafica, Tera Term Pro.

Mediante un adeguato protocollo di comunicazione, il protocollo Xmodem, il codice risultante dalla compilazione del sorgente, ovvero il File.hex, viene caricato nella memoria del CC1010 e in seguito attraverso un'opzione dell'interfaccia grafica è possibile programmare il CC2510 attraverso la Debug interface.

A causa della limitazione nella funzione di buffering del CC1010, che permette il caricamento dei dati al massimo di circa 2 kB, si è reso necessario segmentare e inviare in più programmazioni successive il file.hex.

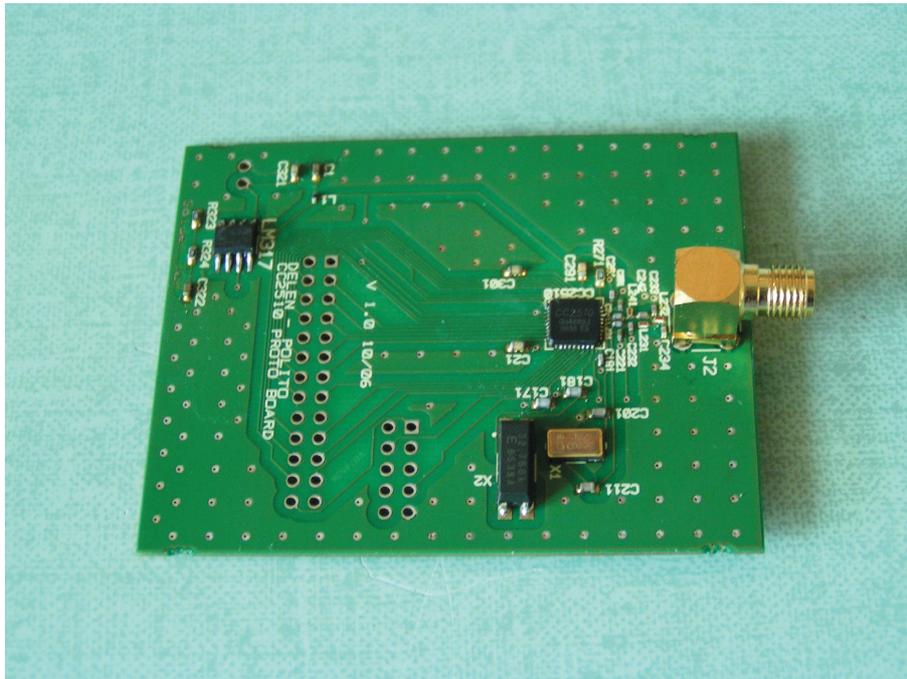


Figura 3.6. Evaluation module CC2510

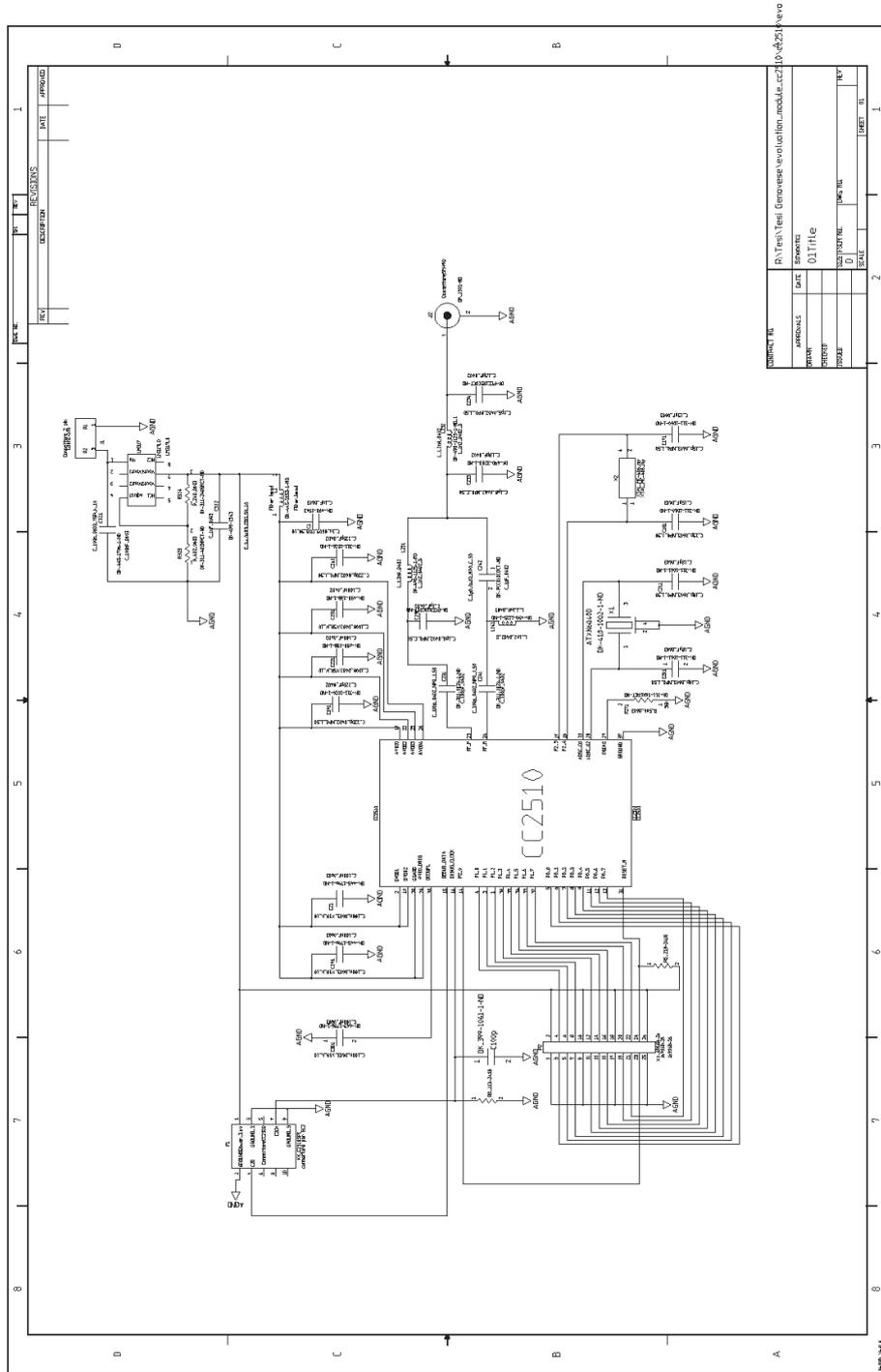


Figura 3.7. Schema elettrico CC2510

### 3.4 Il protocollo xmodem

Il protocollo di comunicazione Xmodem è stato sviluppato nel 1977 da *Ward Christensen* e risulta ancora oggi il protocollo più usato, perché è l'unico comune a diversi programmi di comunicazione.

Si tratta di un semplice protocollo ARQ (*Automatic Repeat Request*), che prevede la ritrasmissione automatica del blocco di dati quando si verifica un errore.

Il pacchetto di dati è suddiviso in 5 campi, di cui un campo dati di lunghezza pari a 128 byte e 4 campi di controllo di lunghezza pari a 1 byte.

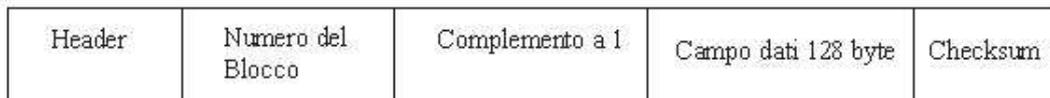


Figura 3.8. Pacchetto dati

in cui:

- il campo Header contiene un carattere detto SOH (start of header), che identifica l'inizio di un blocco dati
- il campo Numero del blocco contiene il numero del blocco corrente
- il campo Complemento a 1 contiene il complemento a 1 della combinazione binaria del numero del blocco corrente, detto anche inversione bit a bit
- il campo Payload contiene i dati
- il campo Crc contiene il controllo di parità sui bit calcolato, sommando in modulo 256 i 128 byte di dato.

La procedura di trasferimento dei dati tra una Stazione Trasmittente (Tx) e una Stazione Ricevente (Rx) è la seguente:

1. il ricevitore invia un carattere Nack (*not-acknowledgement*) al trasmettitore
2. finché il trasmettitore non invia un dato il ricevitore per un certo tempo di circa 10 secondi continua ad inviare un Nack
3. quando il trasmettitore invia un dato si mette in attesa di ricevere:

- (a) un Ack (*acknowledgement*) se il dato ricevuto è corretto e di conseguenza invia il blocco successivo
  - (b) un Nack se il carattere identificativo SOH , o il numero del blocco corrente o il crc non è corretto e di conseguenza ritrasmette il blocco
  - (c) un Can se il pacchetto viene perso e di conseguenza il trasferimento termina senza condizioni
4. la procedura continua fino all'ultimo blocco
  5. alla fine del trasferimento dell'intero pacchetto il trasmettitore invia un Eot (*end of trasmission ASCII*), il ricevitore invia in risposta un Ack e la procedura termina.

Le funzioni utilizzate nella creazione della libreria Xmodem.h sono state di conseguenza:

- `void xmodem_memcpy(unsigned char *dest, unsigned char *src, unsigned char block_num)`, che copia i pacchetti ricevuti attraverso il protocollo in un buffer, controllando che il numero del blocco di ogni pacchetto sia corretto
- `unsigned short xmodem_receive(unsigned char *addr)`, che riceve il file attraverso il protocollo Xmodem e lo immagazzina in un buffer puntato dalla variabile ADDR.

## 3.5 La debug interface

Il CC2510 è costituito da un'interfaccia seriale sincrona doppio-cablata, simile ad una SPI (*Serial peripheral interface*) detta *Debug Interface*. Attraverso questa interfaccia è possibile sia programmare la memoria flash del CC2510 che permettere altre funzionalità, quali l'accesso ai contenuti della memoria dei registri, il controllo dell'abilitazione degli oscillatori e l'esecuzione di istruzioni fornite sulla CPU 8051, implementando le funzioni di un simil-emulatore (breakpoint e step by step execution).

La debug interface utilizza un pin dato bidirezionale detto *Debug-Data* (P2\_1) e un pin di clock detto *Debug-Clock* (P2\_2), quando è nella modalità di *Debug* e il *Reset\_N* è a livello alto pari a 1, altrimenti questi pin sono usati come generici pin di Input\Output.

Per entrare nella *Debug-mode* bisogna forzare 2 transizioni alte del Debug-Clock mentre il *Reset\_N* si trova al livello basso pari a 0.

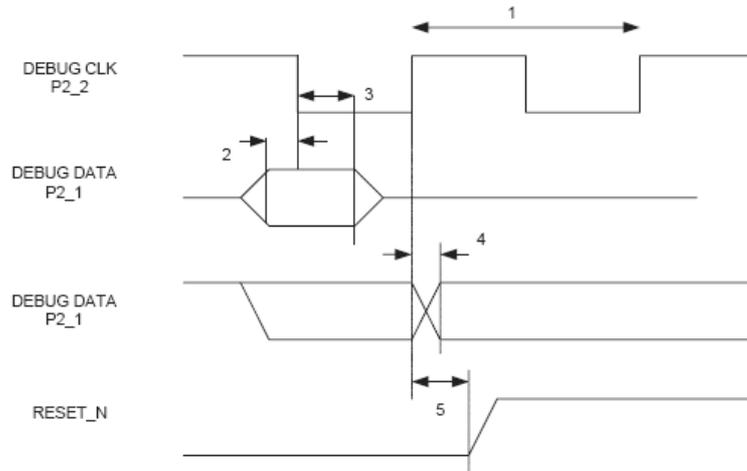


Figura 3.9. Caratteristiche della Debug Interface

I dati vengono caricati sul pin di Debug Data durante il fronte di salita del clock e campionati sul fronte di discesa.

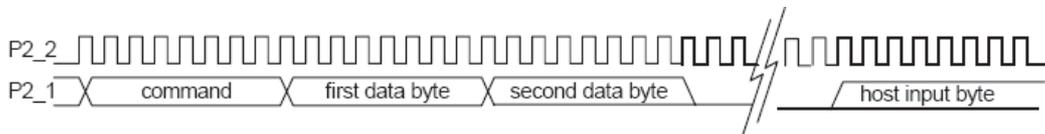


Figura 3.10. Diagramma temporale

I comandi di Debug vengono inviati da un host esterno ed hanno una lunghezza che va da 1 a 4 byte, includendo il byte di comando più un byte opzionale letto dall'host.

I Comandi usati nella programmazione del CC2510 sono stati principalmente:

- Chip\_Erase, che cancella l'intera memoria flash, includendo la protezione dei bit

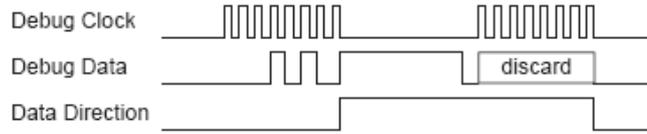


Figura 3.11. caratteristiche del comando Chip Erase

- Read\_Status, che arbitra lo stato dei chip della flash cancellati dopo il comando chip\_erase oppure viene usato quando è richiesto uno stato stabile dell'oscillatore per debuggare altri comandi(

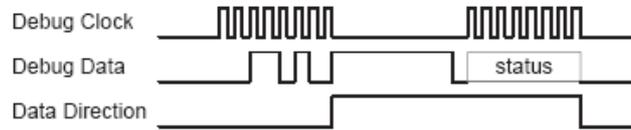


Figura 3.12. Caratteristiche del comando Read Status

- Get\_Chip\_Id, che richiede l'invio di due byte identificativi del dispositivo e per il CC2510 sono i valori esadecimali 0x81 e 0x03

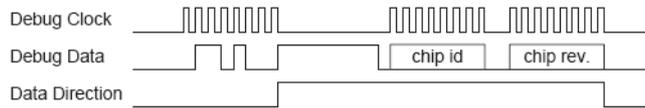


Figura 3.13. Caratteristiche del comando Get Chip Id

- Halt, che arresta la CPU dall'esecuzione del programma

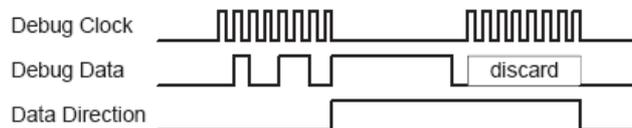


Figura 3.14. Caratteristiche del comando Halt

- Resume, che abilita la CPU ad eseguire il programma

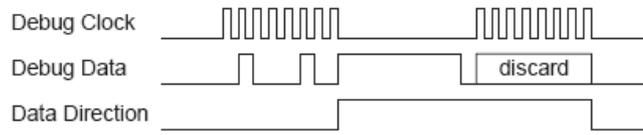


Figura 3.15. Caratteristiche del comando Resume

- Debug\_Instr, che può eseguire 1, 2, 3 byte di istruzioni dell'8051 sulla CPU senza cambiare il program counter.

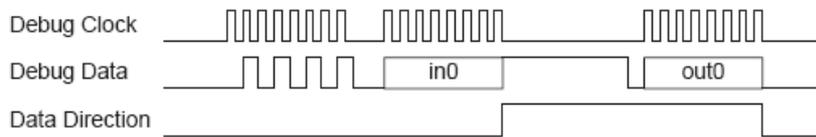


Figura 3.16. Caratteristiche Debug Instr (1 byte)

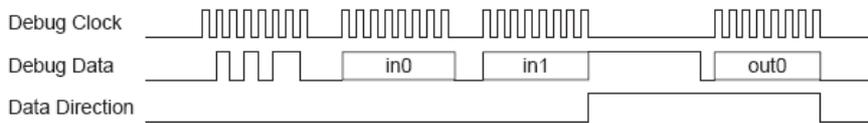


Figura 3.17. Caratteristiche Debug Instr (2 byte)

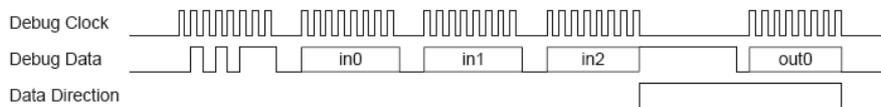


Figura 3.18. Caratteristiche Debug Instr (3 byte)

Per una sicurezza del software la debug interface deve essere protetta e questa modalità è detta *Debug Lock Bit*.

In questa modalità tutti i comandi sono disabilitati, eccetto i comandi `CHIP_ERASE`, `READ_STATUS` e `GET_CHIP_ID`.

Le funzioni utilizzate per creare la libreria `Debug.h` sono state:

- `unsigned char char2int(unsigned char dato)`, che converte un carattere in un intero
- `unsigned char int2char(unsigned char dato)`, che converte un intero in un carattere
- `void Port_init()`, che configura i pin di dato, il clock e il reset
- `void Debug_init()`, che permette l'accesso alla modalità di Debug
- `void Debug_command (unsigned char command)`, che invia il comando richiesto attraverso la Debug Interface
- `unsigned char Debug_Receive()`, che riceve la risposta del chip attraverso la Debug Interface
- `void Debug_EraseFlash()`, che cancella la memoria flash del CC2510
- `unsigned char Debug_ReadFlash(unsigned short addr)`, che legge un byte all'indirizzo richiesto nella flash
- `unsigned char Debug_ReadRAM(unsigned short addr)`, che legge un byte all'indirizzo richiesto nella RAM
- `void Debug_WriteRAM(unsigned short addr, unsigned char dato)`, che scrive un byte di dato all'indirizzo puntato dalla variabile `addr`
- `unsigned short Debug_ProgramLoad(unsigned char xdata *hex)`, che decodifica il file `hex` puntato dalla variabile `hex` e lo scrive nella RAM
- `void Debug_BootLoad(unsigned char *addr, unsigned short num)`, che copia nella RAM la funzione `Ram2Flash`
- `void Debug_LoadPC(unsigned short PC)`, che carica il program counter da dove si era interrotta l'esecuzione, invocando il comando `RESUME`.

### 3.6 Il file.hex

Per trasferire il programma nella memoria il formato di file utilizzato è *l'Intel Hex*, che permette di codificare le istruzioni in un file di testo, in cui ogni linea contiene valori esadecimali.

Ogni linea consiste in:

1. Start code, un carattere ASCII “:” che indica l’inizio di una sequenza
2. Byte count, 1 byte che rappresenta il numero di byte di dato nella sequenza
3. Address, 2 byte che indicano l’indirizzo di partenza della porzione di memoria dove risiedono i dati
4. Record type, 1 byte che rappresenta il tipo di sequenza.

Nel nostro caso è sempre utilizzata l’opzione “ 00 “ che indica un “ *Data Record*”

5. Data, numero di byte che rappresentano la quantità di dati stessi
6. Crc, 1 byte calcolato sommando i valori di ogni coppia di digit nella sequenza, modulo 256, prendendone il complemento a 2.

Per esempio:

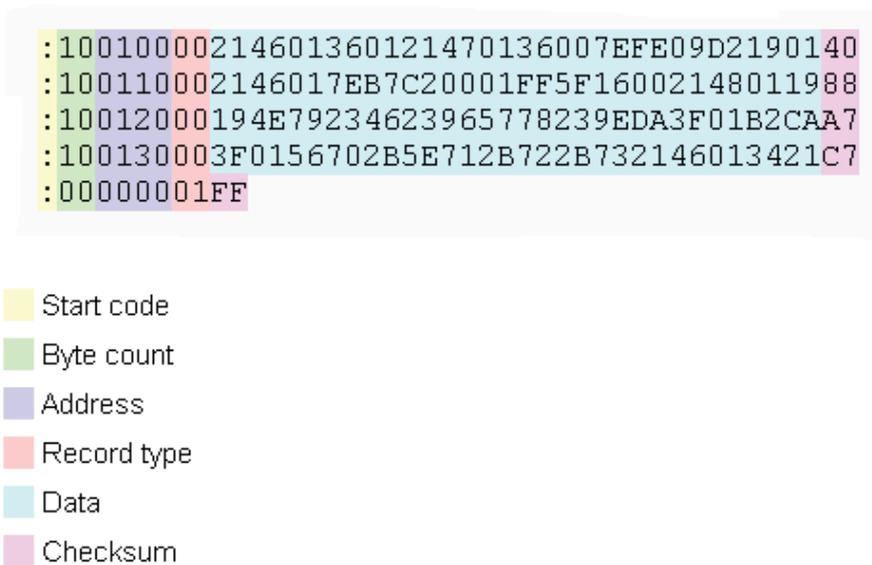


Figura 3.19. Esempio di file.hex

## 3.7 Programmazione della memoria Flash

Per programmare la memoria flash del CC2510 l'host dall'esterno deve inizialmente inviare le istruzioni attraverso la `Debug_instr`, che permette la programmazione della memoria flash con la *Flash Controller*.

La Flash Controller si occupa della scrittura e cancellazione della memoria flash e ha le seguenti caratteristiche:

- Programmabile a parole da 16 bit
- Cancellabile per pagine
- Bit di Lock per la protezione su scrittura e sicurezza del codice
- Tempo di cancellazione di una pagina flash pari a circa 20 ms
- Tempo di scrittura di una pagina flash pari a circa 20 us

Quando si effettuano delle operazioni di scrittura la flash è indirizzabile a parola, usando 14 bit di indirizzo scritti sui registri di indirizzo *FADDRH* e *FADDRL*.

Quando si effettuano operazioni di cancellazione di una pagina, la pagina da cancellare è indirizzabile attraverso i bit più alti del registro *FADDRH*.

La CPU può scrivere i dati direttamente sulla flash, mentre il codice del programma è in esecuzione sulla RAM.

I dati sono scritti nel registro di scrittura *FWDATA*.

La scrittura sulla flash avviene ogni volta che sono scritti due bit sul registro *FWDATA*, con il settaggio di *FTCL.SWBSY* = 1.

La gestione della scrittura nel registro *FWDATA* è arbitrata dalla CPU attraverso il segnale di stato.

Il tutto ha un *Timeout* di 40  $\mu$ s e ad un errore in questa fase di scrittura corrisponde la cancellazione della parte relativa all'errore.

Il *FCTL.BUSY* bit riporta la locazione dell'operazione fallita nei registri e il tutto poi si ripete.

Per ricominciare completamente è necessaria la cancellazione completa della pagina. La cancellazione avviene settando *FTCL.ERASE* = 1 e se in contemporanea anche *FCTL.WRITE* = 1, la fase di cancellazione viene eseguita prima dell'operazione di scrittura.

Il *FCTL.BUSY* bit rileva quando la cancellazione è terminata.

La Flash Controller contiene un generatore di clock, che controlla la distribuzione nel tempo delle operazioni di scrittura e cancellazione nella flash.

Questa informazione è settata nel registro *FLASH WRITE TIMING* e alla frequenza di 26 MHz l'FWT è pari al valore esadecimale 0x23.

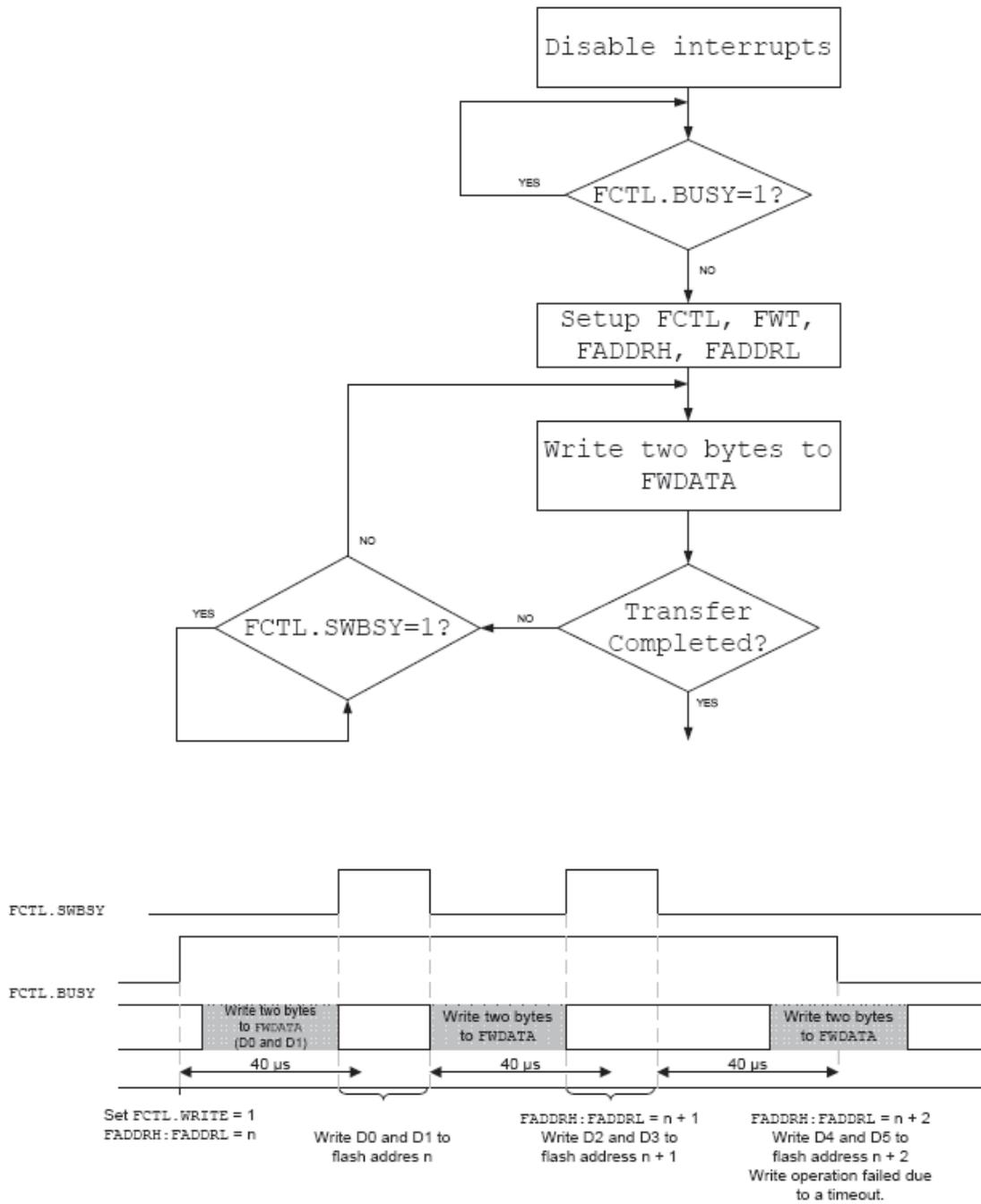


Figura 3.20. Scrittura sulla flash dalla ram e relativo timeout

## 3.8 Programmatore del CC2510

Il codice sviluppato per programmare il CC2510 segue come filosofia di base la seguente funzione Main.c, che richiama le librerie precedentemente citate, Xmodem.c. e Debug.c:

```
uart_setup
do
    wait for command ;
    if (GETID_CMD)
        .
        .
    if (ERASE_CMD)
        .
        .
    if (PROGRAM_CMD)
        .
        .
    if (DUMP_CMD)
        .
        .
    if (RESET_CMD)
        .
        .
repeat forever
```

Le funzioni principali utilizzate sono:

- `void printUART(unsigned char *message)`, che invia tutti i caratteri di message sulla uart
- `void GetID(void)`, che inizializza la Debug Interface, richiede l'ID del chip, leggendo due variabili per verificare il funzionamento della uart e il protocollo richiede la ritrasmissione al chip delle due variabili
- `void Erase(void)`, che inizializza la Debug Interface, blocca il processo in corso e richiede la cancellazione, aspettando che il valore restituito abbia l'MSB = 0
- `unsigned short ProgramLoad(unsigned char xdata *hex)`, che esegue i controlli sulla lunghezza e la correttezza dei dati da trasmettere.

Essa inizializza l'indirizzo di memoria da cui partire a scrivere i dati, separando lo spazio in cui scrivere in due parti (1 byte alla volta).

Per tutta la lunghezza del pacchetto del file da inviare prende i dati, compone il byte, lo trasmette, e ripete questo fino alla fine del file.

- `void Program(void)`, che inizializza la Debug Interface, trasferisce il codice del programma dalla RAM alla FLASH attraverso la funzione `Ram2Flash`.

Essa ricarica in RAM il programma scritto nel buffer, controllando gli eventuali errori e carica il program counter

- `void Ram2Flash(void)`, che carica la lunghezza del file eseguibile da trasferire alla flash.

Essa inizializza le porte, carica l'indirizzo di partenza dell'eseguibile all'interno della flash, e dopo aver configurato la flash esegue la copia dato per dato

- `void ReadRAM(void)`, che predispone quanti dati leggere in RAM
- `void Reset(void)`, che resetta il sistema ogni qualvolta richiesto.

### 3.9 Interfaccia Utente

Durante l'implementazione del software l'interfaccia utente, utilizzata per visualizzare il set di istruzioni, è stata l'interfaccia *Tera Term Pro* (Fig. 3.21) con i seguenti settaggi:

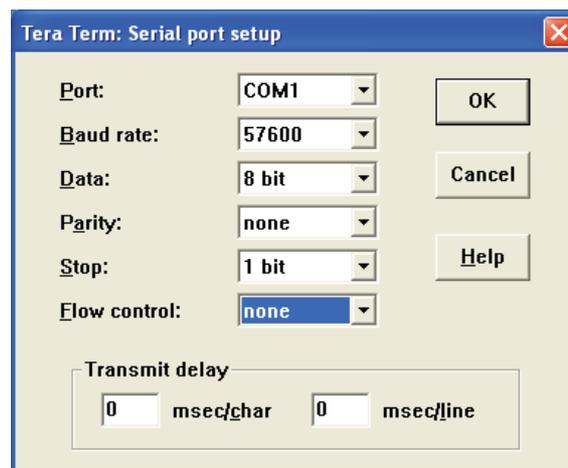


Figura 3.21. Tera Term Pro

In parallelo all'implementazione del software veniva realizzata dal *Dottorando Stefano Speretta*, che ha collaborato con me a questo progetto, l'interfaccia grafica definitiva attraverso l'utilizzo del software *Labwindows*, che mi ha poi permesso di realizzare le successive interfacce grafiche presenti nel continuo del progetto.

In Fig. 3.22 è riportata l'interfaccia grafica del programmatore del CC2510:

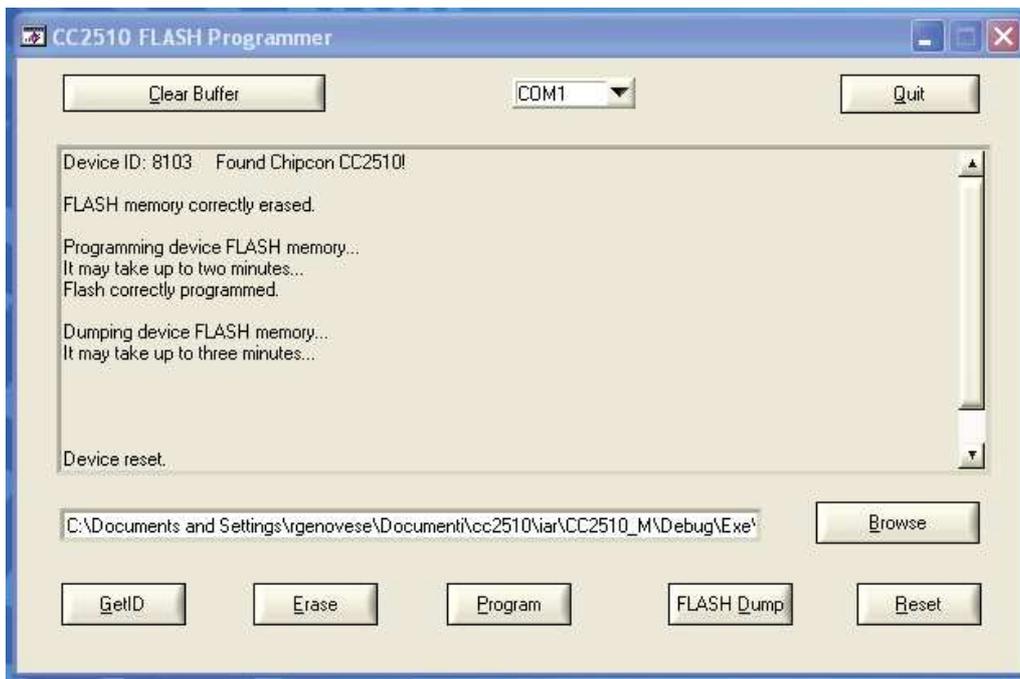


Figura 3.22. Programmer CC2510

per mezzo della quale:

- Selezionando il pulsante *GetId*, viene acquisito e visualizzato l'identificativo del chip.

Nel caso di ID corretto(8103) viene di seguito visualizzato il messaggio *FOUND CHIPCON CC2510!*

- Selezionando il pulsante *Erase*, viene eseguita la cancellazione della memoria flash del chip, per eliminare qualsiasi configurazione spuria a bordo.

Di seguito viene visualizzato il messaggio *FLASH memory correctly erased*

- Selezionando il pulsante *Browse*, viene eseguita la navigazione per la ricerca del sorgente in esadecimale, desiderato, da caricare in memoria, in seguito al pulsante *Program*.

Il sorgente viene visualizzato nella barra di indirizzo presente alla sinistra del pulsante stesso

- Selezionando il pulsante *Program*, viene programmata la memoria flash del chip.

Di seguito viene visualizzato il messaggio *Programming device FLASH memory... It may take up to two minutes... Flash correctly programmed*, in cui è presente il relativo messaggio di tempo impiegato per la programmazione

- Selezionando il pulsante *FLASH dump*, viene letta la memoria flash del chip per poterne controllare il contenuto.

Di seguito viene visualizzato il messaggio *Dumping device FLAH memory... It may take up to three minutes...*, anche in questo caso con il relativo messaggio di tempo impiegato per l'operazione

- Selezionando il pulsante *Reset*, viene resettato il chip.

Di seguito viene visualizzato il messaggio *Device reset*

- Selezionando il pulsante *Quit*, viene chiusa l'interfaccia grafica
- Selezionando il pulsante *Clear buffer*, viene cancellato il buffer di messaggi temporanei sull'interfaccia.

Inizialmente, però, deve essere selezionata la porta di collegamento seriale, utilizzando la finestra a tendina in cui sono riportate le porte COM.

L'utilizzo di queste interfacce segue il caricamento del firmware a bordo dell'evaluation board del CC1010, che viene programmata tramite l'interfaccia parallela usando il software della Chipcon, *Chipcon CC1010 Flash Programmer* :



Figura 3.23. Programmer CC1010

per mezzo della quale:

selezionando il pulsante *Browse*, viene eseguita la navigazione per la ricerca del sorgente in esadecimale, desiderato, da caricare in memoria, in seguito al pulsante *DO IT*, dopo aver selezionato la frequenza di clock del chip pari a 11.0592 MHz e l'operazione da eseguire, che in questo caso è *Erase, Program & Verify*.

# Capitolo 4

## Software di trasmissione (Tx) e ricezione (Rx)

Dopo la realizzazione del programmatore del transceiver CC2510 si è passati allo sviluppo del *core* dell'intero progetto, ossia del codice di trasmissione e ricezione di pacchetti.

Lo schema di base per la gestione della comunicazione è il seguente:

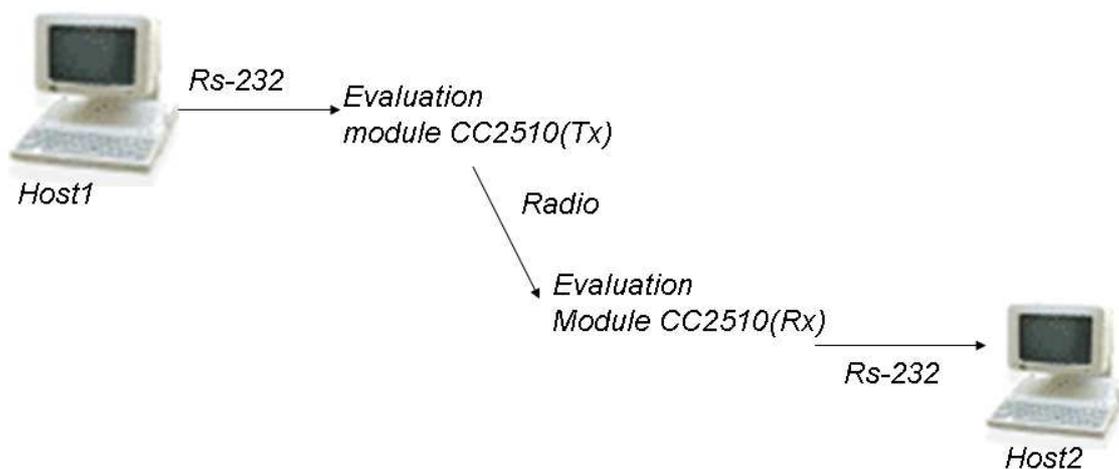


Figura 4.1. Schema di comunicazione

In esso è possibile individuare le due tipologie di canali di comunicazione: il canale wire, *il cavo seriale Rs-232* e il canale wireless.

## 4.1 Struttura dei pacchetti

Il pacchetto che viene inviato alla Radio attraverso il cavo seriale RS-232 è stato, convenzionalmente, configurato nei 4 campi:

Command	Length	Payload	Crc
1 BYTE	1 BYTE	0-255 BYTE	1 BYTE

Figura 4.2. Pacchetto di dati via seriale

- Command, un carattere ASCII che indica le istruzioni relative ai diversi funzionamenti della Radio
- Length, 1 byte che indica la lunghezza del campo payload
- Payload, che contiene i dati di lunghezza variabile tra 0 e 255 byte al massimo
- Crc, 1 byte che indica il controllo di parità sui bit calcolato, facendo la somma in (modulo-8) dei byte.

Di seguito viene riportata la macro estrapolata dal codice per il calcolo del CRC:

```
unsigned char CRC( unsigned char *pDato, unsigned short dim )
{
    uart_crc = 0;

    for( uart_i = 0; uart_i < dim; uart_i++ )
        uart_crc+ = pDato[uart_i];

    return uart_crc;
}
```

Per quanto riguarda invece il pacchetto via Radio, esso è suddiviso nei 6 campi:

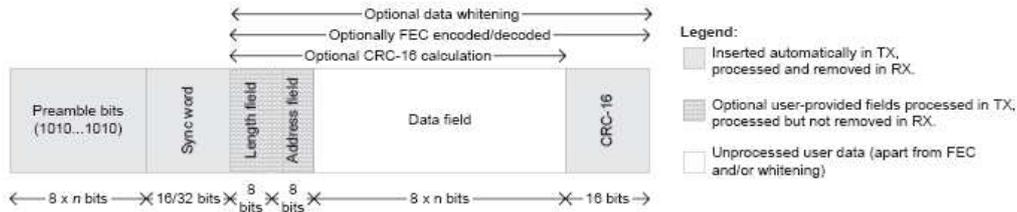


Figura 4.3. Pacchetto di dati via radio

- Preamble, che contiene una sequenza alternata di uni e zeri (1 0 1 0.....) per sincronizzare la parte ricevente con la velocità di trasmissione della parte trasmittente e riconoscere i livelli in ricezione 0 e 1
- Synchronization word, 4 byte per la sincronizzazione a livello di byte all'inizio dell'intero pacchetto
- Length byte, che contiene la lunghezza del campo payload.

Si tratta di un campo opzionale, in quanto la lunghezza può essere fissa, e in questo caso è settata in un registro di configurazione della Radio, o variabile da 0 a 255 byte al massimo e in questo caso fa parte del formato del pacchetto insieme al crc

- Address byte, che contiene l'indirizzo di destinazione dei byte.

Anch'esso è un campo opzionale e può essere disabilitato

- Payload, che contiene i dati
- Crc, 2 byte opzionali per il controllo di parità dei dati.

## 4.2 Gestione delle funzionalità della Radio

La CPU utilizza un set di *command strobes* da 1 byte per il controllo delle diverse funzionalità della Radio, quali: l'abilitazione del sintetizzatore di frequenza, la trasmissione, la ricezione e altre.

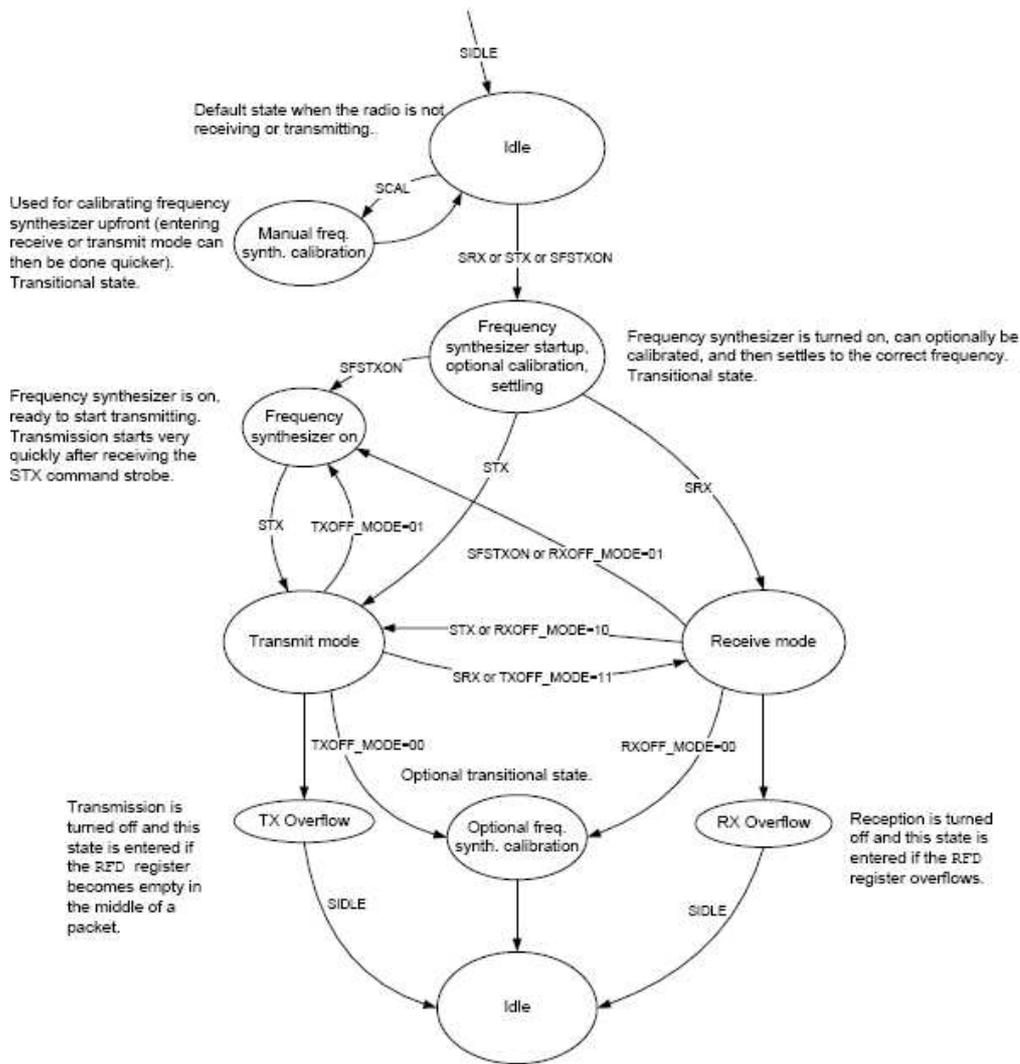


Figura 4.4. Diagramma a stati della radio

Questi comandi vengono dati alla Radio scrivendo sul registro SFR.

Il funzionamento della Radio è configurato attraverso un set di registri RF, che sono mappati sullo spazio di memoria Xdata e forniscono anche informazioni sullo stato stesso della Radio.

Per la configurazione di questi registri si utilizza il Software della Texas Instruments *SmartRf Studio*, attraverso il quale settando le specifiche caratterizzanti la Radio del CC2510, tra cui la frequenza di lavoro, la modulazione, la deviazione di frequenza e la velocità di trasmissione dei byte, è possibile ottenerne la corretta configurazione.

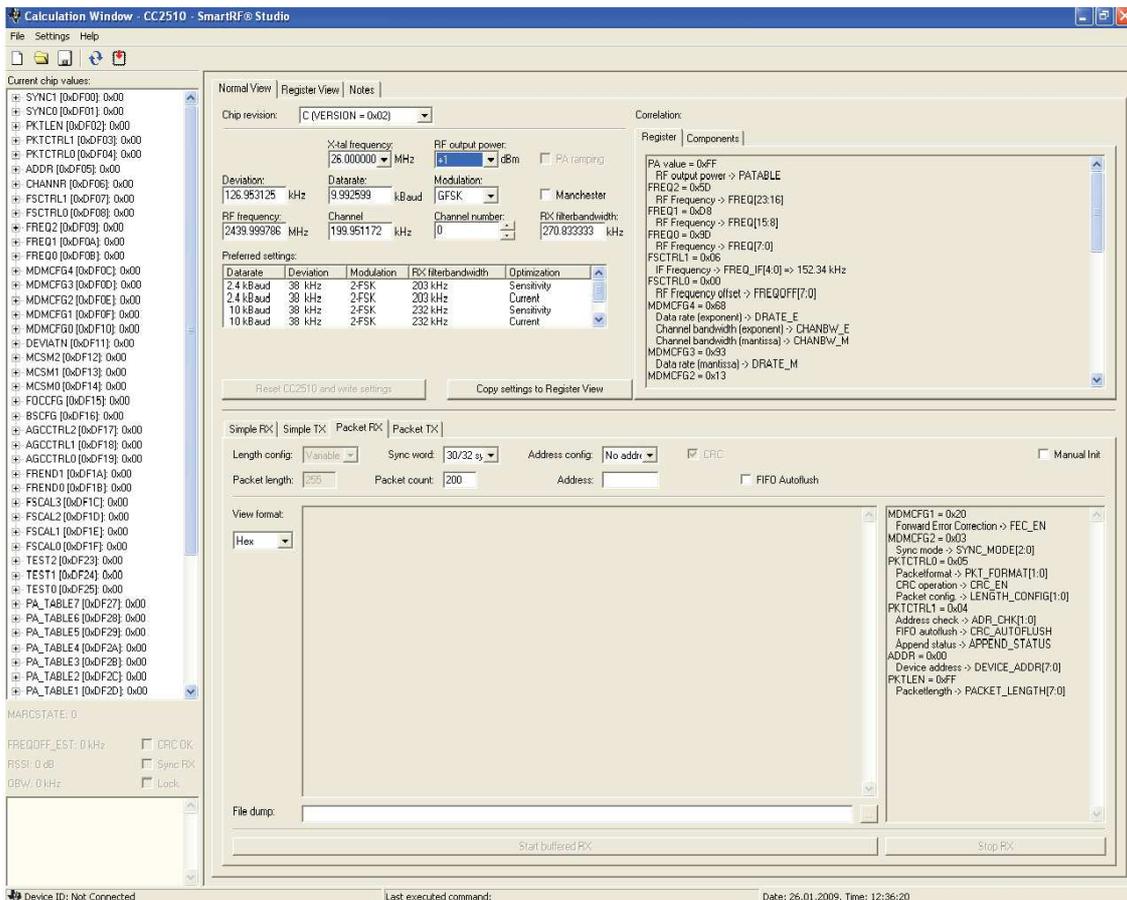


Figura 4.5. SmartRf Studio

La Radio è associata a due vettori di interrupt: *l'interrupt rftarrx*, che indica quando un dato è stato trasmesso o ricevuto e *l'interrupt rf flag*, che segnala altri eventi come lo svuotamento imprevisto del buffer di trasmissione, l'incapacità del buffer di ricezione di contenere i dati, la trasmissione o ricezione del pacchetto nel tempo previsto dal timeout o la rilevazione della sync\_word.

Le funzioni realizzate, sulla base delle librerie e degli esempi di codice forniti dalla Chipcon, per la ricetrasmisione attraverso il canale Radio sono state raggruppate nella libreria RF.h e sono:

- void RfSendPacket(unsigned char \*pDato, char dim), che trasmette i pacchetti
  1. Viene settata la lunghezza del pacchetto da inviare
  2. Viene abilitata la Radio, quindi si passa allo stato di Idle del diagramma a stati

3. Vengono abilitati gli interrupt rftxrx e rf flag
  4. Viene abilitata la Tx in seguito al command strobe Stx()
  5. Se non si verifica nessun interrupt i dati vengono scritti nel registro RFD e successivamente caricati in uno spazio di memoria utilizzato come buffer per la Tx
  6. Dopo alcuni controlli sui flag, si attende che la Tx termini per effettuare un ulteriore caricamento dei dati, quindi vengono resettati i valori degli interrupt e viene ritornato il comando al microcontrollore.
- `void RfReceivePacket( unsigned char *pDato)`, che riceve i pacchetti
    1. Viene abilitato il Timer atto a monitorare il Timeout
    2. Vengono abilitati i relativi interrupt
    3. Vengono cancellati gli interrupt flag e i dati vengono caricati nel buffer pPayload
    4. Viene abilitata la Rx in seguito al command strobe srx()
    5. Se non si riceve nulla il Timer si incrementa fino al valore di Timeout; ciò viene segnalato dall'interrupt, vengono resettati i valori degli stessi e viene ritornato il comando al microcontrollore
    6. Se si riceve qualcosa il Timer viene resettato fino a ricezione terminata.

L'implementazione di questa macro ha comportato la realizzazione fisica di una seconda evaluation module del CC2510, che fungesse da ricevitore per poterne così valutare il corretto funzionamento.

- `void RfConfig(registri *config\_struct)`, che configura i registri della Radio

La configurazione viene già predefinita nel codice con i settaggi di SmartRF Studio dall'impostazione di determinati parametri come il formato del pacchetto, la modulazione e la potenza d'uscita, oppure è possibile settare i registri dall'esterno.

Nel primo caso i registri sono così configurati(si veda Fig 4.5):

```
{IF FREQUENCY}
FSCTRL1 = 0x06;
FSCTRL0 = 0x00;
```

{FILTER BANDWIDTH, DATA RATE}

MDMCFG4 = 0x68;

MDMCFG3 = 0x93;

{CHANNEL SPACING, CHANNEL NUMBER}

MDMCFG1 = 0x13;

MDMCFG0 = 0x20;

CHANNER = 0x00;

{DEVIATION}

DEVIATN = 0x62;

{RF output power(PA\_POWER)}

FREND0 = 0x10;

FREND1 = 0x56;

{FREQUENCY SYNTHESIZER CALIBRATION}

MCSM0 = 0x10;

MCSM2 = 0x07;

{FREQUENCY OFFSET COMPENSATION}

FOCCFG = 0x16;

{BIT SYNCHRONIZATION}

BSCFG = 0x6C;

{AGC control}

AGCCTRL2 = 0x43;

AGCCTRL1 = 0x41;

AGCCTRL0 = 0x91;

{FREQUENCY SYNTHESIZER}

FSCAL2 = 0xA9;

FSCAL1 = 0x0A;

FSCALO = 0x11;

TEST2 = 0x81;

TEST1 = 0x35;

TEST1 = 0x09;

Per quanto riguarda invece la configurazione dall'esterno alcuni settaggi rimangono invariati mentre altri registri vengono programmati come segue:

l'ampiezza di banda del filtro di canale del ricevitore, che varia al variare della frequenza d'oscillazione, è programmata dai valori dei registri

MDMCFG4.CHANBW\_E e MDMCFG4.CHANBW\_M.

La relazione tra il settaggio di questi registri e l'ampiezza di banda del filtro di canale è data dalla formula:

$$BW_{channel} = \frac{f_{xosc}}{8 \cdot (4 + Chanbw\_m) 2^{Chanbw\_e}}$$

da cui si ricava:

$$Chanbw\_m = \frac{f_{xosc}}{8 \cdot BW_{channel} 2^{Chanbw\_e}} - 4$$

da cui ponendo Chanbw\_m = 0, si ha:

$$Chanbw\_e = \log_2 \frac{f_{xosc}}{BW_{channel} 2^5}$$

Dalle specifiche di progetto se Chanbw\_m è maggiore di 8:

Chanbw\_e++;

$$Chanbw\_m = \frac{f_{xosc}}{8 \cdot BW_{channel} 2^{Chanbw\_e}} - 4$$

Il registro MDMCFG4, come nel caso di tutti gli altri, viene disabilitato, mascherando a zero i bit della sua posizione e abilitato se il reset è attivo.

Dai valori relativi a i bit di posizione e al reset riportati sul datasheet del transceiver segue l'implementazione delle linee di codice che si seguono:

```
configurazione.r_MDMCFG4 = (configurazione.r_MDMCFG4 & ~0xC0) |
    (((unsigned char)(chanbw_e)<<6) & 0xC0)
configurazione.r_MDMCFG4 = (configurazione.r_MDMCFG4 & ~0x30) |
    (((unsigned char)(chanbw_m)<<4) & 0x30)
```

Il valore del data rate, che viene usato in trasmissione oppure che ci si aspetta in ricezione, è programmato dalla configurazione dei registri

MCMDFG3.DRATE\_M e MCMDFG3.DRATE\_E.

Il data rate è dato dalla formula:

$$Rdata = \frac{(256 + Drate\_m) 2^{Drate\_e}}{2^{28}} \cdot f_{xosc}$$

da cui si ricava:

$$Drate\_m = \frac{Rdata 2^{28}}{f_{xosc} 2^{Drate\_e}} - 256$$

da cui ponendo  $Drate\_m = 0$ , si ha:

$$Drate\_e = \log_2 \frac{Rdata 2^{20}}{f_{xosc}}$$

Se  $Drate\_m$  è arrotondato all'intero piu' vicino e diventa 256:

```
Drate_e ++;
Drate_m = 0
```

```
configurazione.r_MDMCFG4 =(configurazione.r_MDMCFG34& ~0x0F) |
    ((unsigned char)(drate_e) & 0xC0)
configurazione.r_MDMCFG3 = (unsigned char)(drate_m)
```

Anche la deviazione di frequenza è programmata dai registri `DEVIATION_M` e `DEVIATION_E`.

La formula relativa è data da:

$$f_{dev} = \frac{f_{xosc}}{2^{17}} \cdot (8 + Deviation\_m) 2^{Deviation\_e}$$

da cui si ricava:

$$Deviation\_m = \frac{f_{dev} / 2^{17}}{f_{xosc} 2^{Deviation\_e}} - 8$$

da cui ponendo  $Deviation\_m = 0$ , si ha:

$$Deviation\_e = \log_2 \frac{f_{xosc} / 2^{14}}{f_{xosc}}$$

Dalle specifiche, se  $Deviation\_m$ , che è interpretato come un valore a 4 bit, è maggiore di 7.5:

```
Deviation_e ++;
```

$$Deviation_m = \frac{f_{dev}/2^{17}}{f_{xosc}2^{Deviation_e}} - 8$$

```
configurazione.r_DEVIATN =(configurazione.r_DEVIATN & ~0x70) |
  (((unsigned char)(deviation_e)<<4) & 0x70)
configurazione.r_DEVIATN = (configurazione.r_DEVIATN & ~0x07) |
```

Nel caso di modulazione MSK:

```
configurazione.r_DEVIATN = 0;
configurazione_deviazione = 0.0;
```

La potenza in uscita può essere programmata con opportuni settaggi del registro PA\_TABLE, dipendenti dai diversi valori di potenza dissipata, dalla banda di frequenza e dal relativo consumo di corrente del tranceiver.

Output Power [dBm] Typical 25°C, 3.0 V	Setting	Current Consumption, Typ. [mA]
(-55 or less)	0x00	12
-30	0x44	13
-28	0x41	13
-26	0x54	15
-24	0x53	14
-22	0x83	14
-20	0xC1	14
-18	0xC8	15
-16	0x87	14.5
-14	0x59	15
-12	0x95	15.5
-10	0xCB	16
-8	0x99	16.5
-6	0x7F	18.5
-4	0xAA	20
-2	0xBF	21.5
0	0xFE	26
1	0xFF	26.5

Figura 4.6. Settaggi relativi ai diversi valori di potenza in uscita

`configurazione.r_PA_TABLE0 = configurazione.r_PA_TABLE0 = power`

I pacchetti sono controllati dai registri PKTCTRL1 e PKTCTRL1.

Il registro PKTCTRL0 è costituito dai campi:

1. `Append_status`, che abilitato fa sì che il payload sia seguito da due byte di stato che conterranno l'Rssi e l'Lqi

```
configurazione.r_PKTCTRL1 =
(configurazione.r_PKTCTRL1 & ~0x04)|((status << 2) & 0x04)
```

2. `Adr_Chk`, che controlla l'indirizzo della corretta configurazione dei pacchetti ricevuti e può avere le seguenti opzioni di configurazione: `No_Addr_Check` (0x00), `No_Broadcast` (0x01), `On_Broadcast` (0x03)

```
configurazione.r_PKTCTRL1 =
(configurazione.r_PKTCTRL1 & ~0x03)|(addr_option & 0x03)
```

Il registro PKTCTRL0 è invece costituito dai campi:

1. `White_Data`, che spazia la sequenza dei dati, diminuendo così la potenza dissipata e aumentando il segnale utile

```
configurazione.r_PKTCTRL0 =
(configurazione.r_PKTCTRL0 & ~0x40)|((white <<6 ) & 0x40)
```

2. `Pkt_Format`, che può essere: `Normal_Mode` (0x00), `Synchronous_Mode` (0x01), `Random_Mode` (0x02), `Asynchronous_Mode` (0x03)

```
configurazione.r_PKTCTRL0 =
(configurazione.r_PKTCTRL0 & ~0x30)|(pkt_format & 0x30))
```

3. `CC2400_En`, che permette di usare come supporto il CC2400, in particolare la stessa implementazione del crc

```
configurazione.r_PKTCTRL0 =
(configurazione.r_PKTCTRL0 & ~0x08)|(( CC2400 << 3 )& 0x08)
```

4. `Crc_En`, che abilita o disabilita il crc

```
configurazione.r_PKTCTRL0 =
(configurazione.r_PKTCTRL0 & ~0x04)|(( crc_x << 2)& 0x04)
```

5. Length\_Config, che configura la lunghezza del pacchetto e può essere: Fixed\_Length (0x00), Variable\_Length (0x01)

```
configurazione.r_PKTCTRL0 =  
(configurazione.r_PKTCTRL0 & ~0x03) | (length_config & 0x03)
```

Il modem in trasmissione è configurato dal registro MDMCFG1 che si divide nei due campi:

1. Fec\_En, che consente di cancellare l'effetto di eventuali errori in trasmissione, quindi una diminuzione della probabilità di ritrasmissione dei pacchetti

```
configurazione.r_MDMCFG1 =  
(configurazione.r_MDMCFG1 & ~0x80) |  
((configurazione_old.fec << 7) & 0x80);
```

2. Preamble, che consente di settare il numero di byte dedicato al preambolo da trasmettere

```
configurazione.r_MDMCFG1 =  
(configurazione.r_MDMCFG1 & ~0x70) |  
(configurazione_preamble & 0x70);
```

Le modulazioni permesse dal CC2510 sono: FSK (0x00), GFSK (0x01), OOK (0x03) e MSK (0x07).

Il registro per settarle è così programmato:

```
configurazione.r_MDMCFG2 =  
(configurazione.r_MDMCFG2 & ~0xF0) |  
(modulazione & 0xF0)
```

Il registro MDMCFG2 si divide ancora nei due campi:

1. Manchester\_En, che permettere di utilizzare la codica e decodifica Manchester

```
configurazione.r_MDMCFG2 =  
(configurazione.r_MDMCFG2 & ~0x08) |  
((manchester << 3) & 0x08)
```

2. Sync\_mode, che setta il valore della syncword, che può assumere i seguenti valori:

No\_Preamble\_Sync (0x00), Sync\_15\_16 (0x01), Sync\_16\_16 (0x02), Sync\_30\_32 (0x03)

```

configurazione.r_MDMCFG2 =
(configurazione.r_MDMCFG2 & ~0x07) |
(syncword & 0x07)

```

Gli otto bit più significativi dei 16 bit della syncword sono settati nel registro  $Syncl = sync \gg 8$  e quelli meno significativi nel registro  $Synco = sync \& 0xFF$ .

- void RfCarrier(void), che genera la portante
  1. Viene abilitato il Timer e gli interrupt ad esso correlati
  2. Se la condizione di Timer coerente è verificata, allora viene generata la portante altrimenti si ritrasmette via seriale il comando K

Inizialmente questa funzione è stata utilizzata per verificare che il trasmettitore lavorasse correttamente, analizzando il fenomeno di trasmissione con l'analizzatore di spettro.

Ciò è stato possibile in quanto il CC2510 prevede tra le modulazioni anche la modulazione OOK (*On Off Keying*) chiamata anche ASK (*Amplitude Shift Keying*).

Un segnale OOK è rappresentato così:

$$s(t) = Am(t)\cos(\omega t)$$

dove  $m(t)$  è un segnale unipolare binario

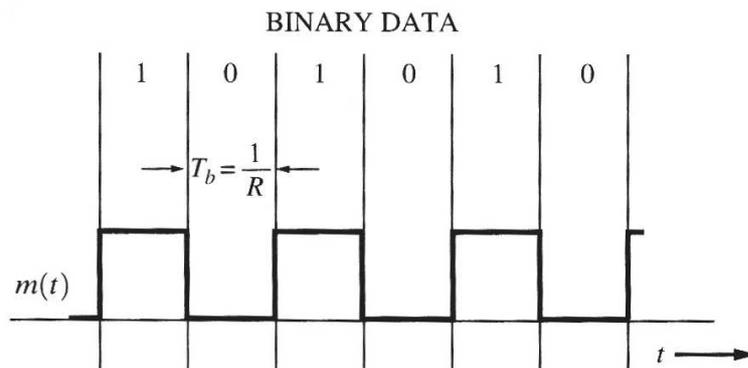


Figura 4.7. Modulazione unipolare

da cui ne deriva la densità spettrale di potenza qui riportata:

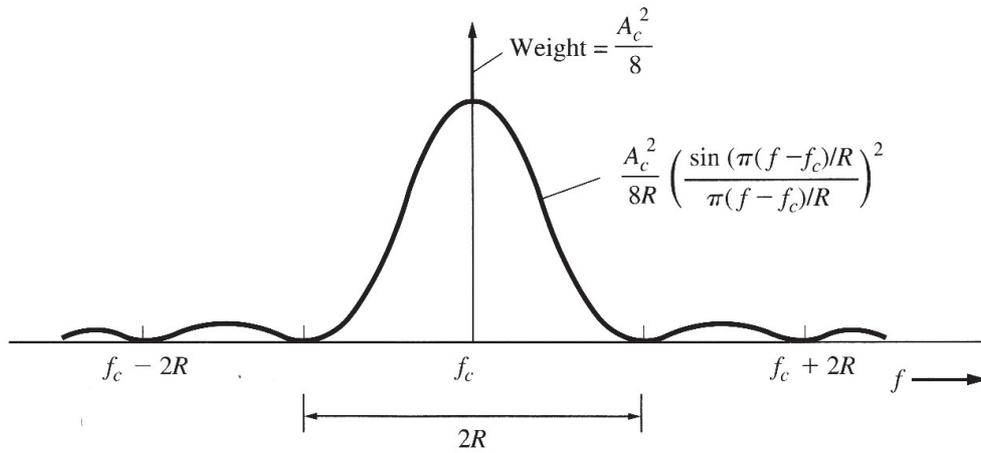


Figura 4.8. Densità spettrale di potenza

Nel nostro caso, invece, la portante generata è stata:

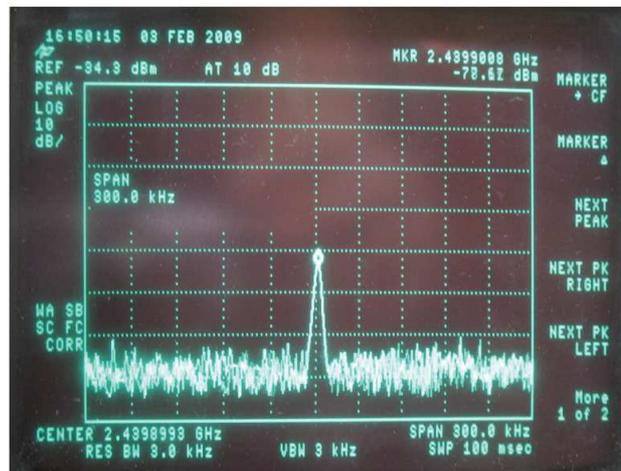


Figura 4.9. Portante generata a 2.439 GHz

- `void RfSetRadioFrequency(UINT32 frequency)`, che setta la frequenza della Radio. La frequenza deve preferibilmente essere espressa in kHz.
- `void rf_wait(BYTE wait)`, che attende un dato numero di ms indipendentemente dalla frequenza d'oscillazione.
- `#pragma vector=RFTXRX_VECTOR __interrupt static void rf_txrx_IRQ(void)`,

che cancella gli elementi di memoria all'interno dei quali sono memorizzati gli stati evolutivi degli interrupt.

Le funzioni che gestiscono il Timer sono riportate nella libreria `TIMER.h` e sono :

```
WORD timer1_SetPeriod(DWORD period),
```

che imposta il Timer su un certo periodo di tempo espresso in  $\mu$ s.

Il Timer viene inizializzato dalla macro `TIMER_INIT()`, che resetta i settaggi delle varie impostazioni e cancella i flag e viene avviato o spento dalla macro `TIMER_RUN(TRUE/FALSE)`.

In generale i 3 step sono :

```
TIMERx_INIT();  
BOOL halSetTimerxPeriod(period);  
TIMERx_RUN(TRUE)
```

la variabile *period* assume il valore massimo quando è impostata a zero.

### 4.3 Gestione della comunicazione via seriale Rs-232, libreria `uart.h`

Lo standard Rs-232 è stato definito dall'EIA (*Electronic industries association*), nei primi anni '60 come protocollo di comunicazione seriale, asincrono, a bassa velocità per collegare un PC ad un dispositivo esterno.

Secondo le specifiche dell'Rs-232, poichè il canale deve essere immune dai disturbi, dovrebbe avere dei valori di tensione in uscita e in ingresso compresi tra -15V e +15V, incompatibili con quelli dei circuiti realizzati in tecnologia TTL o in CMOS, per i quali i valori vanno di solito da 0V a 5V. Quindi è necessario traslare i valori di tensione per adattare il cavo, e inoltre anche invertirli, in quanto un segnale Rs-232 è codificato in logica negativa.

Per far ciò si utilizza il *Max232*, qui di seguito riportato.

Come si può notare è costituito da due canali di comunicazione Rs-232 bidirezionali.

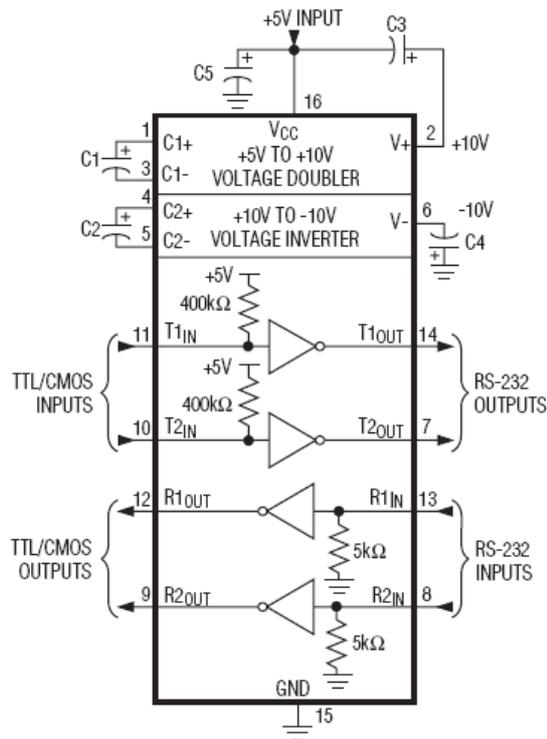


Figura 4.10. Max232

Nel nostro caso questi due canali sono stati utilizzati per la comunicazione tra l'Host1/Host2 e il CC2510(Tx)/CC2510(Rx).

Il CC2510 può utilizzare due interfacce seriali, la Usart0 e la Usart1, ognuna contenente due canali di comunicazione.

Nel codice è stata utilizzata l'interfaccia Usart0.

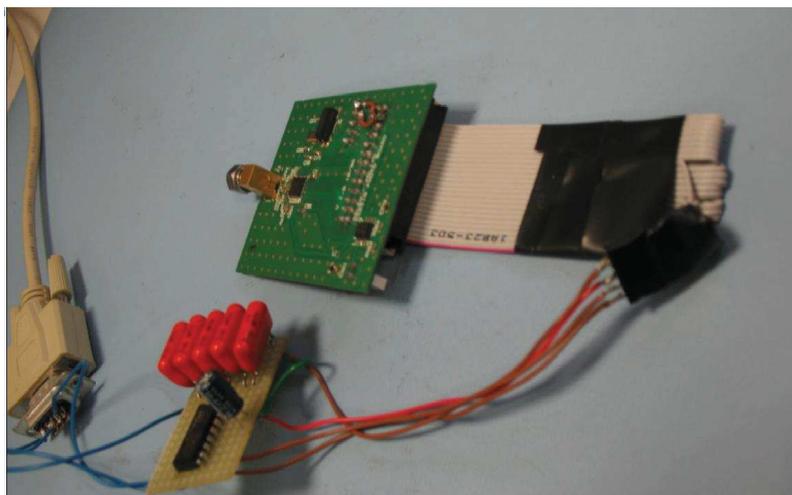


Figura 4.11. Comunicazione RS232-CC2510

La libreria realizzata a riguardo, chiamata `Uart.h`, è costituita dalle funzioni:  
`unsigned char uart_SendPacket(unsigned char *pDato)`, che trasmette i pacchetti alla Radio, controllando il CRC.

`unsigned char uart_ReceivePacket(unsigned char *pDato)`, che riceve i pacchetti dalla Radio, effettuando il controllo del CRC.

## 4.4 Principio di funzionamento della Ricetrasmis- sione: `main.c`

La procedura di Trasmissione e Ricezione è stata così analizzata:

- *Trasmissione e Ricezione Corretta dei Dati*, L'Host1 invia il comando via seriale al CC2510(Tx).

Il comando viene interpretato correttamente e di conseguenza viene inviato il pacchetto di riscontro `Ack_Pkt()` sulla seriale all'Host1.

A questo punto viene Trasmesso il pacchetto via radio.

Il pacchetto viene Ricevuto dal CC2510(Rx) e inviato via seriale all'Host2. Il suddetto Host invia un pacchetto di riscontro `Ack_Pkt()` di corretta ricezione, il quale compie il cammino inverso fino all'Host1

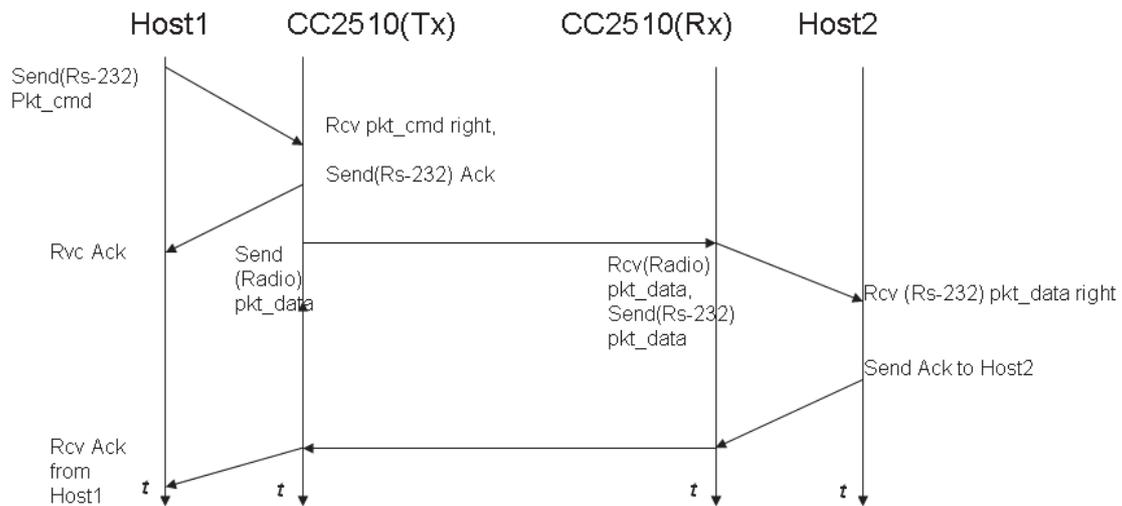


Figura 4.12. Tx e Rx corretta dei dati

- Possibilità di Errori in Trasmissione(Tx):

1. *Crc del pacchetto di comando*, inviato dall'Host1 al CC2510(Tx), errato e di conseguenza viene inviato all'Host1 il pacchetto di riscontro `Nack_Wrong_Crc_Pkt()`

2. *Comando non interpretato correttamente dal CC2510(Tx)* e di conseguenza viene inviato all'Host1 il pacchetto di riscontro

`Nack_Wrong_Command_Pkt()`

3. *Errore del Crc del pacchetto trasmesso via radio* e di conseguenza viene inviato via radio il pacchetto di riscontro

`NACK_Wrong_Crc_Pkt()` al CC2510(Tx), che rinvia il pacchetto

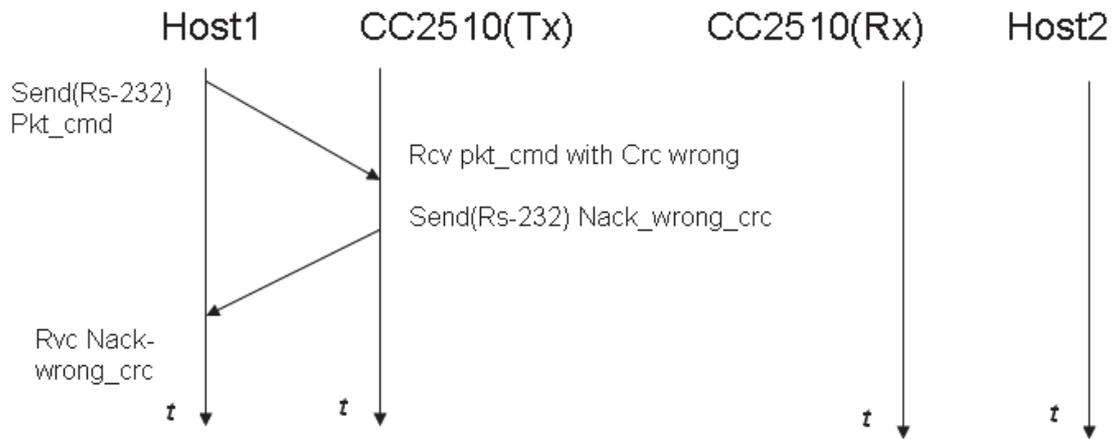


Figura 4.13. Errore di crc(Rs-232) in Tx

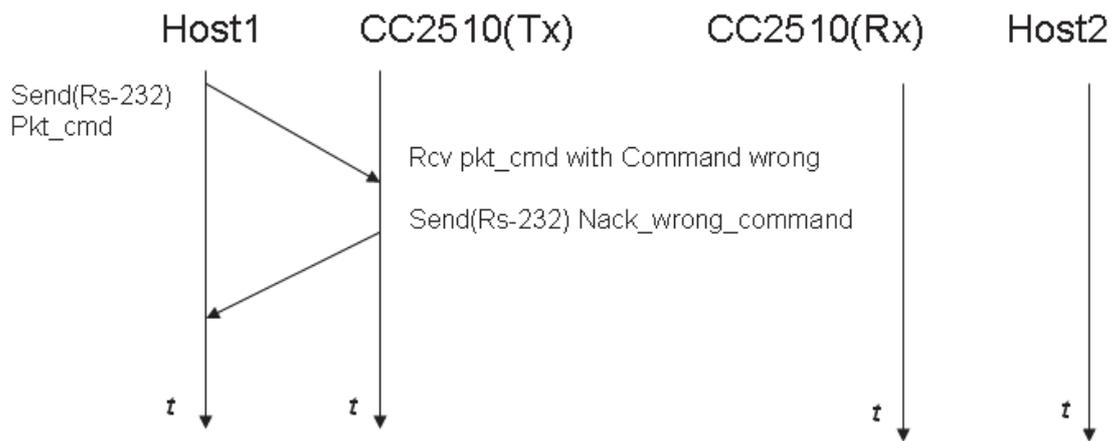


Figura 4.14. Errore di comando in Tx

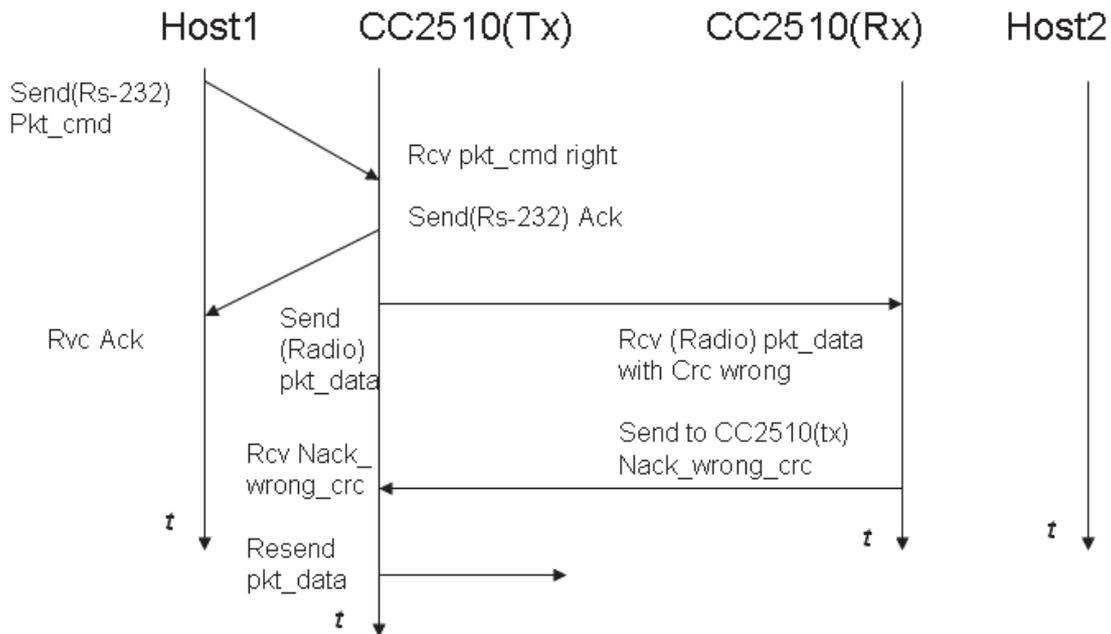


Figura 4.15. Errore di crc(Radio) in Tx

- Possibilità di Errori in Ricezione(Rx):

1. *Perdita del Pacchetto via Radio*, scade il Timeout e il CC2510(Tx) invia all'Host1 il pacchetto di riscontro Nack\_Timeout()
2. Se il Timeout non è scaduto possiamo avere due tipologie di errori:
  - *Errore di Comando del pacchetto ricevuto rilevato dall'Host2*, che invia il pacchetto di riscontro Nack\_Wrong\_Comand\_Pkt() all'Host1
  - *Errore di Crc rilevato dall'Host2*, che invia al CC2510(Rx) il pacchetto di riscontro Nack\_Wrong\_Crc\_Pkt(), di conseguenza viene ritrasmesso dal CC2510(RX) via seriale il pacchetto precedentemente ricevuto via radio dal CC2510(Tx).

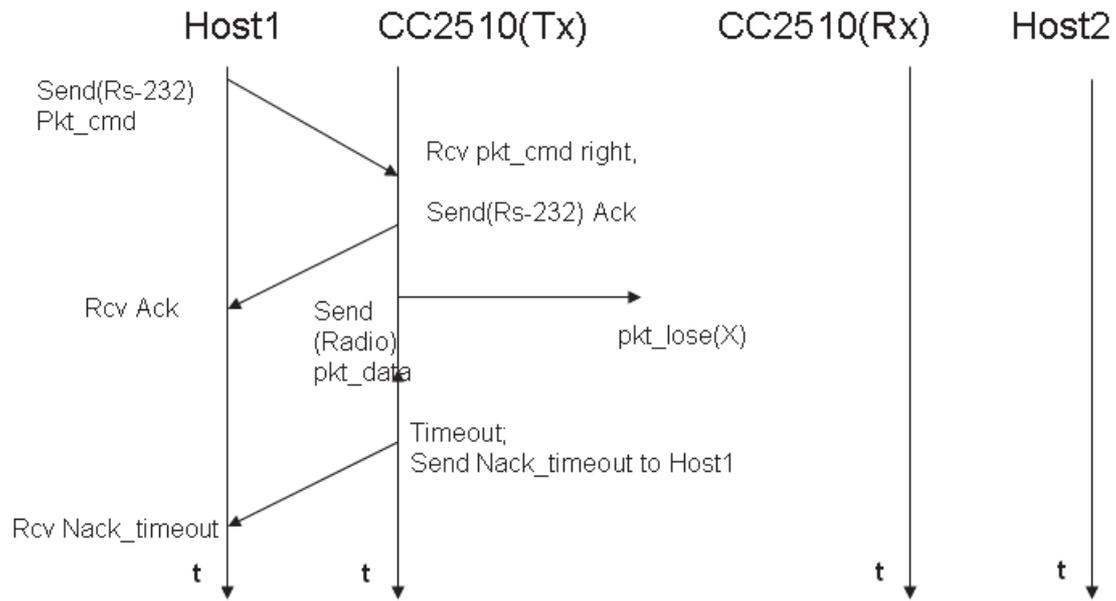


Figura 4.16. Perdita del pacchetto via radio

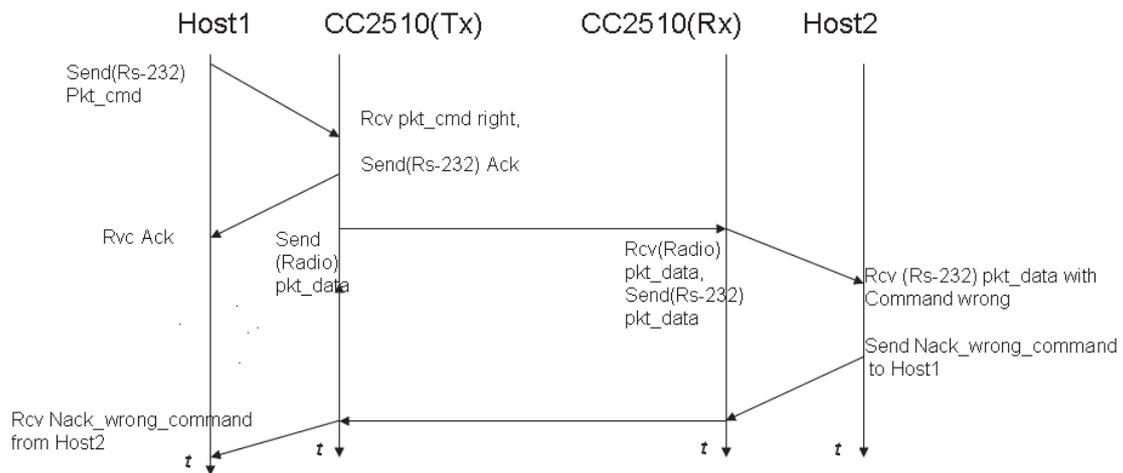


Figura 4.17. Errore di comando(Rs-232) in Rx

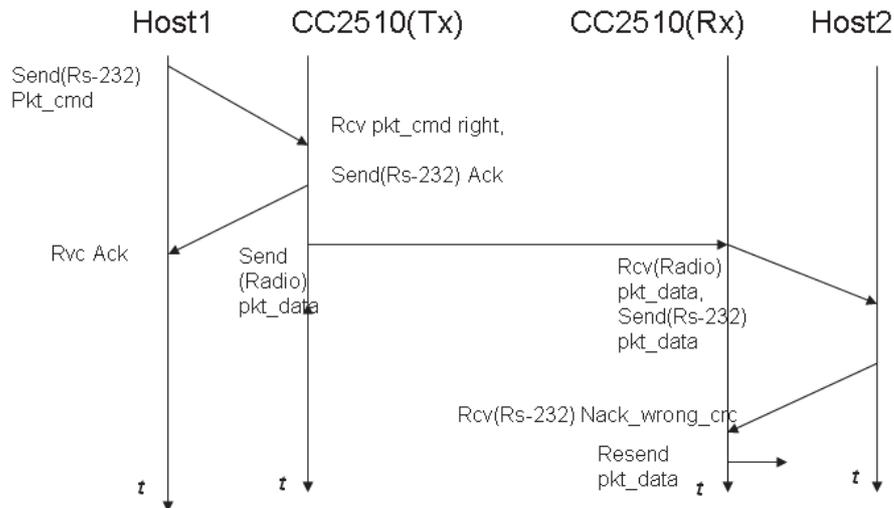


Figura 4.18. Errore di crc(Rs-232) in Rx

I comandi via seriale RS-232 inviati alla Radio e implementati nel progetto con la relativa struttura del pacchetto, di cui si è già precedentemente parlato, sono i seguenti:

PORTANTE\_cmd, che genera una portante

```

COMMAND: "p" (ASCII), 0x70;
LENGTH: 0;
PAYLOAD: 0
    
```

TX\_cmd, che trasmette pacchetti di lunghezza data

```

COMMAND: "T" (ASCII), 0x54;
LENGTH: (0-255)byte, 0-0xFF;
PAYLOAD: (0-255)byte
    
```

RX\_cmd, che riceve pacchetti

```

COMMAND : "r" (ASCII), 0x72;
LENGTH : 1 byte;
PAYLOAD : struct{
    unsigned char TIMEOUT;
}RX_cmd;
    
```

in cui il Timeout indica il tempo di attesa massimo tra l'invio di un pacchetto e la conferma di avvenuta ricezione = ( durata x 0.15)

CONFIG\_cmd, che configura i registri della Radio

```
COMMAND: "c" (ASCII), 0x63;
LENGTH: 34 byte;
PAYLOAD: struct {
    unsigned char config;
    unsigned char r_FREQ2;
    unsigned char r_FREQ1;
    unsigned char r_FREQ0;
    unsigned char r_FSCTRL1;
    unsigned char r_FSCTRL0;
    unsigned char r_MDMCFG4;
    unsigned char r_MDMCFG3;
    unsigned char r_MDMCFG1;
    unsigned char r_MDMCFG0;
    unsigned char r_CHANNR;
    unsigned char r_DEVIATN;
    unsigned char r_FREND0;
    unsigned char r_FREND1;
    unsigned char r_MCSM0;
    unsigned char r_MCSM2;
    unsigned char r_FOCCFG;
    unsigned char r_BSCFG;
    unsigned char r_AGCCTRL2;
    unsigned char r_AGCCTRL1;
    unsigned char r_AGCCTRL0;
    unsigned char r_FSCAL3;
    unsigned char r_FSCAL2;
    unsigned char r_FSCAL0;
    unsigned char r_TEST2;
    unsigned char r_TEST1;
    unsigned char r_TEST0;
    unsigned char r_PA_TABLE0;
    unsigned char r_PKTCTRL1;
    unsigned char r_PKTLEN;
    unsigned char r_PKTCTRL0;
    unsigned char r_MDMCFG2;
    unsigned char r_SYNC1;
    unsigned char r_SYNC0;
} registri;}
```

ECO\_cmd, che ritrasmette lo stesso pacchetto

```
COMMAND: "e" (ASCII), 0x65;
```

LENGTH: (0-255)byte (0-0xFF);

PAYLOAD: (0-255)byte

STATUS\_cmd , che trasmette l'RSSI, una stima del livello del segnale nel canale scelto e l'LQI, una stima della qualità del segnale ricevuto

COMMAND: "s" (ASCII), 0x73

LENGTH: 2 byte

PAYLOAD: struct{  
    unsigned char r\_LQI;  
    unsigned char r\_RSSI;  
}registri;

I pacchetti di riscontro sono :

Uart\_Ack\_pkt, che segnala se il pacchetto ricevuto è giusto

COMMAND: "A" (ASCII), 0x41;

LENGTH: 0;

CRC: 1 byte (A)

Uart\_Nack\_wrong\_command\_pkt, che segnala se il pacchetto ricevuto non è corretto perchè non è corretto, il comando

COMMAND: "n" (ASCII), 0x6E;

LENGTH: 0;

PAYLOAD: 1 byte (n)

Uart\_Nack\_wrong\_crc\_pkt, che segnala se il pacchetto ricevuto non è corretto perchè non è corretto, il crc

COMMAND: "'N" (ASCII), 0x4E;

LENGTH: 0;

PAYLOAD: 1 byte (N)

Mentre i pacchetti di riscontro via radio con gli stessi controlli sono:

Ack\_pkt

LENGTH: 2 byte;

PAYLOAD: 1 byte "A" (ASCII), 0x41;

CRC: 1 byte (A)

Nack\_wrong\_command\_pkt

LENGTH: 2 byte;

PAYLOAD: 1 byte "n" (ASCII), 0x6E;

CRC: 1 byte (n)

```

Nack_wrong_crc_pkt
LENGTH: 2 byte;
PAYLOAD: 1 byte "N" (ASCII), 0x4E;
CRC: 1 byte (N)

```

## 4.5 Interfaccia Utente

Per verificare il funzionamento dei principali comandi sopra citati per la trasmissione e ricezione di pacchetti, è stata realizzata, utilizzando come già precedentemente detto il software *Labwindows*, l'interfaccia grafica qui di seguito riportata:

- PORTANTE\_cmd

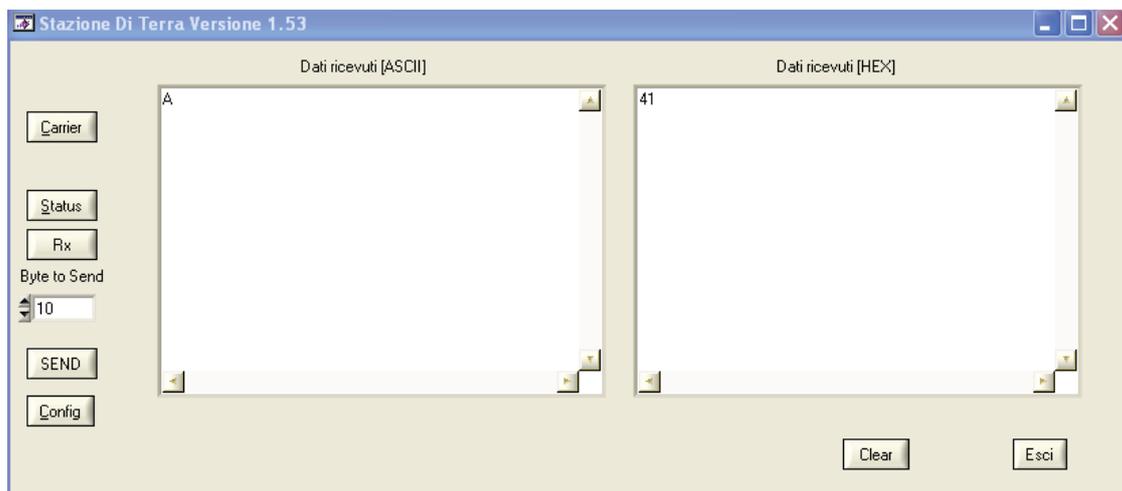


Figura 4.19. Interfaccia relativa al comando Carrier

Selezionando il pulsante *Carrier*, viene generata la portante e visualizzata sull'analizzatore di spettro.

In caso di corretta ricezione del comando viene di seguito visualizzato sull'interfaccia un ACK, ossia nella finestra dei DATI RICEVUTI[ASCII] il carattere "A" e nei DATI RICEVUTI[HEX] il valore 41.

- RX\_cmd, TX\_cmd

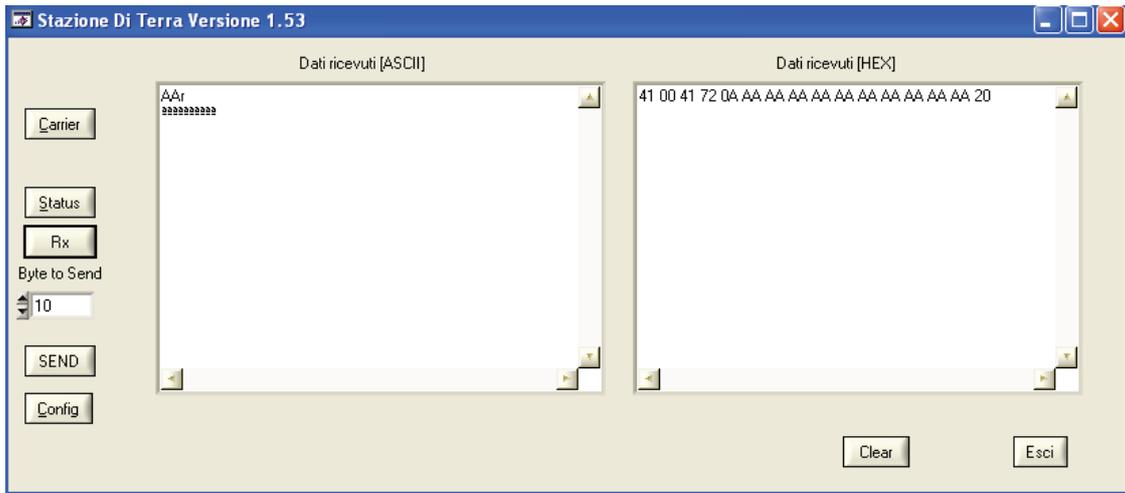


Figura 4.20. Interfaccia relativa al comando Rx

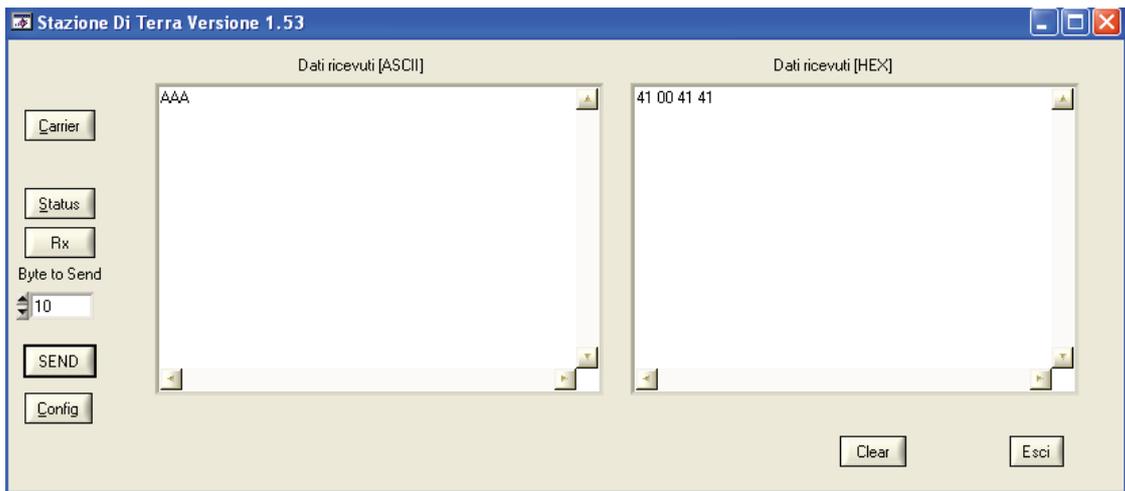


Figura 4.21. Interfaccia relativa al comando Tx

Dopo aver selezionato nella finestra a tendina “Byte to Send” il numero di byte da trasmettere :

1. Viene selezionato il pulsante *Rx* e di seguito viene visualizzato un ACK relativo al corretto stato del ricevitore.
2. Viene selezionato il pulsante *Tx* e di seguito viene visualizzato un ACK relativo al corretto stato del trasmettitore.
3. Il pacchetto viene inviato, e dal lato del ricevitore viene controllato il crc.

In caso di crc corretto viene visualizzato un ACK sia dal lato del ricevitore sia dal lato del trasmettitore.

4. A questo punto il pacchetto viene ricevuto dalla stazione terminale ricevente e di seguito viene visualizzato l'intero pacchetto con il relativo comando, lunghezza, payload, crc.

Nella finestra dei DATI RICEVUTI[HEX] rispettivamente i valori : “72”, “0A”, “AA AA AA AA AA AA AA AA AA”, “20” con i corrispettivi caratteri nella finestra dei DATI RICEVUTI[ASCII].

In caso di corretta ricezione del pacchetto viene ricevuto e visualizzato un ACK dalla stazione terminale trasmittente.

- *CONFIG\_cmd*

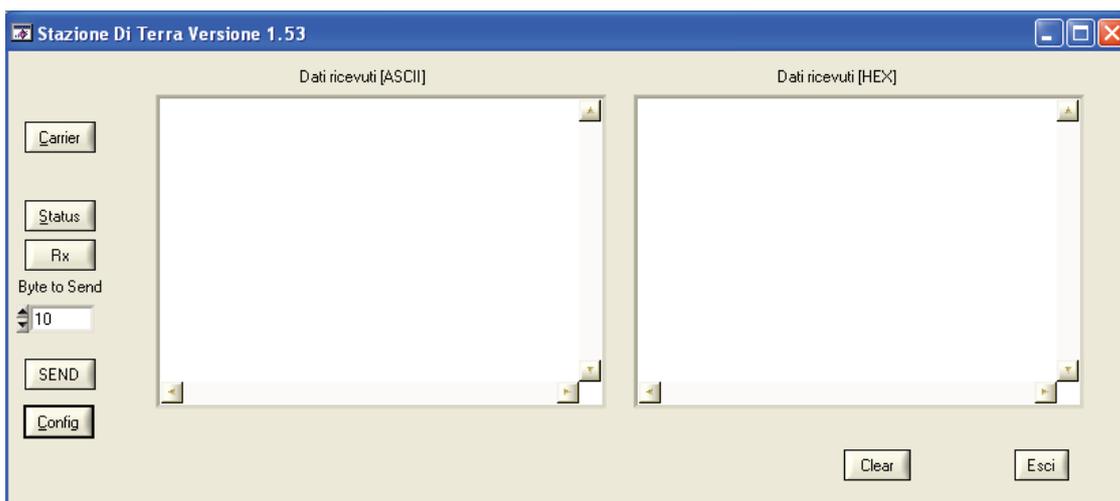


Figura 4.22. Interfaccia relativa al comando CONFIG

Selezionando il pulsante *CONFIG* appare il pannello sottostante all'interno del quale è possibile impostare la configurazione dei registri.

Una volta sottomessa la configurazione selezionando il pulsante *OK* per la configurazione scelta o il pulsante *DEFAULT* per lasciare la configurazione di default, questa viene caricata a bordo del chip.

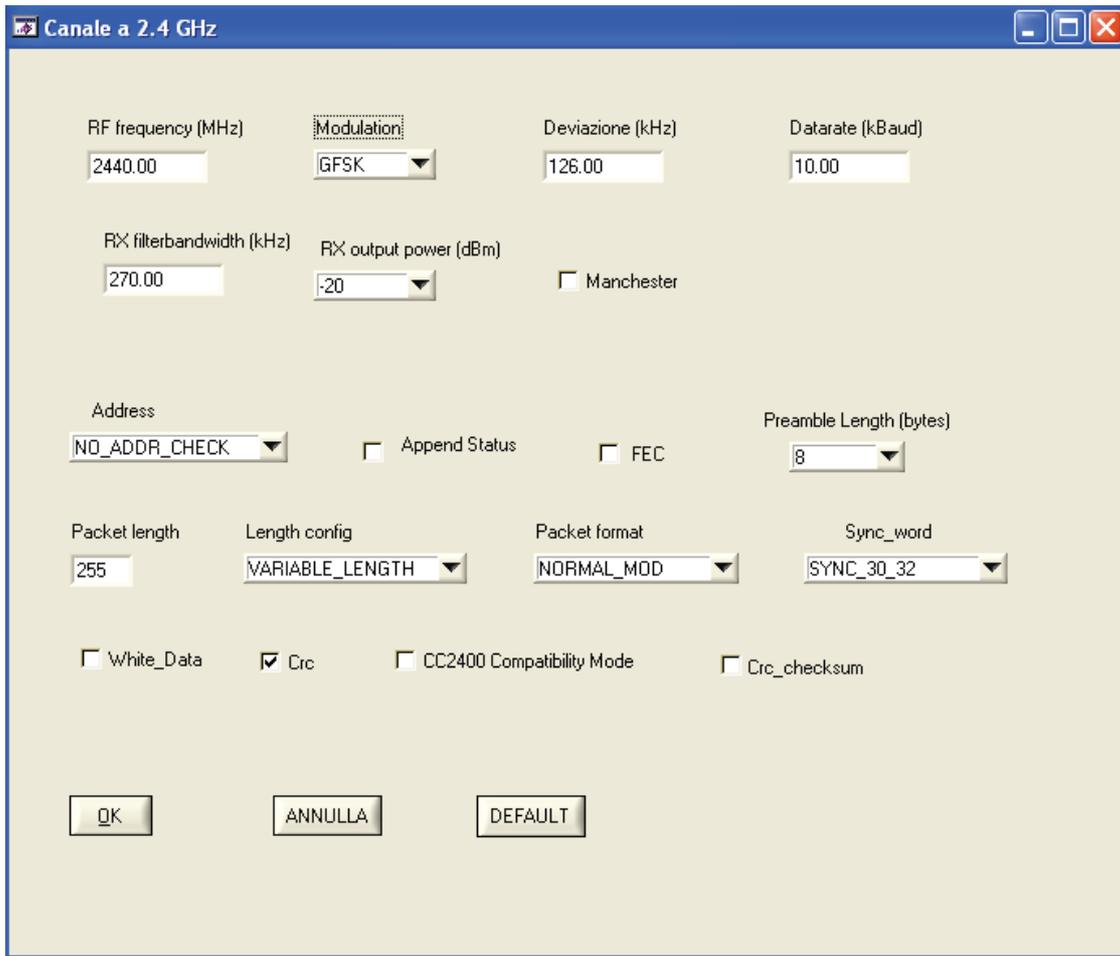


Figura 4.23.

## 4.6 Analisi dell'effetto Doppler

L'aspetto del fenomeno fisico, noto come effetto doppler, che va ad interessare il progetto, è la dispersione di frequenza che può risultare dal moto relativo tra il trasmettitore e il ricevitore.

Nelle comunicazioni satellitari attraverso i satelliti in orbita LEO le stazioni di terra stimano un rilevante doppler, che deve essere compensato per rendere possibile la comunicazione.

Supponiamo che sia il trasmettitore che il ricevitore si muovano ad una velocità costante e che il moto avvenga lungo la retta che li congiunge, ossia colinearmente con la radiazione elettromagnetica. Nel dominio dell'involuppo complesso il segnale ricevuto a distanza  $x = x_0 + vt$  dal trasmettitore sarà dato da:

$$r(t) = A(x_0 + vt)e^{[j2\pi f_0(t - \frac{x_0 + vt}{c})]} = A(x_0 + vt)e^{[-j2\pi f_0 \frac{x_0}{c}]} e^{[j2\pi f_0(1 - \frac{v}{c})t]}$$

dove l'ultimo fattore evidenzia che la frequenza ricevuta è espressa da:

$$f_r = f_0(1 - \frac{v}{c})$$

di conseguenza la deviazione di frequenza doppler è:

$$f_D = f_r - f_0 = -f_0 \frac{v}{c}$$

e la relazione tra la deviazione doppler e la velocità di allontanamento dalla sorgente sarà:

$$\frac{v}{c} = -\frac{f_D}{f_0}$$

Se la direzione del moto non è costante si formerà un certo angolo  $\psi$  tra il vettore  $v$  e la congiungente il trasmettitore e il ricevitore, e la deviazione sarà in generale:

$$f_D = -f_0 \frac{v}{c} \cos \psi$$

Quindi la deviazione doppler è una variazione offset della portante, che si modifica con la velocità e direzione del mobile. Ogni ricevitore mobile dovrebbe avere la capacità di ricevere un segnale con un offset di frequenza e cercare di compensare queste variazioni quando modifica la sua velocità e direzione di spostamento.

In riferimento al progetto la  $f_D$  vale:

$$f_D = \frac{v}{c} f_0 = 61 \text{ kHz}$$

in cui:

- $f_0 = 2.44 \text{ GHz}$
- $v = \text{velocità del satellite Picpot pari a } 7.4546 \text{ (Km/s)}$
- $c = \text{velocità della luce pari a } 3 \cdot 10^8 \text{ (m/s)}$

A questo punto facendo variare la  $f_0$  nell'intervallo  $[-61; +61] \text{ kHz}$  rispetto ai diversi valori della larghezza di banda del canale ricevente, per mezzo dell'analizzatore di spettro, si potrebbe valutare di quanto la portante sia influenzata dalla variazione di frequenza e quindi la comunicazione.

Dopo aver verificato la possibilità di poter valutare l'effetto doppler attraverso :

- il controllo della risoluzione massima di frequenza del transceiver pari a 412 Hz, quindi compresa nello intervallo  $[-61; +61] \text{ kHz}$ ,

- il calcolo della differenza tra la portante generata alla frequenza di trasmissione  $f_0$  pari a 2.440.000 kHz e  $f_0 + 61$  kHz e la stessa tra  $f_0$  sempre pari a 2.440.000 kHz e  $f_0 - 61$  kHz, che è risultata di circa 70 kHz, quindi prossima all'intervallo desiderato

sono state eseguite delle misure facendo variare la frequenza in trasmissione rispetto a quella in ricezione.

Partendo dai valori di frequenza compresi nell'intervallo  $[-61; +61]$  kHz e proseguendo oltre fino a condizione di possibile ricezione, relativamente alle diverse tipologie di modulazione e ai diversi valori di data rate, larghezza di banda del canale di ricezione e potenza d'uscita ( $P = -20$  dBm, valore medio e  $P = 1$  dBm, valore massimo), i risultati hanno portato alla conclusione che :

lavorando con una larghezza di banda del canale di ricezione non troppo elevata è migliore la sensibilità di tracking e di conseguenza la qualità del segnale ricevuto.

Viceversa lavorando con una larghezza di banda del canale di ricezione maggiore, la sensibilità di tracking è decisamente minore.

Le misure effettuate sono state:

1. Data Rate = 10 kBaud, Deviation = 126 kHz, GFSK, RX filterbandwidth = 270 kHz

$P_{out}$	$f_0 + \Delta f$	Rx
-20 dBm	$[f_0 + 10 \text{ kHz} \div f_0 + 50 \text{ kHz}]$	SI
-20 dBm	$f_0 + 60 \text{ kHz}$	NO
1 dBm	$f_0 + 60 \text{ kHz}$	SI
1 dBm	$f_0 + 70 \text{ kHz}$	NO

$P_{out}$	$f_0 - \Delta f$	Rx
-20 dBm	$[f_0 - 10 \text{ kHz} \div f_0 - 50 \text{ kHz}]$	SI
-20 dBm	$f_0 - 60 \text{ kHz}$	NON BENE
-20 dBm	$f_0 - 70 \text{ kHz}$	NON BENE
-20 dBm	$f_0 - 80 \text{ kHz}$	NO
1 dBm	$f_0 - 60 \text{ kHz}$	SI
1 dBm	$[f_0 - 70 \text{ kHz} \div f_0 - 90 \text{ kHz}]$	NON BENE
1 dBm	$f_0 - 100 \text{ kHz}$	NO

2. Data Rate = 10 kBaud, Deviation = 126 kHz, GFSK, RX filterbandwidth = 500 kHz

$P_{out}$	$f_0 + \Delta f$	Rx
-20 dBm	$[f_0 + 10 \text{ kHz} \div f_0 + 100 \text{ kHz}]$	SI
-20 dBm	$f_0 + 110 \text{ kHz}$	NO
1 dBm	$f_0 + 110 \text{ kHz}$	SI
1 dBm	$f_0 + 120 \text{ kHz}$	SI
1 dBm	$f_0 + 130 \text{ kHz}$	NO

$P_{out}$	$f_0 - \Delta f$	Rx
-20 dBm	$[f_0 - 10\text{ kHz} \div f_0 - 140\text{ kHz}]$	SI
-20 dBm	$[f_0 - 150\text{ kHz} \div f_0 - 180\text{ kHz}]$	NON BENE
-20 dBm	$f_0 - 190\text{ kHz}$	NO
1 dBm	$f_0 - 150\text{ kHz}$	SI
1 dBm	$f_0 - 160\text{ kHz}$	SI
1 dBm	$[f_0 - 170\text{ kHz} \div f_0 - 210\text{ kHz}]$	NON BENE
1 dBm	$f_0 - 220\text{ kHz}$	NO

3. Data Rate = 10 kBaud, Deviation = 38 kHz, GFSK, RX filterbandwidth = 232 kHz

$P_{out}$	$f_0 + \Delta f$	Rx
-20 dBm	$[f_0 + 10\text{ kHz} \div f_0 + 80\text{ kHz}]$	SI
-20 dBm	$f_0 + 90\text{ kHz}$	NO
1 dBm	$f_0 + 90\text{ kHz}$	NO

$P_{out}$	$f_0 - \Delta f$	Rx
-20 dBm	$[f_0 - 10\text{ kHz} \div f_0 - 70\text{ kHz}]$	SI
-20 dBm	$f_0 - 80\text{ kHz}$	NON BENE
-20 dBm	$f_0 - 90\text{ kHz}$	NO
1 dBm	$f_0 - 80\text{ kHz}$	NON BENE
1 dBm	$f_0 - 90\text{ kHz}$	NO

4. Data Rate = 10 kBaud, Deviation = 38 kHz, GFSK, RX filterbandwidth = 500 kHz

$P_{out}$	$f_0 + \Delta f$	Rx
-20 dBm	$[f_0 + 10\text{ kHz} \div f_0 + 120\text{ kHz}]$	SI
-20 dBm	$[f_0 + 130\text{ kHz} \div f_0 + 160\text{ kHz}]$	NON BENE
-20 dBm	$f_0 + 170\text{ kHz}$	NO
1 dBm	$[f_0 + 130\text{ kHz} \div f_0 + 160\text{ kHz}]$	SI
1 dBm	$f_0 + 170\text{ kHz}$	NO

$P_{out}$	$f_0 - \Delta f$	Rx
-20 dBm	$[f_0 - 10\text{ kHz} \div f_0 - 110\text{ kHz}]$	SI
-20 dBm	$f_0 - 120\text{ kHz}$	NO
1 dBm	$f_0 - 120\text{ kHz}$	NO

5. Data Rate = 250 kBaud, Deviation = 0 kHz, MSK, RX filterbandwidth = 540 kHz

$P_{out}$	$f_0 + \Delta f$	Rx
-20 dBm	$[f_0 + 10 \text{ kHz} \div f_0 + 90 \text{ kHz}]$	SI
-20 dBm	$f_0 + 100 \text{ kHz}$	NON BENE
-20 dBm	$f_0 + 110 \text{ kHz}$	NON BENE
-20 dBm	$f_0 + 120 \text{ kHz}$	NO
1 dBm	$f_0 + 100 \text{ kHz}$	SI
1 dBm	$f_0 + 110 \text{ kHz}$	NON BENE
1 dBm	$f_0 + 120 \text{ kHz}$	NON BENE
1 dBm	$f_0 + 130 \text{ kHz}$	NO

$P_{out}$	$f_0 - \Delta f$	Rx
-20 dBm	$[f_0 - 10 \text{ kHz} \div f_0 - 90 \text{ kHz}]$	SI
-20 dBm	$f_0 - 100 \text{ kHz}$	NO
1 dBm	$f_0 - 100 \text{ kHz}$	NO

6. Data Rate = 500 kBaud, Deviation = 0 kHz, MSK, RX filterbandwidth = 812 kHz

$P_{out}$	$f_0 + \Delta f$	Rx
-20 dBm	$[f_0 + 10 \text{ kHz} \div f_0 + 130 \text{ kHz}]$	SI
-20 dBm	$f_0 + 140 \text{ kHz}$	NO
1 dBm	$f_0 + 140 \text{ kHz}$	NON BENE
1 dBm	$f_0 + 150 \text{ kHz}$	NO

$P_{out}$	$f_0 - \Delta f$	Rx
-20 dBm	$[f_0 - 10 \text{ kHz} \div f_0 - 40 \text{ kHz}]$	SI
-20 dBm	$f_0 - 50 \text{ kHz}$	NO
1 dBm	$f_0 - 50 \text{ kHz}$	NO

## 4.7 Probabilità d'errore per un sistema di comunicazione binario

La qualità di un segnale ricevuto potrebbe essere deteriorata da molti fattori, tra cui il rumore (*noise*).

Il rumore è un segnale, casuale nel tempo, che si sovrappone al segnale utile e pertanto ne influenza l'integrità. La Fig. 4.24 mostra il diagramma a blocchi di un generico sistema di comunicazione binario con particolare attenzione alla struttura del ricevitore.

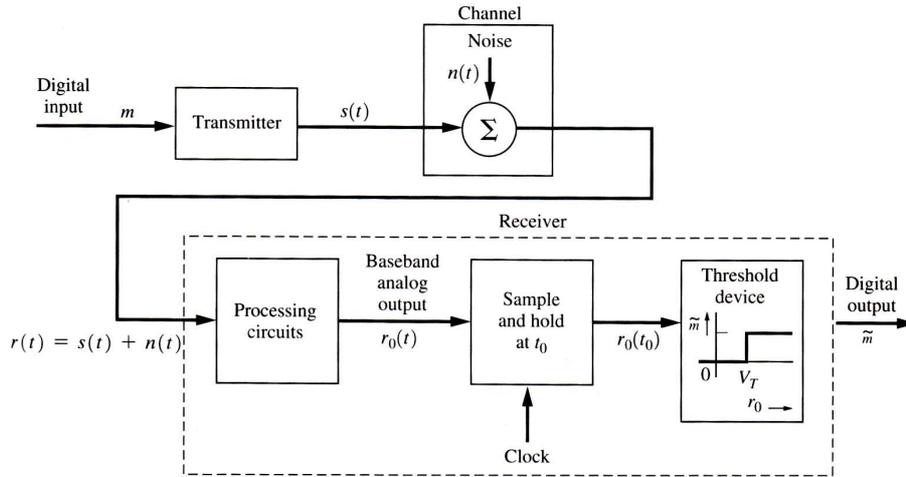


Figura 4.24. Sistema di comunicazione binario

Il segnale ricevuto  $r(t)$  è la somma del segnale trasmesso  $s(t)$  più una componente di rumore  $n(t)$ ,  $r(t) = s(t) + n(t)$ . Se  $r(t)$  è un segnale in banda base il *processing circuits* sarà un filtro passa-basso con un'appropriata amplificazione. Nel caso invece di segnale in banda passante, come OOK, BPSK, FSK, il processing circuits sarà un ricevitore a supereterodina come per il CC2510. Per entrambi i casi in uscita si ha un segnale analogico in banda base  $r_0(t)$ . Questo segnale viene campionato al tempo di clock  $t = t_0 + nT$  nei campioni  $r_0(t_0 + nT)$  che passano attraverso un comparatore producendo in uscita la sequenza di dati binari  $m(t)$ .

In generale il procedimento per il calcolo del *bit error rate* (BER) consta dei seguenti passaggi: sia  $T$  il tempo di trasmissione di un bit di dato; il segnale trasmesso sull'intervallo di trasmissione di un bit  $(0, T)$  sarà:

$$s(t) = \begin{cases} s_1(t), 0 < t \leq T, & \text{se il bit trasmesso è un "1"} \\ s_2(t), 0 < t \leq T, & \text{se il bit trasmesso è uno "0"} \end{cases}$$

Se  $s_1(t) = -s_2(t)$ ,  $s(t)$  è detto segnale antipodale.

Il segnale  $s(t)$ , corrotto dal rumore, fornisce in uscita dal processing circuits il segnale:

$$r_0(t) = \begin{cases} r_{01}(t), 0 < t \leq T, & \text{se il bit tx è un "1"} \\ r_{02}(t), 0 < t \leq T, & \text{se il bit tx è uno "0"} \end{cases}$$

Il segnale  $r_0(t)$  viene campionato al tempo  $t_0$  nell'intervallo  $(0, T)$ :

$$r_0(t_0) = \begin{cases} r_{01}(t_0), 0 < t \leq T, & \text{se il bit tx è un "1"} \\ r_{02}(t_0), 0 < t \leq T, & \text{se il bit tx è uno "0"} \end{cases}$$

$r_0(t_0)$  è una variabile casuale, che ha una distribuzione continua perchè affetta da rumore. Quindi si può semplicemente scrivere:

$$r_0 = r_0(t_0) = \begin{cases} r_1, & \text{se il bit tx è un "1"} \\ r_2, & \text{se il bit tx è uno "0"} \end{cases}$$

$r_0$  è detta variabile statistica.

Per il momento si valuta la densità di probabilità delle due variabili casuali  $r_0 = r_{01}$  e  $r_0 = r_{02}$ .

Si tratta di una densità di probabilità condizionata, in quanto essa dipende rispettivamente dalla trasmissione di un bit “1” o di un bit “0”.

Si ha:

$$f(r_0|s_1tx), \text{ per } r_0 = r_{01}$$

$$f(r_0|s_2tx), \text{ per } r_0 = r_{02}$$

La densità di probabilità dipende dalle caratteristiche del rumore, dal tipo di processing circuits utilizzato e dal tipo di segnale trasmesso.

Nell’ipotesi in cui il processing circuits filtra solo il segnale  $s(t)$  non affetto da rumore,  $r_0$  sarà maggiore della tensione di soglia del comparatore  $V_T$  se viene trasmesso un bit “1” o minore di  $V_T$  se viene trasmesso un bit 0.

Quando invece il segnale  $s(t)$  è corrotto da rumore ne viene valutato l’errore come segue:

$$P(error|s_1tx) = \int_{-\infty}^{V_T} f(r_0|s_1)dr_0 \text{ se } r_0 < V_T \text{ e il bit tx è un "1"},$$

$$P(error|s_2tx) = \int_{V_T}^{\infty} f(r_0|s_2)dr_0 \text{ se } r_0 > V_T \text{ e il bit tx è uno "0"}.$$

Poichè i due eventi corrispondenti alla trasmissione di un bit “1” o “0” sono disgiunti si può scrivere utilizzando la probabilità condizionata che il BER è:

$$P_e = P(error|s_1tx)P(s_1tx) + P(error|s_2tx)P(s_2tx)$$

sostituendo:

$$P_e = P(s_1tx) \int_{-\infty}^{V_T} f(r_0|s_1)dr_0 + P(s_2tx) \int_{V_T}^{\infty} f(r_0|s_2)dr_0$$

I simboli trasmessi sono statisticamente indipendenti e la probabilità  $P(s_1sent)$  di trasmettere un “1” è uguale alla probabilità di trasmettere uno “0”,  $P(s_2sent)$ , quindi:

$$P(s_1tx) + P(s_2tx) = 1$$

e

$$P(s_1tx) = P(s_2tx) = \frac{1}{2}$$

da cui si ricava che:

$$P_e = \frac{1}{2} \int_{-\infty}^{V_T} f(r_0|s_1) dr_0 + \frac{1}{2} \int_{V_T}^{\infty} f(r_0|s_2) dr_0$$

## 4.8 Test del CC2510

Anche nel nostro caso è stato implementato il codice atto a valutare la probabilità d'errore in ricezione di un elevato numero di pacchetti trasmessi.

Le specifiche relative all'esecuzione del programma di test del CC2510 sono state:

- dopo aver testato la connessione, comunicare, fornendo gli opportuni comandi al CC2510(Tx) e CC2510(Rx) utilizzando due diverse porte seriali
- rilevare l'errore di crc via radio dei pacchetti
- permettere all'utente di settare i parametri per la configurazione dei registri di entrambi i tranceiver
- permettere all'utente di impostare il numero di pacchetti da inviare e la rispettiva dimensione(da 1 a 255 byte), e di visualizzare alla fine del test le statistiche sui pacchetti ricevuti
- le statistiche devono riportare i seguenti valori:
  1. numero di pacchetti trasmessi
  2. numero di pacchetti ricevuti in modo corretto o errato
  3. numero di bit ricevuti errati
  4. il BER calcolato come

$$\frac{\text{numero di bit errati ricevuti}}{\text{numero totale di bit trasmessi}}$$

Le operazioni eseguite dal software sono quindi:

- Si effettua il test di connessione con il CC2510(Tx)
- Si effettua il test di connessione con il CC2510(Rx)
- Si configurano i registri del CC2510(Tx)(facoltativo)

- Si configurano i registri del CC2510(Rx)(facoltativo)
- Viene inviato il comando di trasmissione al CC2510(Tx)
- Viene inviato il comando di ricezione al CC2510(Rx)
- In caso di ricezione del pacchetto dal CC2510(Rx):
  1. Si richiede lo status byte dal CC2510(Rx)
  2. Se il pacchetto ricevuto è corretto si incrementano i pacchetti trasmessi
  3. Se il pacchetto ricevuto è errato si incrementano sia i pacchetti trasmessi sia i pacchetti ricevuti errati, e viene calcolato il numero di bit di differenza
- In caso di mancata ricezione:
  1. Vengono incrementati i pacchetti trasmessi e quelli ricevuti errati
  2. Viene rinviato il comando di ricezione al CC2510(Rx), e così fino alla fine dei pacchetti

Di seguito è riportata la macro che consente il calcolo del BER e del PER (*Packet Error Rate*):

```
unsigned char controllo_bit (unsigned char dato1, unsigned char dato2)
{
    unsigned char diff;
    unsigned char h;

    int quanti;

    diff = (dato1)^(dato2);
    quanti =0;
    for(h=0;h<8;h++)
    {
        if(( diff & (0x01 << h)) != 0)
            quanti++;
    }
    return quanti;
}
```

Il PER si relaziona al BER con:

$$PER = 1 - (1 - BER)^{packet\ length}$$

L'utente dall'esterno deve effettuare le seguenti operazioni:

- Selezionare le porte seriali COM, che vengono utilizzate, e verificarne la comunicazione con il pulsante *TEST*
- Selezionare il numero di pacchetti e la dimensione
- Selezionare la configurazione dei registri per uno o entrambi i transceiver o mantenere la configurazione di default
- Avviare il test con il pulsante *START* e leggerne il risultato

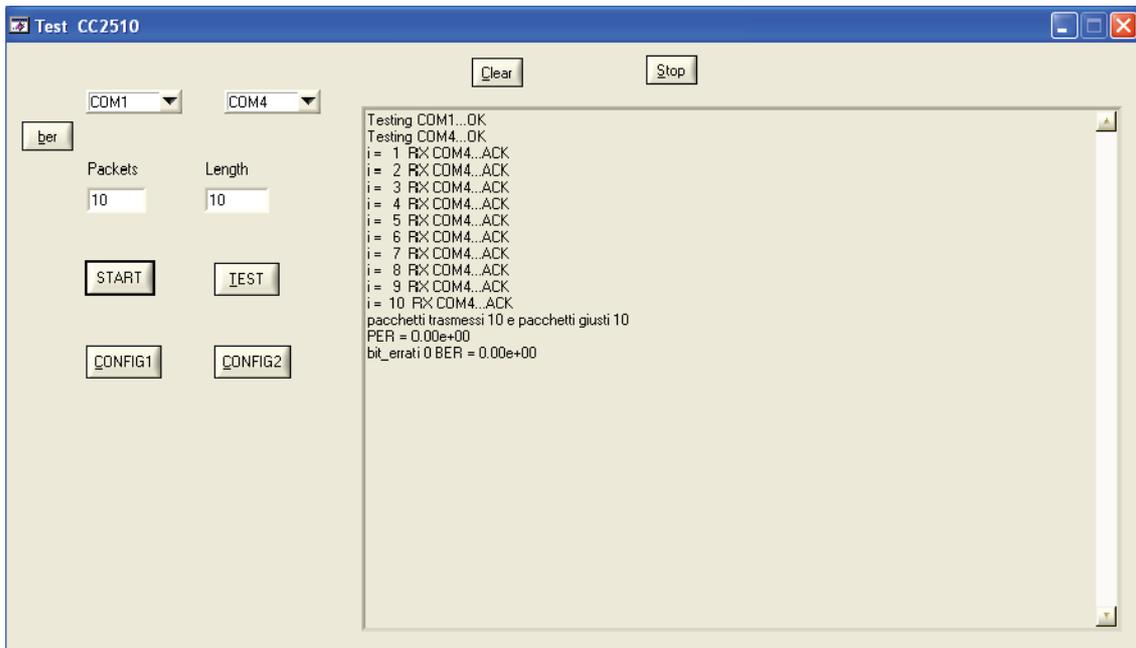


Figura 4.25. Programmer CC1010

Di seguito vengono mostrate alcune misure sulla probabilita' d'errore effettuate in relazione soprattutto ai diversi valori di potenza utilizzata nella trasmissione di pacchetti.

Le misure sono state valutate cercando di andare a lavorare sulle condizioni operative identificate come “worst case analysis”, quindi nei casi :

- Data rate = 2.4 k*Baud*, Deviation = 38 kHz, FSK, RX filterbandwidth = 203 kHz, Packets = 100, Length = 20

$P_{Tx}$	Bit errati	PER
-75 dBm	2	1.25E - 04
-125 dBm	16.000	1.00E + 00

- Data rate = 10 k*Baud*, Deviation = 38 kHz, RX filterbandwidth = 232 kHz, Packets = 500, Length = 100

$P_{Tx}$	Modulation	Bit errati	PER
-75 dBm	FSK	0	0.00E + 00
-125 dBm	GFSK	0	0.00E + 00

- Data rate = 10 k*Baud*, Deviation = 126 kHz, RX filterbandwidth = 270 kHz, Packets = 500, Length = 100

$P_{Tx}$	Bit errati	PER
-90 dBm	0	0.00E + 00
-103.7 dBm	0	0.00E + 00
-104 dBm	0	0.00E + 00
-106.2 dBm	2	5.00E - 06
-108.2 dBm	3	7.50E - 06
-114 dBm	0	0.00E + 00
-114 dBm	4	1.00E - 05
-115 dBm	1500	3.75E - 03
120.4 dBm	6028	1.51E - 02

- Data rate = 10 k*Baud*, Deviation = 126 kHz, RX filterbandwidth = 270 kHz, Packets = 500, Length = 255

$P_{Tx}$	Bit errati	PER
-55 dBm	4	3.92E - 06
-125 dBm	204.000	1.00E + 00

Come si può notare, con una potenza maggiore in trasmissione il PER é minore, mentre con una potenza minore il PER presenta un aumento improvviso con un decadimento delle prestazioni.

# Capitolo 5

## Progetto dell'amplificatore di potenza: SZM-2166Z

### 5.1 Dimensionamento della potenza in trasmissione

Le prestazioni di un sistema di comunicazione dipendono fortemente dalla qualità del segnale ricevuto, in relazione alle varie perdite e al range di potenza in gioco.

Nel caso di trasmissione nello spazio libero, la funzione di trasferimento di potenza, tra un'antenna trasmittente e un'antenna ricevente ad una certa distanza  $D$ , è l'equazione di trasmissione di Friis:

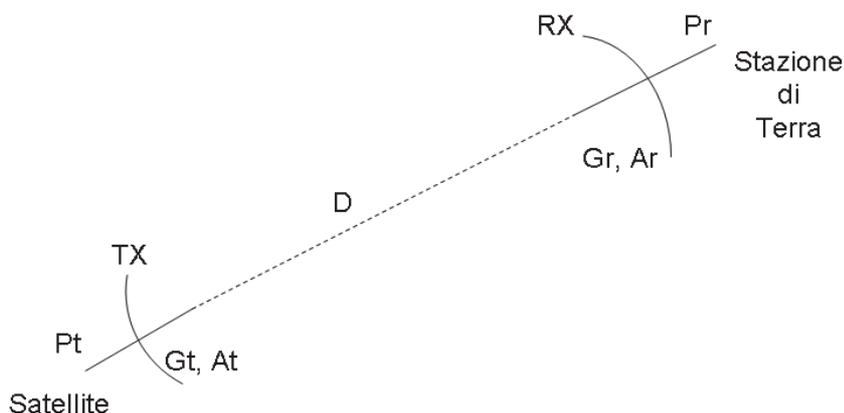


Figura 5.1. Trasmissione nello spazio libero nel collegamento downlink

$$\frac{P_{Rx}}{P_{Tx}} = G_{AT}G_{AR}\left(\frac{\lambda}{4\pi D}\right)^2\frac{1}{\alpha}$$

in cui:

- $P_{Tx}$  è la potenza irradiata dall'antenna trasmittente
- $P_{Rx}$  è la potenza irradiata dall'antenna ricevente
- $G_{AT}$  è il guadagno d'antenna trasmittente
- $G_{AR}$  è il guadagno d'antenna ricevente
- $\left(\frac{\lambda}{4\pi D}\right)^2$  è il guadagno di potenza nello spazio libero espresso anche come

$$\frac{1}{\text{attenuazione}(\alpha)}$$

che corrisponde al rapporto  $\frac{P_{Tx}}{P_{Rx}}$  nel caso di radiatori isotropici.

Si può dedurre che l'attenuazione decresce al crescere della frequenza utile.

In decibel l'equazione diventa:

$$\left(\frac{P_{Rx}}{P_{Tx}}\right) \text{dB} = (G_{AT}) \text{dB} + (G_{AR}) \text{dB} - (\alpha) \text{dB}$$

con

$$\alpha = 20 \log\left(\frac{4\pi D}{\lambda}\right)$$

Da questa equazione è possibile dimensionare la potenza necessaria in trasmissione, nel collegamento downlink, alla frequenza di 2.4 GHz:

$$(P_{Tx}) \text{dB} = (P_{Rx}) \text{dB} - (G_{AT}) \text{dB} - (G_{AR}) \text{dB} + (\alpha) \text{dB}$$

Dalle specifiche di progetto:

- Utilizzando un'antenna di tipo patch per il satellite e un'antenna di tipo parabolico per la stazione di terra, i guadagni saranno rispettivamente:
  1.  $G_{AT} = 3 \text{ dBic}$
  2.  $G_{AR} = 35 \text{ dBic}$

- Valutando una lunghezza d'onda pari a :

$$\lambda = \frac{c}{f} = \frac{3 \cdot 10^8 \text{ m/s}}{2.44 \text{ GHz}} = 0.123 \text{ m}$$

e calcolando la distanza  $D$  del satellite, in orbita circolare e all'orizzonte rispetto alla terra, applicando il teorema di Pitagora:

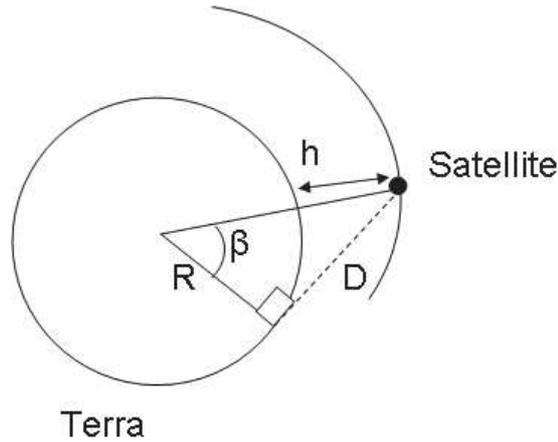


Figura 5.2. Distanza satellite-terra

$$D = \sqrt{(R + h)^2 - R^2} = 2767 \text{ km}$$

in cui:

1.  $h$  è la distanza del satellite dalla superficie terrestre pari a circa  $600 \text{ km}$
2.  $R$  è il raggio della terra pari a  $6.378.17 \text{ km}$

si avrà un'attenuazione  $\alpha$  pari a circa  $169 \text{ dB}$

- Assumendo come valore minimo di potenza, che può arrivare al ricevitore, una  $P_{Rx}$  pari a  $-98 \text{ dBm}$ , ossia pari al minimo valore di sensitivity che può caratterizzare il ricevitore del CC2510 con una banda di canale di  $232 \text{ kHz}$ , un data rate di  $10 \text{ kbps}$  e una modulazione GFSK.

il valore di potenza in trasmissione risulta quindi :

$$P_{Tx} = (-98) \text{ dBm} - (35) \text{ dB} - (3) \text{ dB} + (169) \text{ dB} = 33 \text{ dBm}$$

Sulla base di questo valore, considerando che il transceiver eroga in uscita una potenza al massimo di  $1 \text{ dBm}$ , è stata essenziale la ricerca di un amplificatore di potenza in grado di fornire la potenza calcolata in trasmissione.

## 5.2 L'amplificatore di potenza SZM-2166Z

L'amplificatore di potenza, scelto in funzione delle specifiche di progetto, è stato l'*SMZ-2166Z* della *Sirenza Microdevices*.

Si tratta di un amplificatore ad alta linearità di potenza progettato per applicazioni 802.16 WiMax e, per la sua flessibilità, anche per applicazioni 802.11b/g WiFi.

Il componente è realizzato con tecnologia HBT (Heterojunction Bipolar Transistor) con InGaP su GaAs entrambi semiconduttori usati nei dispositivi ad alta frequenza.

L'*SZM-2166Z* opera ad una frequenza compresa tra  $(2.3 \div 2.7)$  GHz con 37 dB di guadagno e può fornire ben oltre 27 dBm di potenza d'uscita lineare con un'ampiezza del vettore di errore (*EVM, error vector magnitude*) del 2.5% , con modulazione 64QAM 802.11g a 54 Mbit/s.

L'*EVM, error vector magnitude* misura la differenza tra la forma d'onda ideale e quella realmente misurata in ricezione.

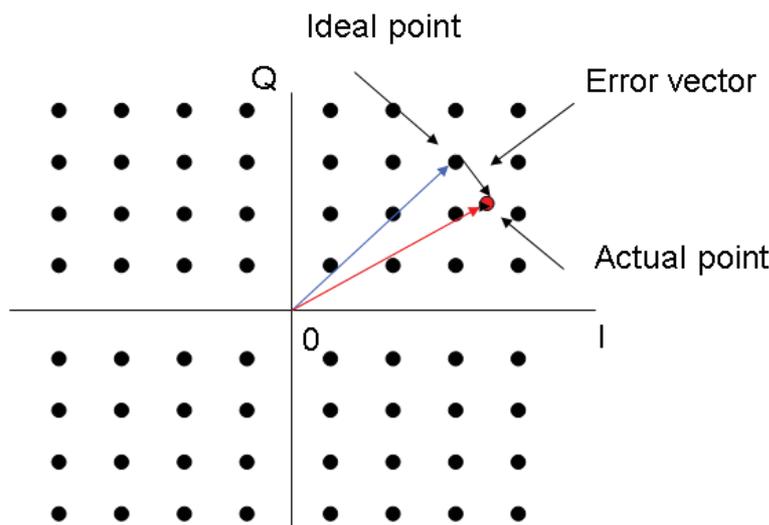


Figura 5.3. EVM, error vector magnitude per una modulazione 64QAM

L'amplificatore incorpora un circuito di bias attivo, che fornisce ai progettisti la flessibilità per ottimizzare le performance del chip in specifiche applicazioni, un rilevatore di potenza e uno switch di potenza per eventuali sviluppi di politiche energy saving durante la modalità di ricezione.

Il range di alimentazione del componente è compreso tra  $(3.0 \div 6.0)$  V.

Nel nostro caso per avere 2W di potenza in uscita l'amplificatore è alimentato solamente a 6V, e alla frequenza di 2.4GHz lo schematico della sua Evaluation Board, con i relativi componenti, è il seguente:

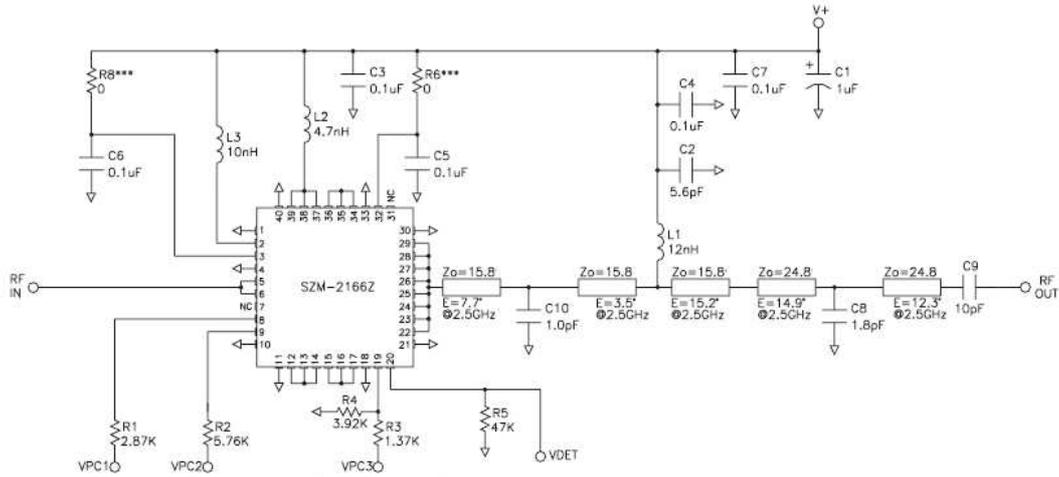


Figura 5.4. Schematico dell'evaluation board dell'SZM-2166Z

Il dispositivo è fornito con un package a basso costo di tipo QFN con dimensione  $6 \times 6 \text{ mm}$ .

### 5.3 I componenti passivi

Il progetto ha inizio con la ricerca dei componenti passivi legati ai vincoli presenti nello schematico rispettandone valori e tolleranze.

L'utilizzo di componenti di piccole dimensioni migliora il comportamento del campo a radiofrequenza.

Di conseguenza le scelte sono state fatte in funzione del principio che componenti di piccole dimensioni danno origine a minori componenti parassite, tenendo però conto della loro realizzabilità fisica.

Quindi i componenti per cui non sono stati importanti i parametri parassiti sono di dimensione  $06 \times 03 \text{ mils}$ .

Invece i componenti relativi alla parte a radiofrequenza sono di dimensione  $04 \times 02 \text{ mils}$ . In particolare si tratta delle capacità:  $C = 10 \text{ pF}$ ,  $C = 1.8 \text{ pF}$  e  $C = 10 \text{ pF}$ . Esse potevano anche essere di dimensione  $02 \times 01 \text{ mils}$  ma sarebbe stata più complicata la fase di realizzazione fisica del circuito stampato.

Solo l'induttanza  $L = 12 \text{ nH}$  è di dimensione  $08x05 \text{ mils}$  pur facendo parte dei componenti a radiofrequenza in quanto non è stato possibile trovare un'induttanza di dimensioni più piccole che fornisse la corrente necessaria.

## 5.4 Realizzazione della libreria centrale

Per la realizzazione dell'evaluation board dell'amplificatore di potenza è stata utilizzata la suite della *Mentor Graphics* con i relativi tool.

Il punto di partenza è stato creare una libreria contenente tutti i componenti inerenti al progetto con l'utilizzo del software *Library Manager*.

La libreria è realizzata nelle sottocartelle:

- *Symbols*, che contiene la rappresentazione a livello logico dei componenti: *il simbolo*, che viene realizzato mediante il tool *Symbol Editor*.

Dopo aver creato il simbolo vengono posizionati i pin indicandone il nome, il numero e la funzionalità di cui se ne elencano alcune: input, output, power, ground, analog, digital, no connect e altre.

Di seguito è riportato il simbolo dell'amplificatore di potenza:

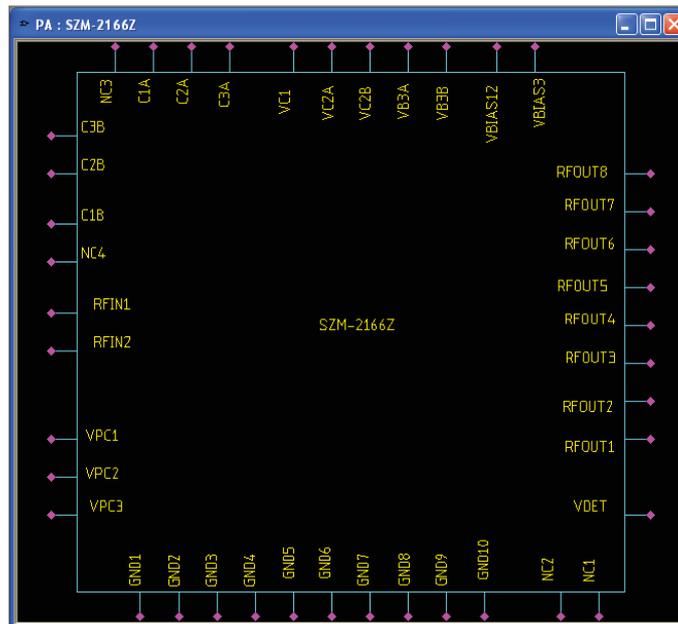


Figura 5.5. Simbolo dell'SZM-2166Z

in cui il pin 41 con funzionalità GND10 è riferito al *pad centrale* a massa.

- *Cells*, che contiene la rappresentazione a livello fisico dei componenti: le dimensioni del package, *la cella*, che viene realizzata mediante il tool *Cell Editor*.

Per poter creare la cella è necessario prima generare i relativi *padstack* mediante il tool *Padstack Editor*.

Nel caso di componenti *SMD (Surface Mount Device)* il padstack viene associato al pad del footprint definito nella sua forma e dimensione.

Nel caso, invece, di componenti *through hole* il padstack viene associato oltre al pad anche ai fori di alloggiamento definiti anch'essi nella loro forma e dimensione.

La procedura per creare la cella dell'SZM-2166Z si suddivide nelle seguenti fasi:

– Utilizzo del *Padstack Editor*:

- 1 Poichè l'amplificatore di potenza è un componente through hole il pad centrale viene generato come *custom pad* e avrà dimensione  $4.2 \times 4.2$  mm.

Di conseguenza viene creato il padstack ad esso associato e in questa fase può già essere aggiunto un foro non metallizzato, l'*hole*

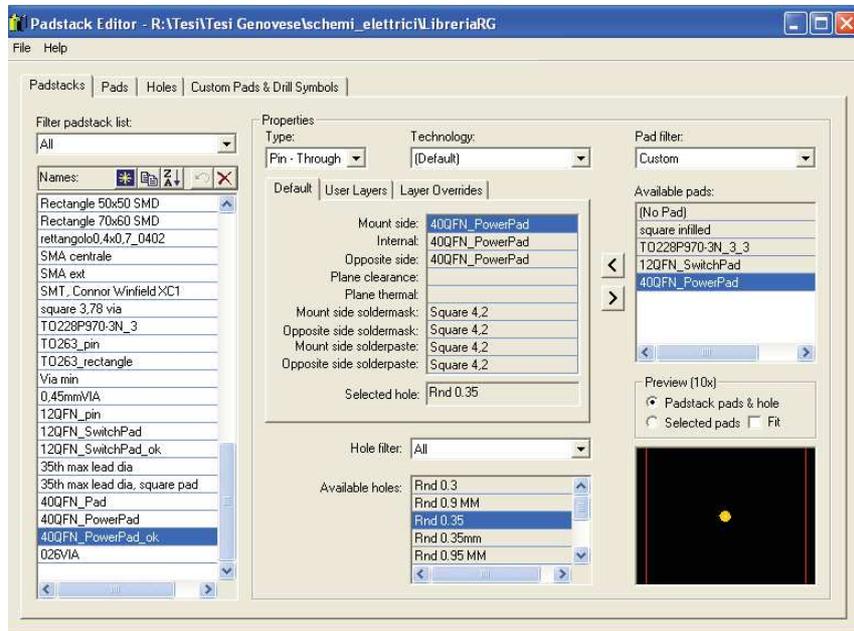


Figura 5.6. Padstack relativo al pad centrale

2 Di seguito viene generato il pad esterno, di dimensione  $0.5 \times 0.29$  mm per quanto riguarda la parte metallica della board e di dimensione  $0.5 \times 0.25$  mm per quanto riguarda la parte della *soldermask*, una vernice usata per proteggere le piste dall'ossidazione e per facilitare la saldatura.

Di conseguenza viene creato il padstack associato

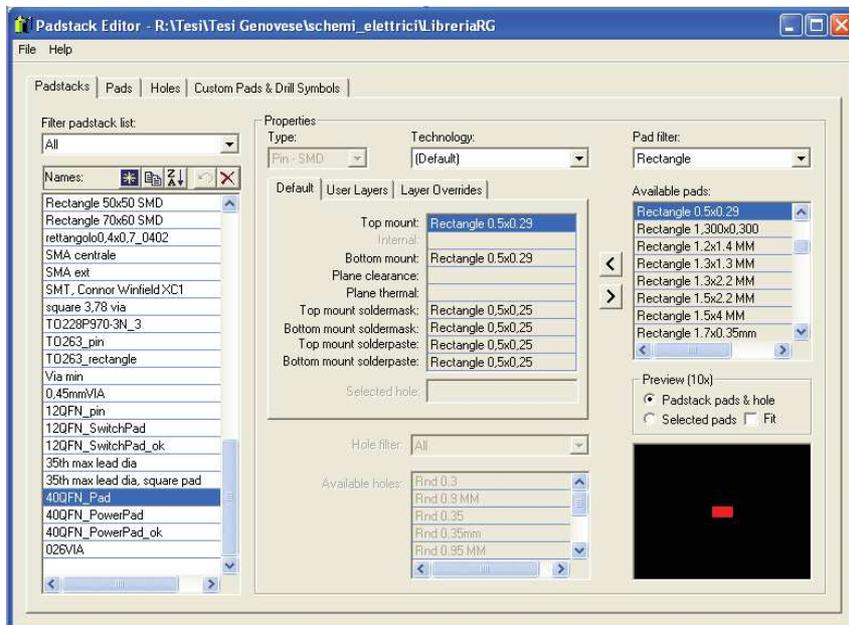


Figura 5.7. Padstack relativo al pad esterno

3 Infine viene generato il foro metallizzato , *il via*, di diametro  $0.5$  mm di cui  $0.35$  mm di hole e  $0.15$  mm di contorno metallizzato con il relativo padstack associato.

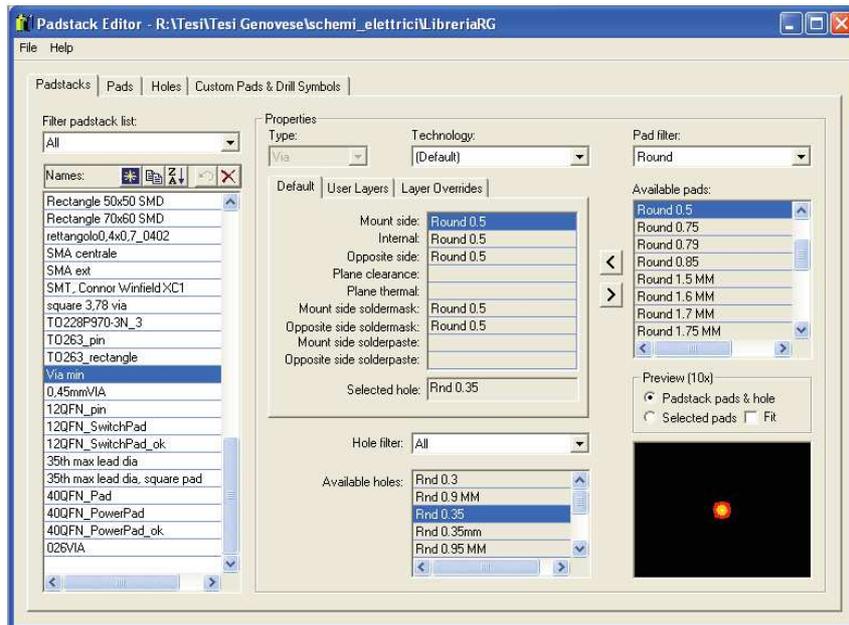


Figura 5.8. Padstack relativo al via

Nel progetto, rispettando le dovute regole fisiche realizzative, si è cercato di posizionare il maggior numero di via possibili per avere un maggior contatto elettrico e un maggior trasporto di calore.

#### – Utilizzo del *Cell Editor*

- 1 Viene etichettata la cella con il suo nome, *40QFN* e le sue caratteristiche realizzative come il numero di pin e il tipo di montaggio
- 2 Vengono posizionati i padstack relativi al pad centrale e ai pad esterni mediante il *Place pin*
- 3 Vengono posizionati i via a cui deve essere associato un nome, *un net-name*, che deve essere lo stesso nome del pad centrale a cui essi sono associati e nel nostro caso sarà *NET – 12*
- 4 Si torna nel Padstack Editor, in particolare nel custom pad e si iconizza l'opzione *fill*. Al momento del salvataggio la cella viene aggiornata automaticamente con la nuova opzione.

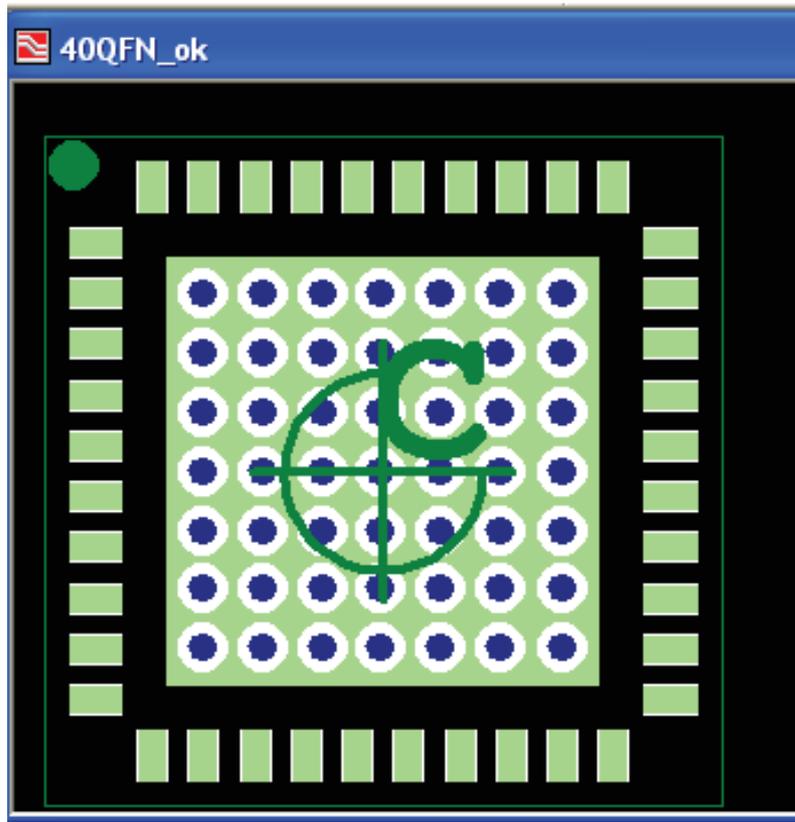


Figura 5.9. Cella dell'SZM-2166Z

- *Parts*, che contiene l'associazione dei pin tra il simbolo logico e la cella fisica.

Mediante il tool *Part Editor* l'amplificatore viene etichettato con un nome, *il Part name*, un numero, *il Part number* e un'identificativo, *il Part label*: *RFMDSZM-2166Z* in cui RFDM (*RF Micro Devices*) indica il tipo di dispositivo

Ne viene anche assegnato il rispettivo *reference designator*: *U* in quanto si tratta di un integrato.

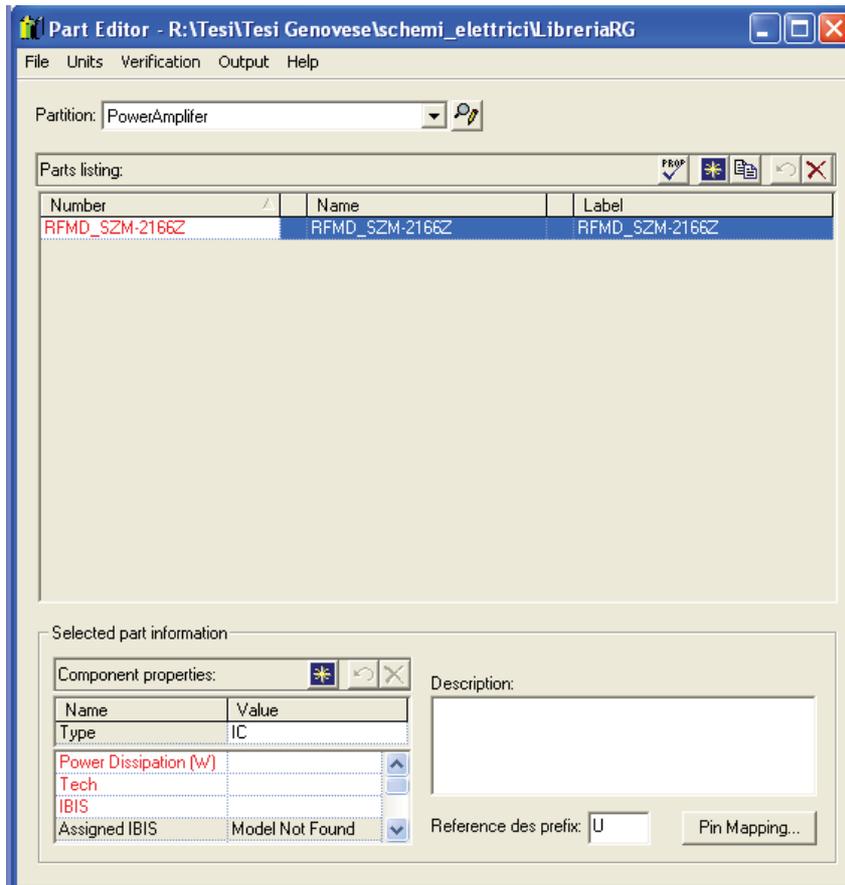


Figura 5.10. Part editor dell'SZM-2166Z

Di seguito è riportata la tabella con i reference designator corrispondenti alla diversa tipologia di componente:

Ref Designator	Componente
R	Resitore
C	Condensatore
L	Induttore
D	Diodo
U	Integrato
J	Connettore
X	Quarzo

Infine mediante il *Pin mapping* è possibile fare l'associazione dei pin ordinandoli per nome e numero.

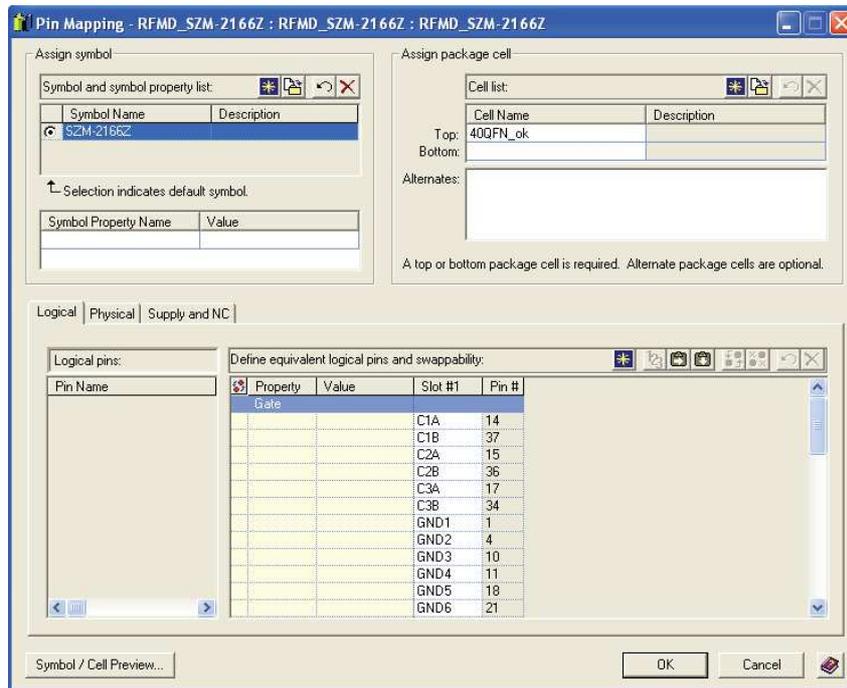


Figura 5.11. Pin mapping dell'SZM-2166Z

I simboli, le celle e le parti vengono create dopo aver generato le relative partizioni in cui saranno contenuti. Il tool utilizzato è il *Partition Editor* e la partizione in cui si trova l'SZM-2166Z è la *Power amplifier*.

Per quanto riguarda i componenti passivi dello schematico sono stati utilizzati i simboli e le celle standard già esistenti in libreria.

## 5.5 Realizzazione dello schema elettrico

Dopo la creazione della libreria si può procedere alla realizzazione del disegno dello schema elettrico mediante il software *Design Capture*.

Il progetto, inizialmente, viene associato alla libreria appena creata da cui importare i simboli inerenti.

I simboli vengono posizionati nel disegno col *Place device* e collegati elettricamente tra di loro tramite il *Place wire*.

Ad ogni collegamento è associato un net-name che deve combaciare con quello delle piste di collegamento del layout.

Anche un collegamento connesso dallo stesso connettore *intrapage* deve avere lo stesso net-name.

Il software consente anche di inserire simboli comuni a tutti gli schematici come la massa analogica,  $AGND$ , e le alimentazioni,  $V_{cc}$ ,  $V_{pc1}$ ,  $V_{pc2}$ ,  $V_{pc3}$ ,  $V_{det}$ .

Al termine del disegno si procede alla compilazione mediante il tool *Compile, CDB (Common Database Compiler)* e alla fase di debug per eliminare eventuali errori.

In caso di compilazione corretta, prima di procedere alla fase di realizzazione del layout, il progetto viene impacchettato, ossia i simboli logici vengono accorpate con le rispettive celle fisiche comprendendo anche i rispettivi collegamenti. Questa operazione avviene con il *Package Design*.

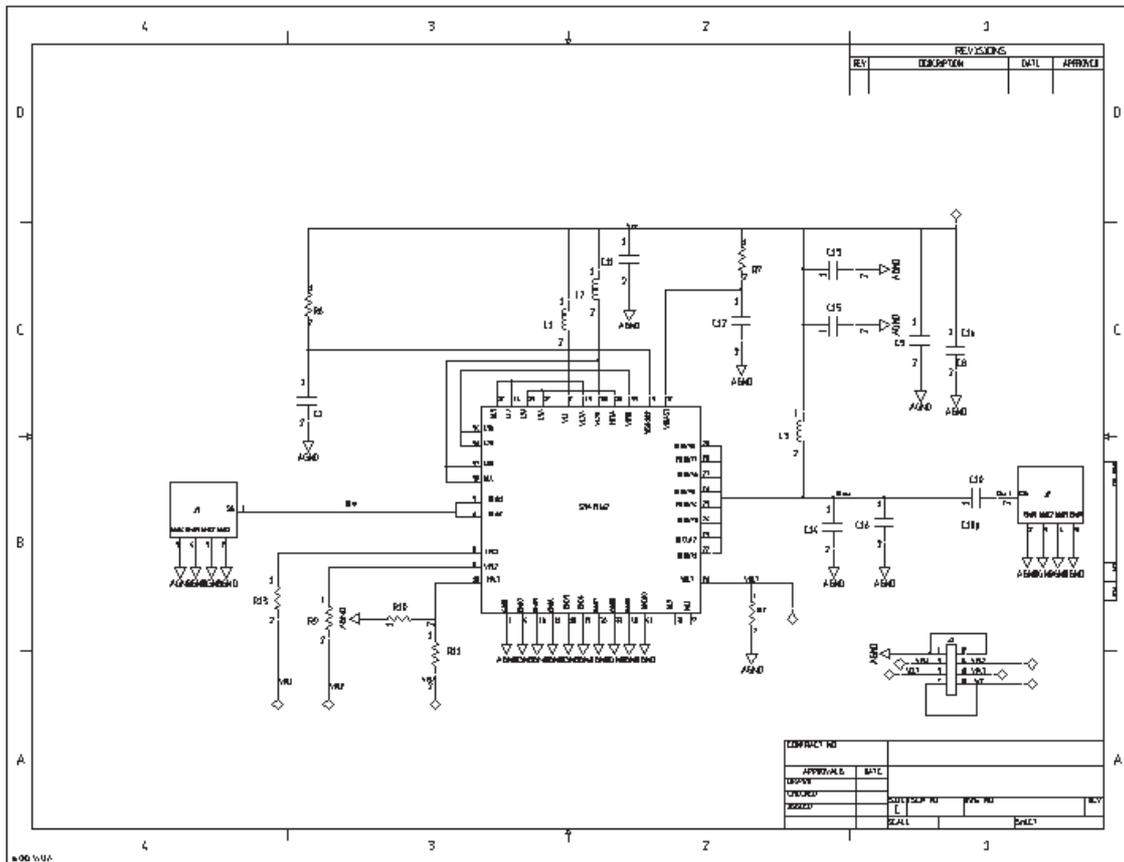


Figura 5.12. Schema elettrico dell'SZM-2166Z

## 5.6 Realizzazione del circuito stampato: *PCB, Printed circuit board*

La fase più delicata dell'intero progetto è stata la realizzazione del PCB mediante il software *Expedition PCB*.

I circuiti stampati vengono realizzati alternando strati di dielettrico isolante, 'core', con strati di materiale elettricamente conduttivo, 'layer e plane'.

Per il nostro stampato il dielettrico utilizzato è stato l'*FR4* costituito da fibre di vetro e rame impregnati in resina epossidica, mentre il materiale conduttivo è stato il *rame*.

L'evaluation board dell'amplificatore di potenza è stata realizzata su sei layer di cui il primo e l'ultimo sono layer di segnale, mentre tutti gli altri sono layer di massa.

La scelta di usare quattro layer come piani di massa è stata dettata dai seguenti motivi:

- fare da schermo alle interferenze elettromagnetiche
- assicurare un buon trasporto di calore in quanto lo spessore del layer è di  $17\ \mu m$  e quindi la conduzione termica è conseguentemente peggiore
- avere strati sottili di dielettrico per ottenere lo spessore delle piste a radiofrequenza.

In particolare sul primo layer risiedono i segnali a radiofrequenza mentre sul sesto layer risiede il segnale di alimentazione.

Le piste a radiofrequenza sono state realizzate in *microstriscia* con il software *Microwave Office* di AWR. Mediante il tool *TxLine*, inserendo i parametri elettrici relativi alla linea riportati sullo schematico, quali la lunghezza elettrica  $L_e$  e il valore dell'impedenza caratteristica  $Z$ , è stato possibile dimensionare le grandezze geometriche, ossia la lunghezza  $l$  e la larghezza  $w$  di metallizzazione della microstriscia.

I progetti delle microstrisce al variare dei due parametri elettrici, citati prima, con le seguenti impostazioni alla  $f_0 = 2.5\ GHz$  sono:

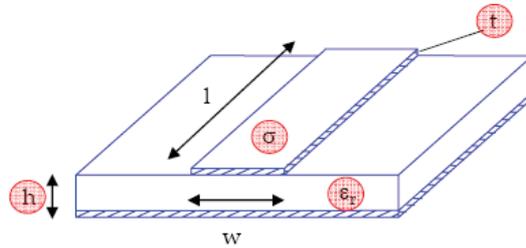


Figura 5.13. Schema di una microstriscia

in cui:

- . utilizzando come substrato il dielettrico l'FR4 la costante dielettrica  $\epsilon_r$  è pari a 4.4
- . utilizzando come conduttore l'oro la conducibilità  $\sigma$  è pari a  $4.1E + 0.7 S/m$
- . utilizzando sei layer lo spessore del dielettrico  $h$  è pari a 0.2 mm e lo spessore del metallo  $t$  è pari a  $17 \mu m$

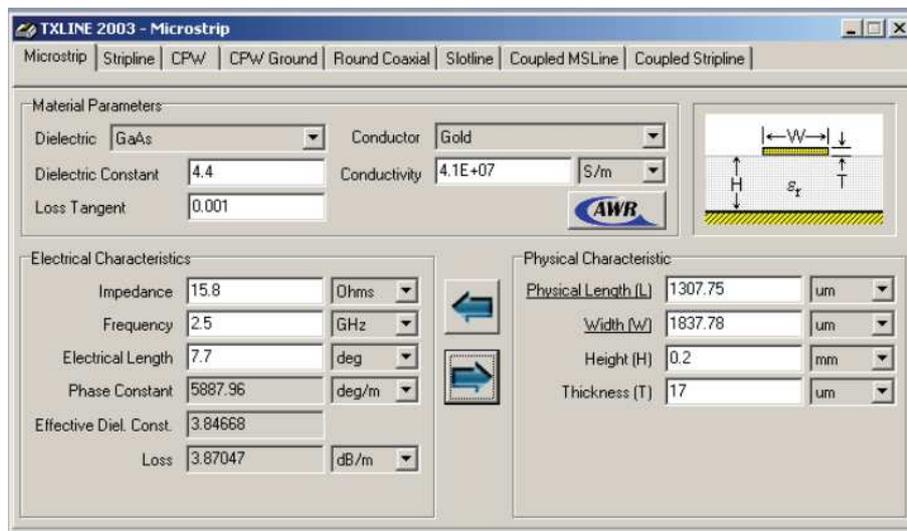


Figura 5.14. Progetto microstriscia,  $L_e = 7.7^\circ$  e  $Z_0 = 15.8 \text{ Ohm}$

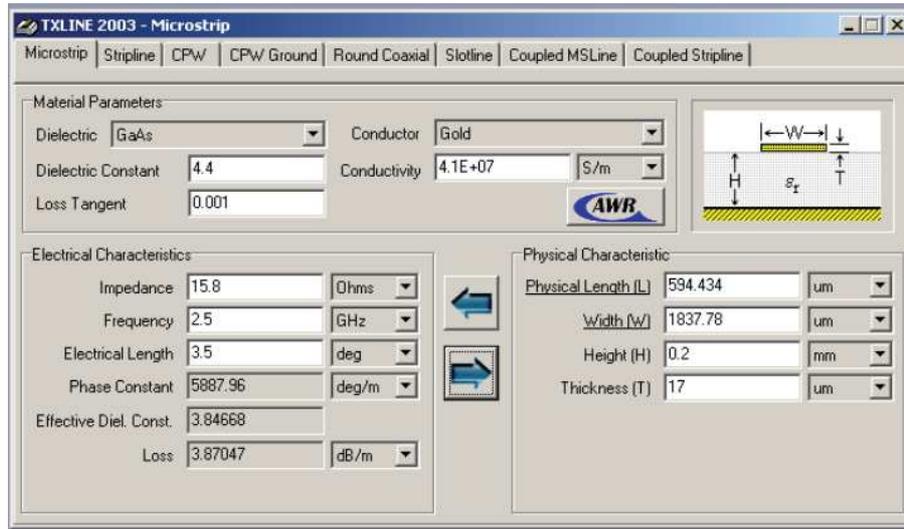


Figura 5.15. Progetto microstriscia,  $L_e = 3.5^\circ$  e  $Z_0 = 15.8$  Ohm

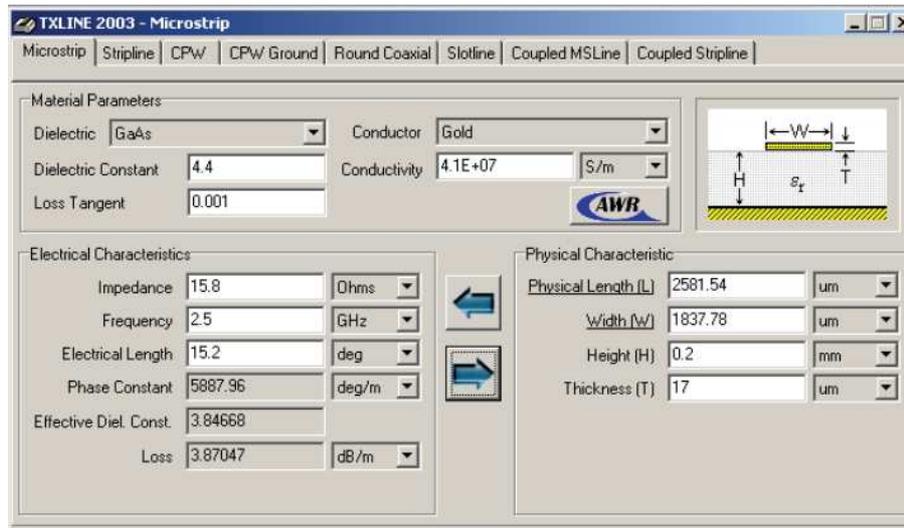


Figura 5.16. Progetto microstriscia,  $L_e = 15.2^\circ$  e  $Z_0 = 15.8$  Ohm

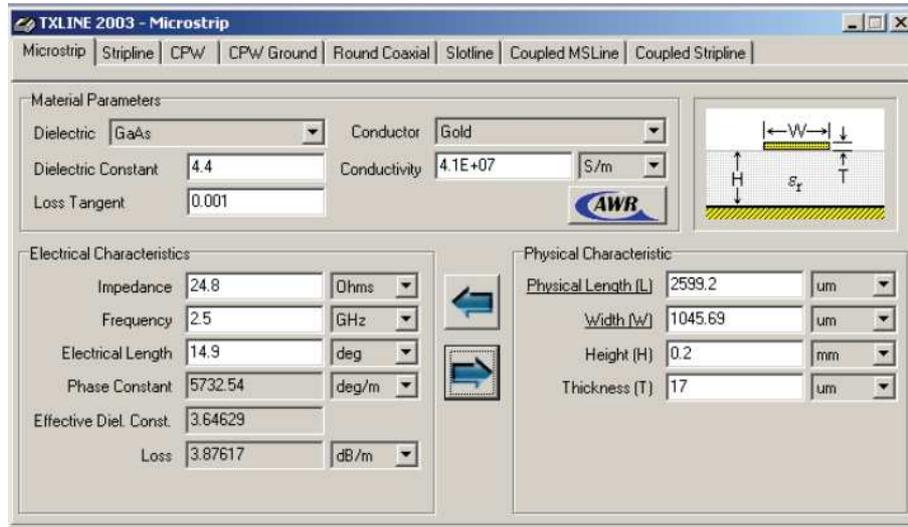


Figura 5.17. Progetto microstriscia,  $L_e = 14.9^\circ$  e  $Z_0 = 24.8$  Ohm

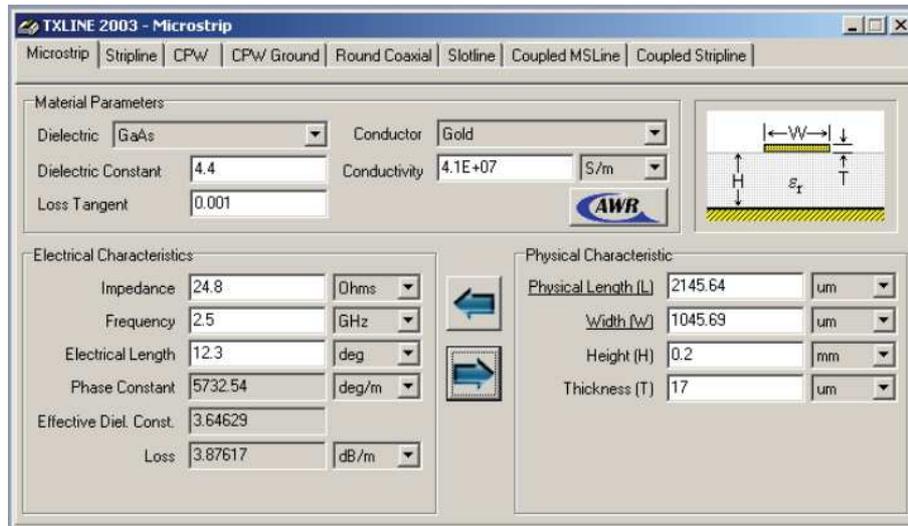


Figura 5.18. Progetto microstriscia,  $L_e = 12.3^\circ$  e  $Z_0 = 24.8$  Ohm

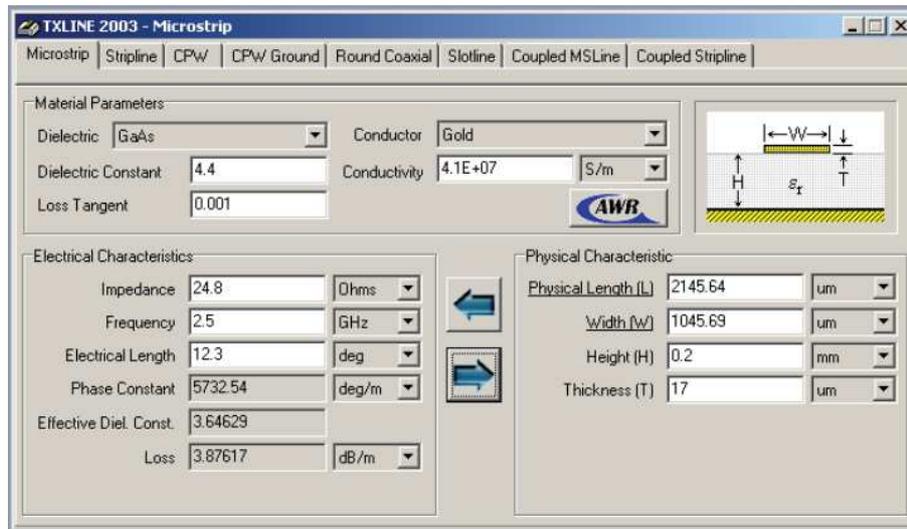


Figura 5.19. Progetto microstriscia,  $L_e = 90^\circ$  e  $Z_0 = 50$  Ohm

Per limitare le perdite non solo gli strati di substrato dielettrico devono essere sottili ma anche la costante dielettrica  $\epsilon_r$  deve avere un piccolo valore.

Date le loro dimensioni, le microstrisce sono state inserite nel PCB come *Plane Shape*, che permette di inserire un piano conduttivo della dimensione desiderata sul layout.

Anche le piste dell'alimentazione sono state inserite come plane shape in quanto si è cercato di realizzarle il più larghe possibili per avere un comportamento migliore, ossia agevolare la dissipazione per irraggiamento.

Tutte le altre piste sono di dimensione  $0.2\text{ mm}$  considerando che la dimensione minima limite di realizzazione fisica è  $0.15\text{ mm}$ . Quest'ultima è anche la distanza minima tra due piste, due pad, due via, tra una pista e un pad, una pista e un via e tra un pad e un via. Nel nostro layout la distanza tra due pad è stata forzata a  $0.005\text{ mm}$  per riuscire a posizionare due componenti sulla stessa microstriscia. Questi parametri vengono inseriti nel *Constraint*.

Il processo di realizzazione del PCB consta dunque dei seguenti passaggi:

- Associare i dati del progetto relativi allo schematico dal software Design Capture al software Expedition PCB mediante la *Forward Annotation*
- Definire la dimensione e la forma del PCB inserendo il *Board outline* e l'area dove posizionare le piste e i componenti inserendo il *Route outline*.

L'evaluation board dell'amplificatore ha dimensione  $64 \times 47\text{ mm}$ .

- Posizionare le celle dei componenti col *Place part*

- Posizionare le piste di collegamento tra i componenti col *emphPlane shape* o col *Route mode*
- Generare i piani conduttivi col *Plane processor*
- Posizionare i via per assicurare un potenziale costante tra i piani e una conduzione termica elevata. Nel nostro caso i via sono stati posizionati oltre che in maggior numero possibile anche molto vicini tra di loro in quanto essendo a radiofrequenza la lunghezza d'onda  $\lambda$  è minore
- Fare la verifica del progetto mediante il *Batch DRC (Design Rule Check)* che effettua controlli sulle connessioni e prossimità impostate nei constraint

Al termine del PCB viene generato, mediante *il software Gerbertool*, il file Gerber come risultato definitivo del progetto, che sarà inviato alla ditta produttrice per realizzare fisicamente il circuito stampato.

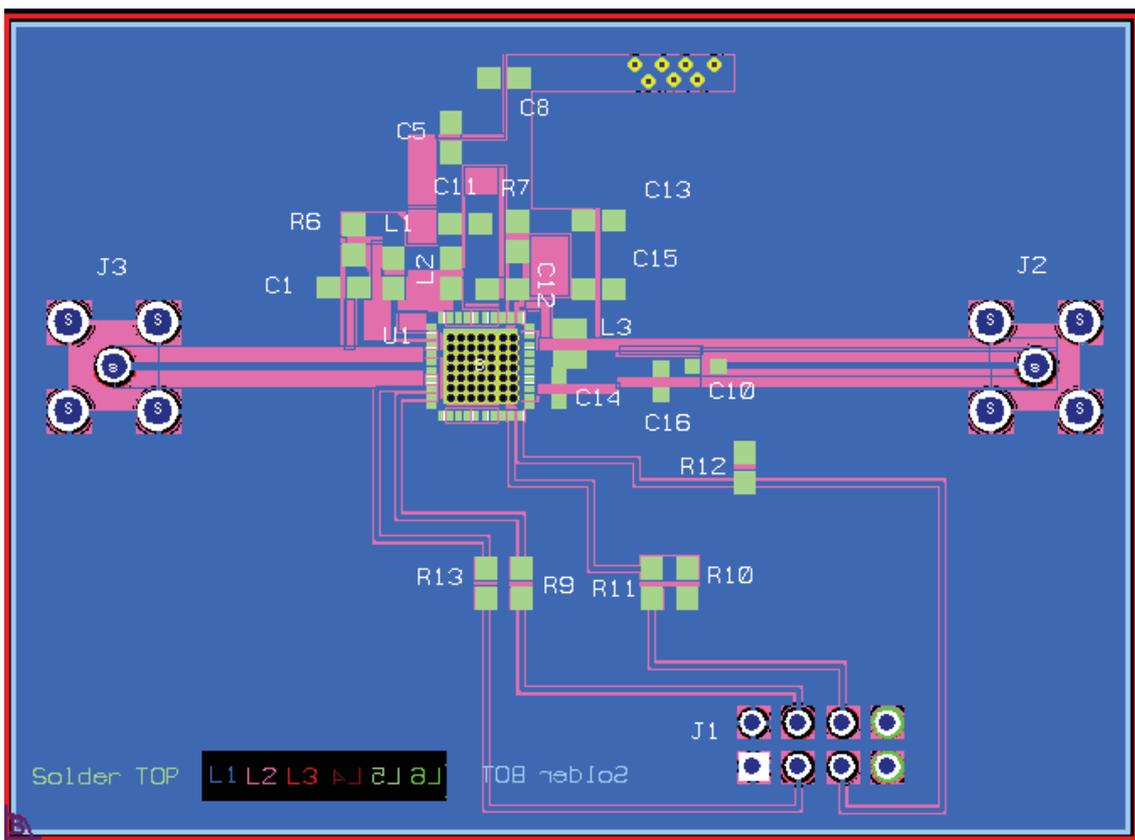


Figura 5.20. PCB dell'SZM-2166Z

## 5.7 Test di funzionamento

Realizzata fisicamente l'evaluation board dell'amplificatore di potenza qui di seguito mostrata

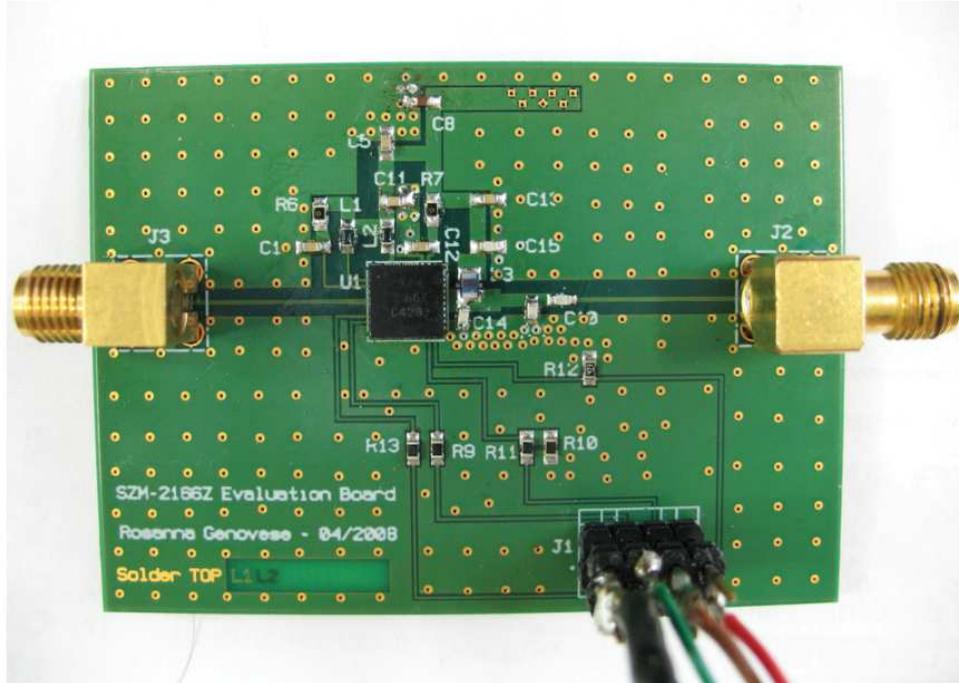


Figura 5.21. Evaluation board dell'SZM-2166Z

sono stati effettuati i test iniziali di funzionamento a partire dall'alimentazione del circuito.

In seguito mediante l'analizzatore di spettro è stata verificata la potenza effettivamente erogata collegando come generatore di potenza in ingresso il transceiver CC2510 e variando le configurazioni dei registri di quest'ultimo.

Questi test preliminari sono stati eseguiti con la strumentazione del *laboratorio hardware* del Dipartimento di Elettronica e hanno permesso di testare il corretto funzionamento dell'amplificatore, quindi che erogasse 36 dBm di potenza d'uscita.

Successivamente sono stati eseguiti test più approfonditi alla *Spin Electronics*, un'azienda italiana fondata nel 1994 e specializzata nel settore delle telecomunicazioni.

I valori riportati sono stati oltre alla potenza d'uscita  $P_{out}$  anche i rispettivi valori di corrente:

$P_{in}$	$P_{out}$	$I (A)$
-10 dBm	24.1 dBm	0.606 A
-9 dBm	25.1 dBm	0.63 A
-8 dBm	26.05 dBm	0.66 A
-7 dBm	27 dBm	0.7 A
-6 dBm	27.9 dbm	0.75 A
-5.3 dBm	28.5 dBm	0.79 A
-4.21 dBm	29.38 dBm	0.855 A
-3.2 dBm	30.25 dbm	0.931 A
-2.1 dBm	31.14 dbm	1.01 A
-1.1 dBm	32 dBm	1.11 A
0 dBm	32.9 dBm	1.2 A
1 dBm	33.74 dbm	1.33 A
2.1 dBm	34.6 dBm	1.46 A
3 dBm	35.3 dBm	1.6 A
4 dBm	36 dBm	1.75 A
5 dBm	36.55 dBm	1.9 A

dai quali è possibile stimare le due grandezze che descrivono il comportamento dell'amplificatore: *il guadagno e l'efficienza*:

$$G \text{ dB} = P_{out} \text{ dBm} - P_{in} \text{ dBm}$$

da cui si ricava:

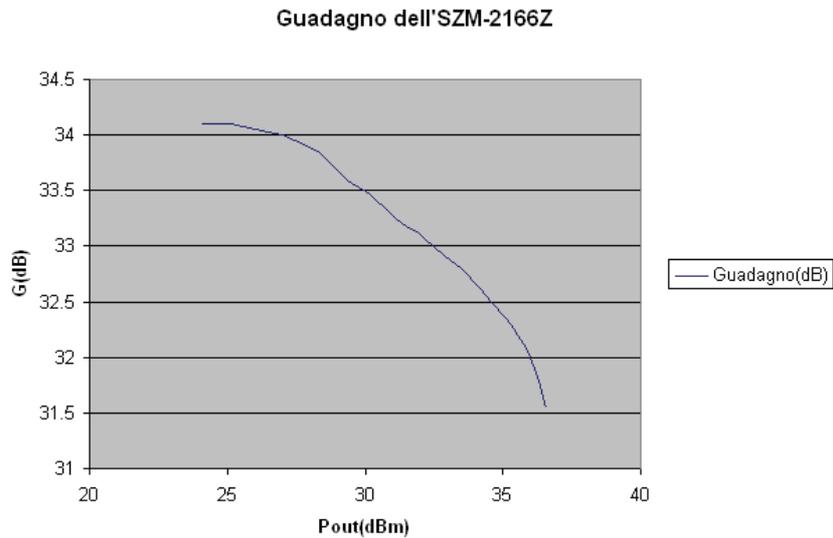


Figura 5.22. Guadagno dell'SZM-2166Z

e

$$\eta = \frac{P_{out}}{P_{al}}$$

in cui:

$$P_{al} = VI$$

da cui si ricava:

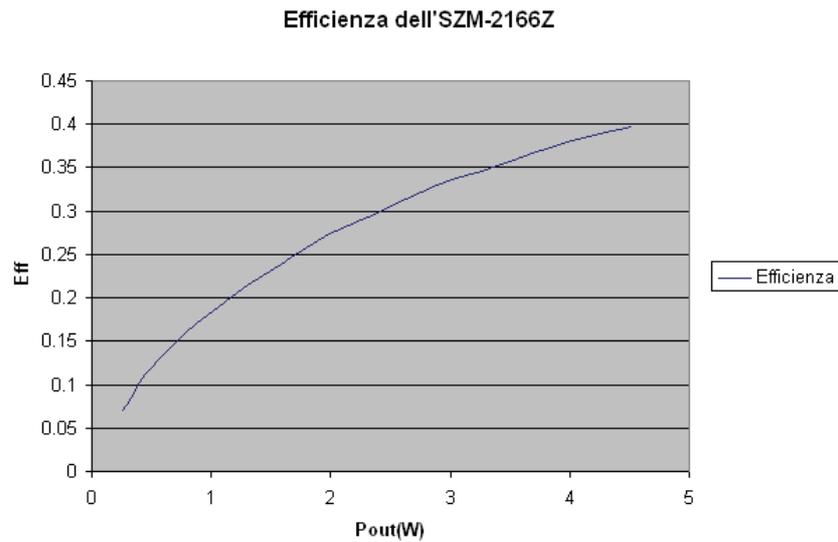


Figura 5.23. Efficienza dell'SZM-2166Z

Gli andamenti dei due grafici forniscono un buon riscontro confrontandoli con quelli riportati sul datasheet.

# Capitolo 6

## Progetto del sistema di comunicazione a 2.4 GHz: la scheda TxRx

### 6.1 Dimensionamento della potenza in ricezione

Dopo aver analizzato il sistema in trasmissione si è passati a valutarne il comportamento in ricezione.

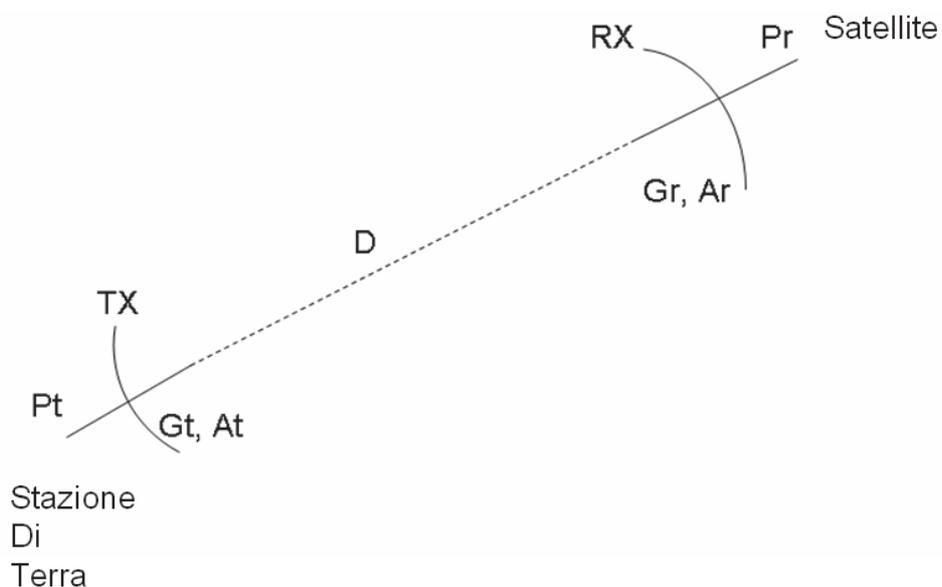


Figura 6.1. Trasmissione nello spazio libero nel collegamento uplink

Dall'equazione di trasmissione di Friis, utilizzando i parametri:

- guadagno d'antenna trasmittente  $G_{AT} = 3 \text{ dBiC}$
- guadagno d'antenna ricevente  $G_{AR} = 35 \text{ dBiC}$
- un'attenuazione  $\alpha = 169 \text{ dB}$

con una potenza in trasmissione  $P_{Tx}$  pari a circa  $40 \text{ dBm}$  (valore maggiore della potenza in trasmissione da satellite, in quanto quest'ultima è limitata nella sua disponibilità perchè fornita principalmente dai pannelli solari), si ha che il valore di potenza in ricezione risulta:

$$P_{Rx} = (40) \text{ dBm} + (35) \text{ dBiC} + (3) \text{ dBiC} - (169) \text{ dB} = -91 \text{ dBm}$$

Dalle caratteristiche riportate sul datasheet del transceiver CC2510 il valore trovato denota che non è possibile trasmettere oltre un certo valore di data rate, ma che la sensitivity migliore si ha per un valore pari a  $-81 \text{ dBm}$  consentendo un data rate pari a  $500 \text{ kbps}$ .

Quindi si è reso necessario l'impiego di un *low noise amplifier*, LNA, in grado di attenuare la potenza di rumore rispetto al segnale utile.

Rispettando le specifiche di progetto la ricerca è stata incentrata su un LNA con alto guadagno e bassa cifra di rumore: *il MAX2644* della *MAXIM* utilizzato già precedentemente nel progetto del sistema di comunicazione del satellite PiCpoT.

Si tratta di un dispositivo progettato per applicazioni 802.11 WLAN e per i sistemi radio bluetooth alla frequenza di  $2.4 \text{ GHz}$  con alto guadagno,  $16 \text{ dBm}$  e bassa cifra di rumore,  $2 \text{ dBm}$ .

Il range di alimentazione va da  $(2.7 \div 5.5) \text{ V}$  e il package è di tipo SC70 di dimensioni ridottissime.

Lo schema elettrico è riportato in Fig. 6.18.

## 6.2 Progetto del sistema di comunicazione

Valutata l'esigenza di aggiungere un amplificatore di potenza, per fornire la potenza necessaria in trasmissione, e un amplificatore a basso rumore, per migliorare la sensitivity, si è progettato il sistema di comunicazione.

Il principio di base è stato quello di utilizzare due transceiver, uno per la trasmissione e uno per la ricezione.

La scheda deve essere connessa al computer di bordo tramite un'unica interfaccia, ma essendoci due transceiver, si è scelto di averne uno *master* ed uno *slave* comunicanti tramite una seriale asincrona tra di loro.

I transceiver sono collegati rispettivamente al *power amplifier*, l'SZM-2166Z e al *low noise amplifier*, il MAX2644.

Questi ultimi sono a loro volta collegati tramite due *switch*, gli FMS2031-001, identici ma pilotati in opposizione di fase tra di loro (quando uno è chiuso, l'altro è aperto):

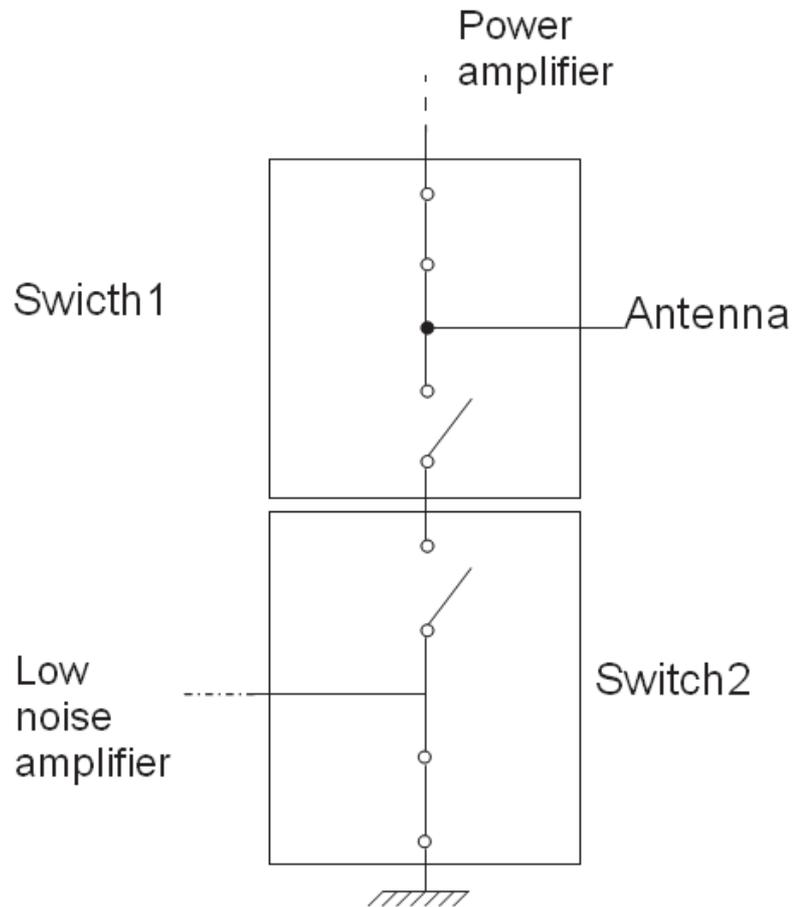


Figura 6.2. Schema degli switch in Trasmissione

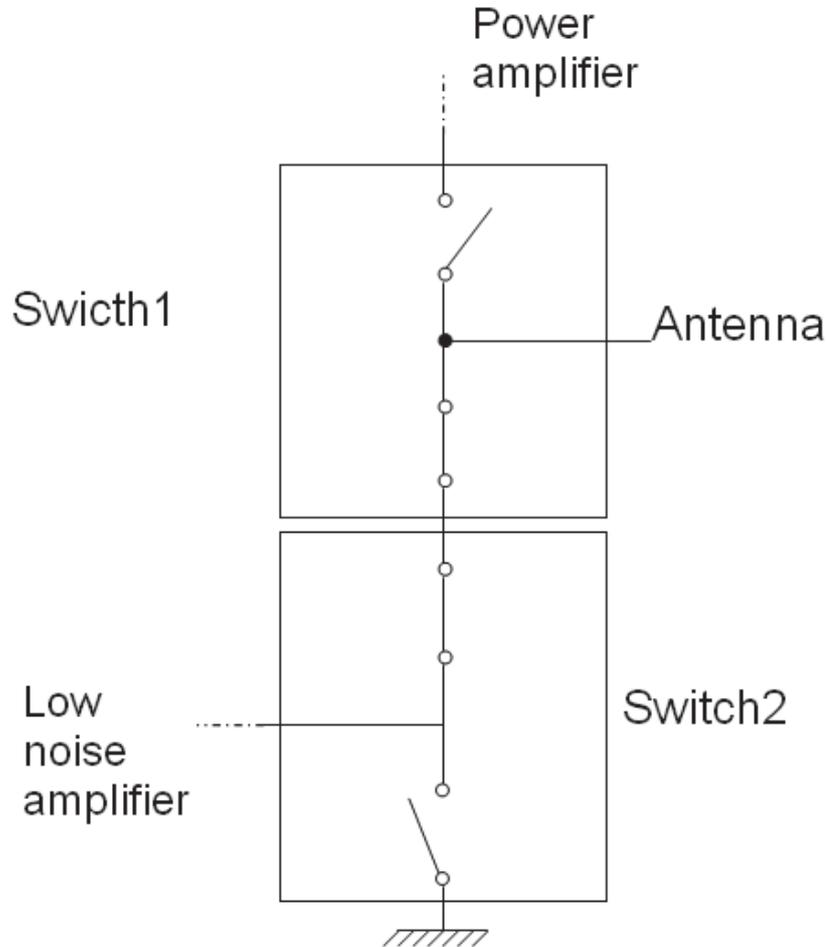


Figura 6.3. Schema degli switch in Ricezione

La scelta di utilizzare due transceiver è stata dettata dal fatto che il CC2510 è dotato di una sola uscita *rf* per la trasmissione e la ricezione, quindi ciò avrebbe comportato l'impiego di uno switch supplementare.

Poichè nella realtà uno switch non ha un isolamento infinito ma finito ciò avrebbe potuto creare un anello di reazione di conseguenza delle possibili oscillazioni rendendo il sistema instabile.

In aggiunta il CC2510 è dotato anche di un *ADC* ed è quindi possibile visualizzare la potenza effettiva in uscita dall'SZM-2166Z collegando la *Vdet*, l'uscita del rivelatore di potenza ad uno dei pin di ingresso dell'ADC.

L'alimentazione che verrà fornita all'intero sistema sarà pari a circa 16 V.

Poichè i dispositivi presenti nel sistema richiedono alimentazioni diverse sono stati impiegati: un regolatore lineare di tensione *LM317D* per il transceiver e

l'LNA che richiedono una tensione d'uscita pari a 3.3 V e un regolatore switching l'LM2596 per alimentare l'SZM-2166Z a 6 V.

Per quanto riguarda gli switch, le due tensioni, che ne regolano lo stato (*open/close*) alternandosi tra il livello alto di tensione, “*High, H*” e il livello basso di tensione, “*Low, L*”, sono fornite dal transceiver.

Lo stesso vale per il pin di accensione e spegnimento dello switching che risulta in fase “*ON*” quando la tensione risulta inferiore a 0.6 V, che è la tensione massima di spegnimento, e in fase “*OFF*” quando la tensione è superiore a 2 V, che è la tensione minima di accensione.

Lo schema elettrico è riportato in Fig. 6.20.

### 6.3 L'alimentatore switching LM2596

Rispettando i limiti delle specifiche di progetto, il regolatore switching scelto per fornire in uscita la tensione  $V_{out}$  atta ad alimentare l'amplificatore di potenza SZM-2166Z, ricevendo in ingresso un dato valore di tensione  $V_{in}$ , è stato: LM2596 della *National Semiconductor*.

Si tratta di un regolatore integrato monolitico di tipo step down con  $V_{out} < V_{in}$ .

L'LM2596 è caratterizzato da un'alta efficienza energetica  $\eta = \frac{P_{out}}{P_{in}}$  con un'alimentazione  $V_{in}$  che può andare ben oltre i 40 V e in grado di erogare una corrente di uscita pari a 3 A.

Il dispositivo opera ad un'alta frequenza di switching stabilita a 150 kHz consentendo così l'impiego di componenti di dimensioni molto ridotte.

La serie di regolatori switching LM2596 è costituita da tre versioni a tensione d'uscita fissa (3.3 V, 5 V, 12 V) e da una versione a tensione d'uscita variabile (1.2 ÷ 37) V, che è quella utilizzata nel nostro progetto in quanto la nostra  $V_{out}$  deve essere pari a 6 V.

L'integrato è dotato anche di uno shutdown termico, di una protezione in caso di sovracorrente e di un pin ON/OFF che può porre il sistema in standby.

In questo modo il regolatore assorbe una corrente non superiore a 80  $\mu A$ .

L'LM2596 è realizzato con un package di tipo *TO263*.

### 6.4 Dimensionamento dei componenti dell'LM2596

Lo schematico di riferimento utilizzato per il progetto è stato:

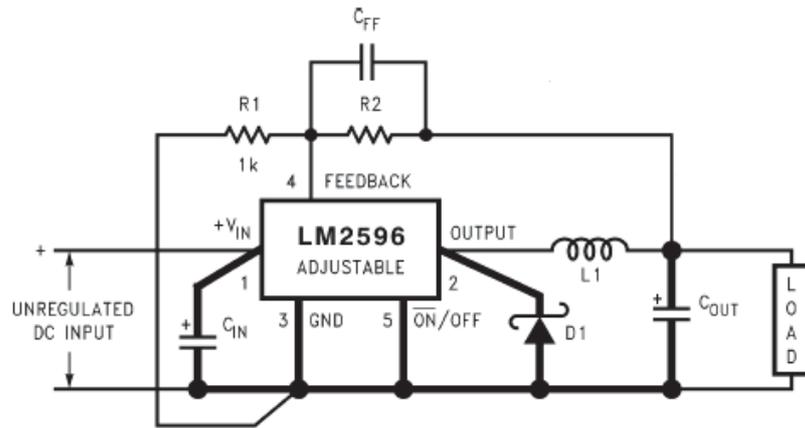


Figura 6.4. Schema dell'LM2596 a tensione d'uscita variabile

in cui i componenti sono stati dimensionati come segue:

poichè la  $V_{in}$  sarà pari a circa di 16 V, dai valori riportati sul datasheet del regolatore, viene posto come valore di tensione nominale  $V_{in} = 25$  V.

Volendo in uscita una tensione variabile si utilizza la seguente formula

$$V_{out} = V_{ref} \left( 1 + \frac{R_2}{R_1} \right)$$

in cui  $V_{ref}$ , la tensione di reazione, è pari a 1.23 V e le due resistenze  $R_1$  e  $R_2$  realizzano l'anello di reazione di controllo del sistema.

Ponendo  $R_1$  pari ad un valore compreso fra 240  $\Omega$  e 1.5 k $\Omega$ , che scegliamo essere pari 1 k $\Omega$  per minimizzare il rumore sul pin di retroazione, e volendo una tensione  $V_{out} = 6$  V, si ricava che:

$$R_2 = R_1 \left( \frac{V_{out}}{V_{ref}} - 1 \right) \cong 3.9 \text{ k}\Omega$$

Per valutare il valore dell'induttanza  $L$  si deduce prima il valore della tensione in ingresso indicata come prodotto  $E(\text{V}) \times T(\mu\text{s})$ :

$$ET = (V_{in} - V_{out} - V_{sat}) \frac{V_{out} + V_D}{V_{in} - V_{sat} + V_D} \frac{1000}{150 \text{ kHz}} (\text{V}/\mu\text{s})$$

in cui:

- $V_{sat}$  è la tensione di saturazione interna dello switch pari a 1.16 V
- $V_D$  è la tensione ai capi del diodo pari a 0.5 V

da cui:

$$ET = 31.8 (V/\mu s)$$

Dopo di che utilizzando il grafico

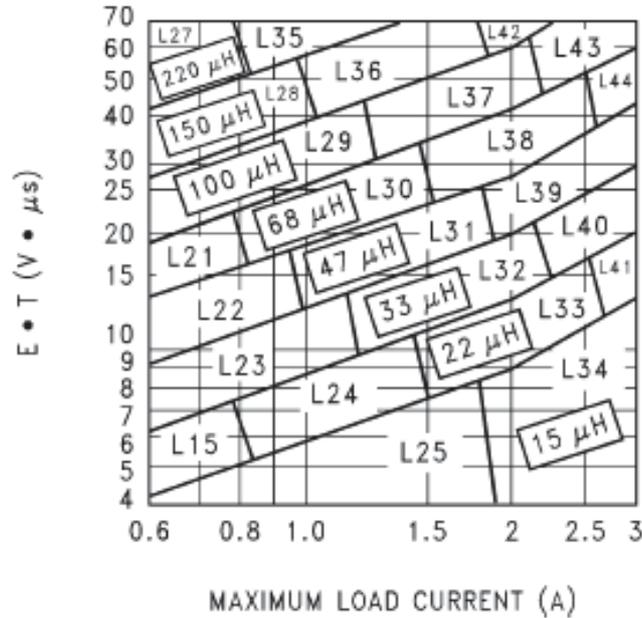


Figura 6.5. Valori di induttanza

dall'intersezione del valore precedentemente trovato con la corrente di carico pari a 3 A ne segue il valore dell'induttanza  $L = 47 \mu H$ .

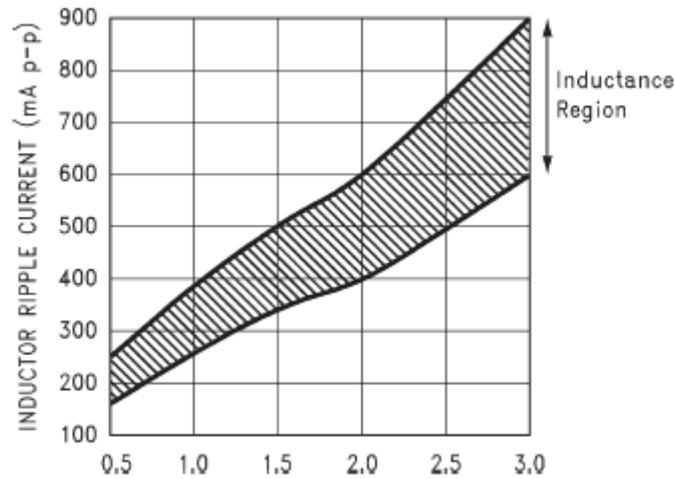
Il condensatore d'uscita  $C_{out}$  viene dimensionato in modo da avere un  $ESR$ , *equivalent series resistor* molto basso.

I condensatori che hanno basso  $ESR$  sono di sicuro i condensatori ceramici.

Il valore della  $C_{out}$  e quello dell' $ESR$  sono responsabili, come si evince dalla formula con  $ESR \cong 0$ , del ripple di tensione in uscita

$$\Delta V_{out} \leq \Delta I_L \left( ESR + \frac{1}{8fC_{out}} \right) \cong \frac{\Delta I_L}{8fC_{out}}$$

Per cui valutando dal relativo grafico una corrente di ripple di induttore  $\Delta I_L = 740 \text{ mA}$  e scegliendo di impiegare due condensatori in parallelo pari a  $47 \mu F$  si ha che:

Figura 6.6. Valori di  $\Delta I_L$ 

$$\Delta V_{out} \leq 6 \text{ mV}$$

che è un valore ottimale.

La capacità  $C_{FF}$  viene utilizzata per garantire una stabilità supplementare per alti valori di tensione in uscita e per aiutare ad abbassare l'ESR della  $C_{out}$ .

Essa è dimensionata dalla formula:

$$C_{FF} = \frac{1}{31 \cdot 10^3 \cdot R_2} \cong 8.2 \text{ nF}$$

Il diodo schottky utilizzato è stato selezionato con i seguenti vincoli:

- la corrente  $I_D$  deve essere 1.3 volte maggiore della massima corrente di carico, quindi pari a 3.9 A
- la tensione inversa  $V_R$  deve essere 1.25 volte maggiore della tensione massima di ingresso, quindi pari a 31.25 V.

Da questi dati, dai valori riportati sulla relativa tabella del datasheet, è stato scelto il diodo *50WQ0A*.

VR	3A Diodes				4A–6A Diodes			
	Surface Mount		Through Hole		Surface Mount		Through Hole	
	Schottky	Ultra Fast Recovery	Schottky	Ultra Fast Recovery	Schottky	Ultra Fast Recovery	Schottky	Ultra Fast Recovery
20V		All of these diodes are rated to at least 50V.	1N5820	All of these diodes are rated to at least 50V.		All of these diodes are rated to at least 50V.	SR502	All of these diodes are rated to at least 50V.
	SK32		SR302				1N5823	
			MBR320				SB520	
30V	30WQ03		1N5821					
	SK33		MBR330		50WQ03		SR503	
			31DQ03				1N5824	
40V		MURS320	1N5822	MUR320		MURS620	SB530	MUR620
	SK34		SR304		50WQ04		SR504	
	MBRS340		MBR340				1N5825	
50V or More	30WQ04	30WF10	31DQ04			50WF10	SB540	HER601
	SK35		SR305					
	MBRS360		MBR350		50WQ05		SB550	
	30WQ05		31DQ05				50SQ080	

Figura 6.7. Modelli di diodi

Infine la capacità di ingresso  $C_{in}$  è stata dimensionata con un alto valore per immagazzinare una carica tale da far sì che il regolatore non ecceda in corrente.

Anche nella scelta di questa capacità si deve avere un basso ESR per prevenire la sovratensione.

In più la corrente  $RMS$  della  $C_{in}$  deve essere pari a ad  $1/2$  della  $I_{MAX}$  di carico, quindi pari a 1.5 A.

Da ciò la scelta di utilizzare due capacità in parallelo da  $22 \mu F$ .

## 6.5 Inserimento dello switching nella Libreria Centrale di Mentor Graphics

Dopo aver scelto i dovuti componenti lo switching è stato inserito nella libreria centrale, già precedentemente creata per il progetto dell'evaluation board dell'amplificatore di potenza, con il simbolo logico e la rispettiva cella fisica.

La partizione in cui si colloca è stata denominata *switching*.

Il simbolo creato è stato il seguente:

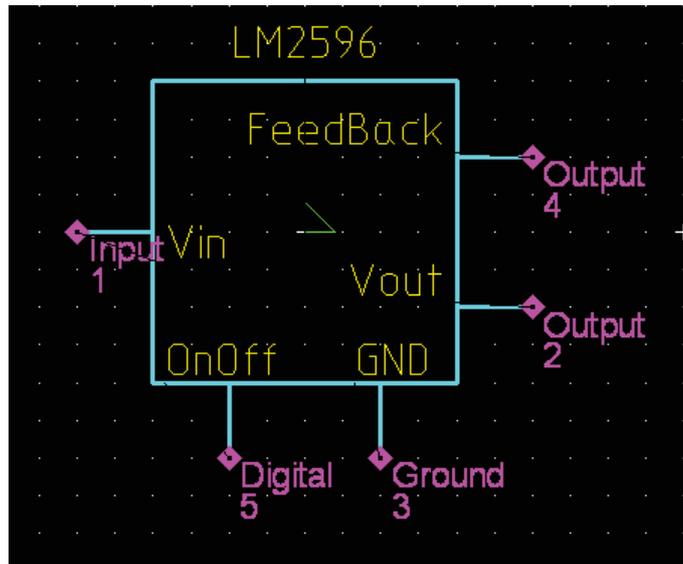


Figura 6.8. Simbolo dell'LM2596

con i cinque pin:  $V_{in}$ ,  $V_{out}$ ,  $GND$ ,  $OnOff$  e  $Feedback$ .

Per quanto riguarda invece la realizzazione della cella fisica, prima sono stati creati i padstack che la costituiscono: il padstack associato al pad centrale di dimensione  $10.41 \times 10.80 \text{ mm}$  e il padstack relativo ai pad esterni di dimensione  $2.16 \times 1.07 \text{ mm}$ .

La relativa cella etichettata col nome  $TO263$  e di tipo *surface mount* è stata quindi:

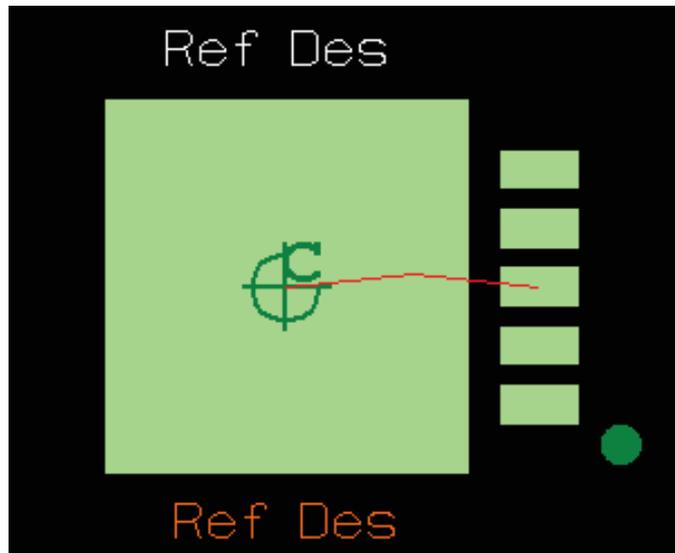


Figura 6.9. Cella dell'LM2596

in cui si nota il collegamento tra il pad centrale e il pad esterno(GND) entrambi a massa.

Conseguentemente è stata fatta l'associazione dei pin tra il simbolo e la cella.

## 6.6 Lo switch FMS2031-001

Lo switch di collegamento tra il power amplifier e il low noise amplifier scelto per le condizioni di progetto è stato: l'*FMS2031-001* della *Filtronic*.

Si tratta di uno switch che consente un alto isolamento tra la parte trasmittente e la parte ricevente in un sistema di comunicazione.

In più le basse perdite di inserzione pari a 0.47 dB e i suoi 10 W di potenza garantiscono un'alta linearità nella trasmissione minimizzando la potenza dissipata.

Il funzionamento dell'*FMS2031-001* è regolato dalle tensioni  $V_1$  e  $V_2$  che si alternano tra il livello alto di tensione  $H$  compreso tra (2.7 ÷ 6) V e il livello basso di tensione  $L$  compreso tra (0 ÷ 2) V.

In questo modo dato lo schema:

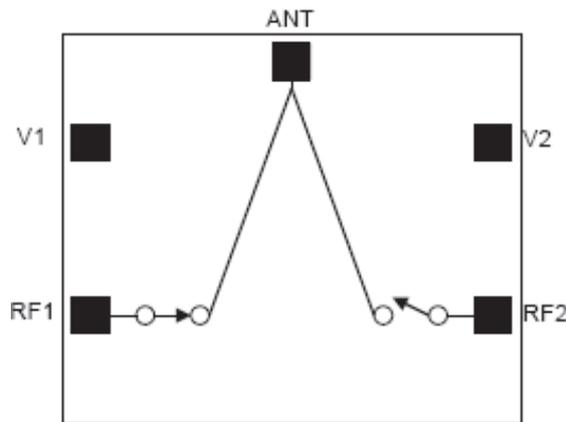


Figura 6.10. Schematico dell'*FMS2031-001*

vale:

Switch State	VC1	VC2	ANT- RF1	ANT- RF2
(A)	HIGH	LOW	Insertion Loss	Isolation
(B)	LOW	HIGH	Isolation	Insertion Loss

Figura 6.11. Funzionamento dell'*FMS2031-001*

Il dispositivo è disponibile con un package a basso costo di tipo *QFN* di dimensione  $3 \times 3 \text{ mm}$

## 6.7 Inserimento dello switch nella libreria Centrale di Mentor Graphics

Anche lo switch è stato conseguentemente inserito in libreria.

La partizione in cui si colloca è stata denominata *switch*.

Il simbolo creato è stato il seguente:

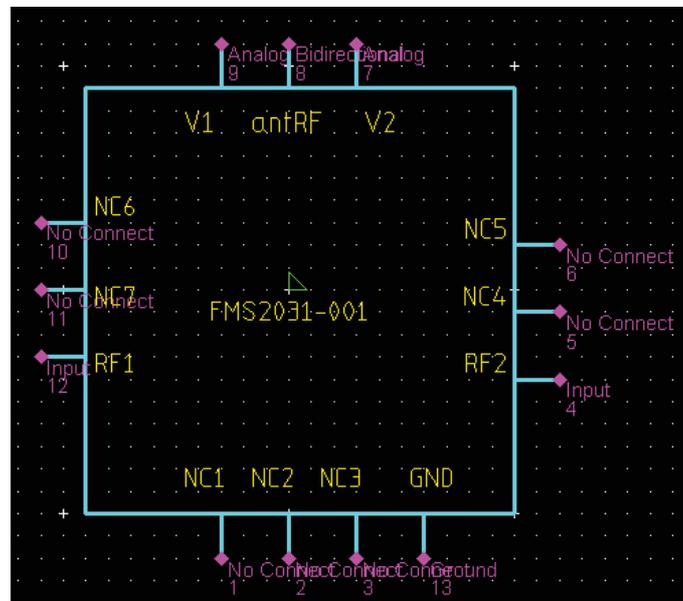


Figura 6.12. Simbolo dell'FMS2031-001

con i tredici pin:  $V_1$ ,  $V_2$ , antRF,  $RF_1$ ,  $RF_2$ , GND e i restanti No connect.

Per quanto riguarda la realizzazione della cella i padstack relativi sono stati quello associato al pad centrale, che viene creato come custom pad di dimensione  $1.2 \text{ mm} \times 1.2 \text{ mm}$ , seguendo lo stesso procedimento usato per l'amplificatore, e quello relativo ai pad esterni di dimensione  $0.55 \text{ mm} \times 0.23 \text{ mm}$ .

Ne segue la cella etichettata col nome *12QFN* e di tipo *through hole*:

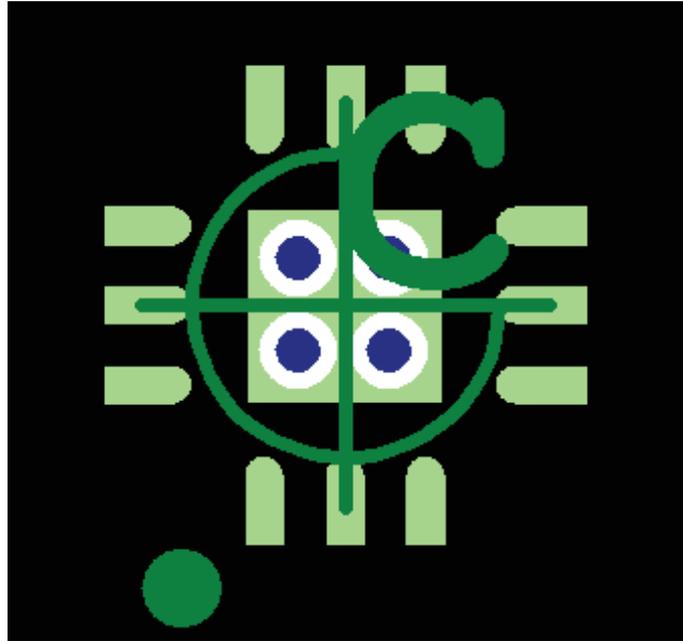


Figura 6.13. Cella dell'FMS2031-001

con conseguente associazione dei pin tra il simbolo e la cella.

## 6.8 Analisi e revisione dei restanti componenti della scheda

I simboli logici e le celle fisiche dei restanti componenti della *schedaTxRx* già presenti in libreria sono stati revisionati e adeguati concordemente ai vincoli del progetto.

In particolare le modifiche hanno riguardato gli schematici.

L'evaluation board del transceiver CC2510 ha subito, oltre alle modifiche note nello schema di figura Fig. 3.7 avvenute durante l'implementazione del software di trasmissione e ricezione, anche il cambiamento del connettore di programmazione in quanto la *Texas Instrument* ha poi sviluppato la board di programmazione che a inizio progetto non era ancora disponibile.

I valori delle resistenze del regolatore LM317D sono stati variati in quanto incompatibili con i limiti di progetto (Fig. 6.15).

Per dimensionare le resistenze  $R_1$  e  $R_2$  vale la seguente formula:

$$V_{out} = V_{ref} \left( 1 + \frac{R_2}{R_1} \right) + I_{ADJ} R_2$$

con  $V_{ref}$  pari a 1.25 V e  $I_{ADJ}$  nulla in quanto la scelta dei valori delle due resistenze deve essere fatta sulla base di questa condizione.

Sostituendo si ricava:

$$\frac{R_2}{R_1} = 1.64$$

da cui i valori scelti:

$$R_2 = 27 \text{ k}\Omega \quad R_1 = 18 \text{ k}\Omega$$

In seguito alla decisione di impiegare l'ADC per visualizzare la potenza in uscita dall'amplificatore di potenza, il valore della resistenza di carico è stato variato da 47 k $\Omega$  a 220 k $\Omega$  considerando che il range della resistenza di carico deve essere compreso tra (10  $\div$  100) k $\Omega$  e che l'ADC presenta un'impedenza pari a 197 k $\Omega$  (Fig. 6.16).

Infine sul pin OnOff del regolatore switching è stata aggiunta una resistenza da 33 k $\Omega$  per limitare la corrente (Fig. 6.17).

## 6.9 Realizzazione dello schema elettrico del sistema di comunicazione e relativo PCB

Aggiornata la libreria con tutti i componenti inerenti al progetto della scheda, il passo successivo è stato la realizzazione dello schematico.

Data la complessità della scheda TxRx il progetto è stato suddiviso in sei schematici tanti quanti sono i dispositivi che la compongono, definendo così una struttura gerarchica.

I collegamenti di *input* e *output* di ogni integrato sono definiti come *hierarchical* e questi ultimi sono poi definiti come pin dei blocchi contenenti i rispettivi schemi elettrici ad essi associati.

I blocchi vengono creati col *Place block* e costituiranno lo schema al livello superiore in cui i collegamenti e l'inserimento dei componenti richiede la stessa metodologia descritta per l'SZM-2166Z.

Lungo i collegamenti relativi agli switch sono state posizionate delle capacità da 9 pF, in quanto esse sono richieste su tutte le uscite *rf* dei due dispositivi.

Il layout della scheda è stato realizzato su otto layer in cui il primo, il sesto, il settimo e l'ottavo sono layer di segnale mentre tutti gli altri sono layer di massa.

Le piste a radiofrequenza sono posizionate sul primo layer e sono tutte realizzate in microstriscia con impedenza caratteristica  $Z = 50 \Omega$  e lunghezza elettrica  $L_e = 90$ .

Fanno eccezione le piste relative all'amplificatore di potenza e una pista dell'amplificatore a basso rumore che è lunga 400 *mils*.

La parte a radiofrequenza è stata adeguatamente separata da quella di potenza per evitare che i componenti interferiscano tra di loro.

La dimensione della schedaTxRx risulta di 82.5 mmx81.5 mm.

Per il resto anche per il PCB valgono le stesse proprietà oltre la stessa metodologia usata per l'amplificatore di potenza. Di seguito sono riportati gli schematici e il PCB della schedaTxRx:



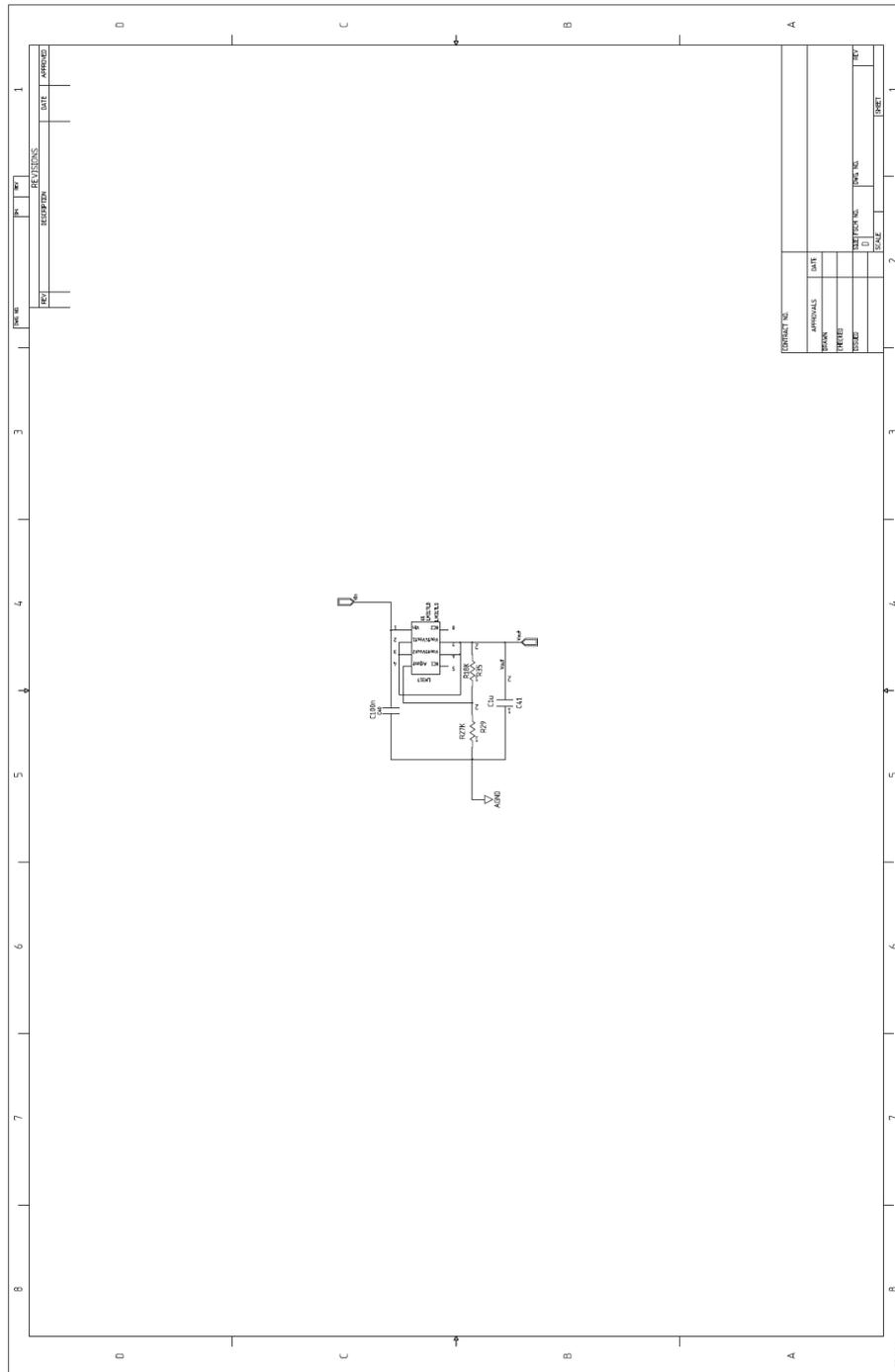


Figura 6.15. Schema elettrico LM317D



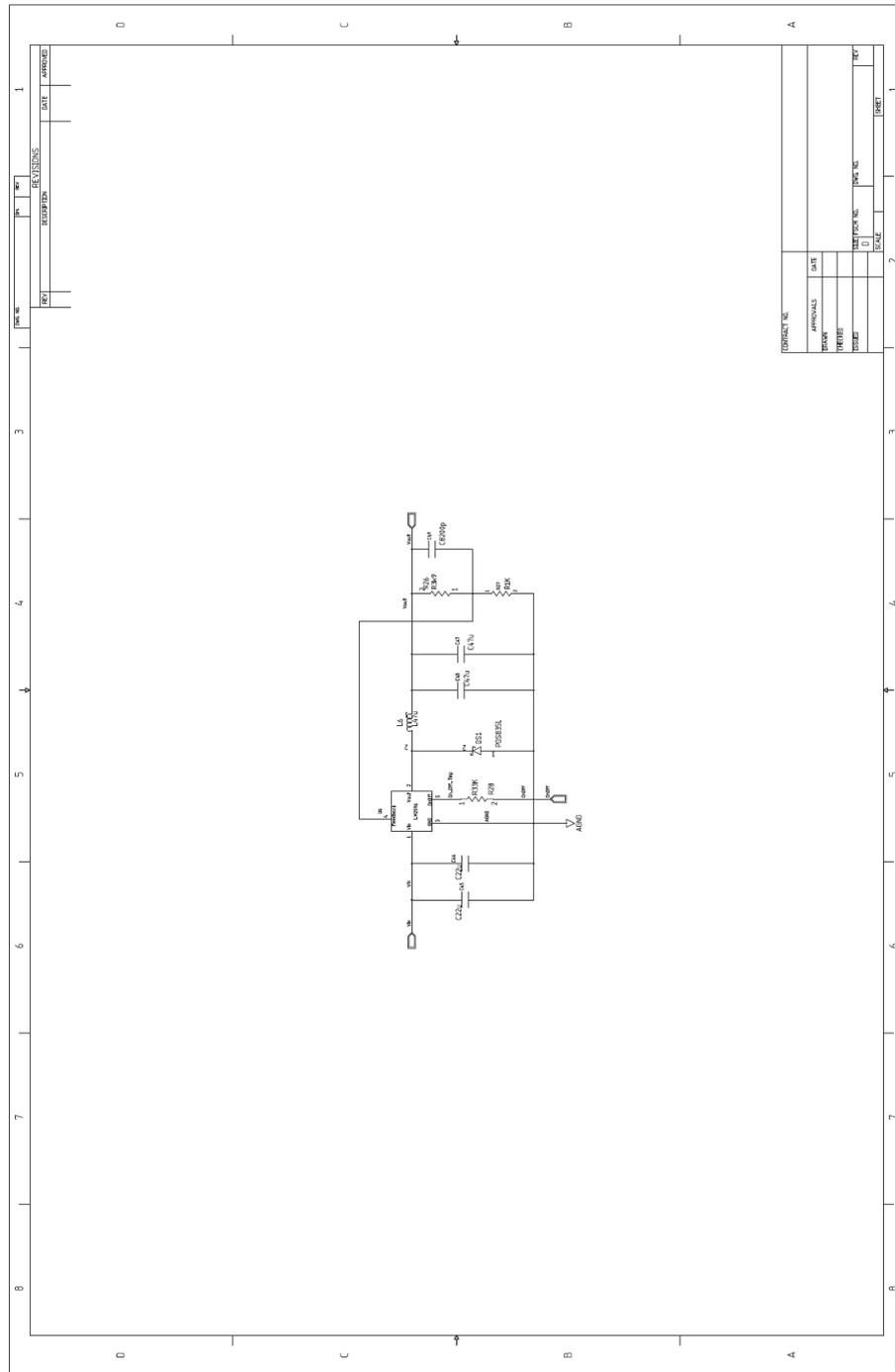


Figura 6.17. Schema elettrico LM2596



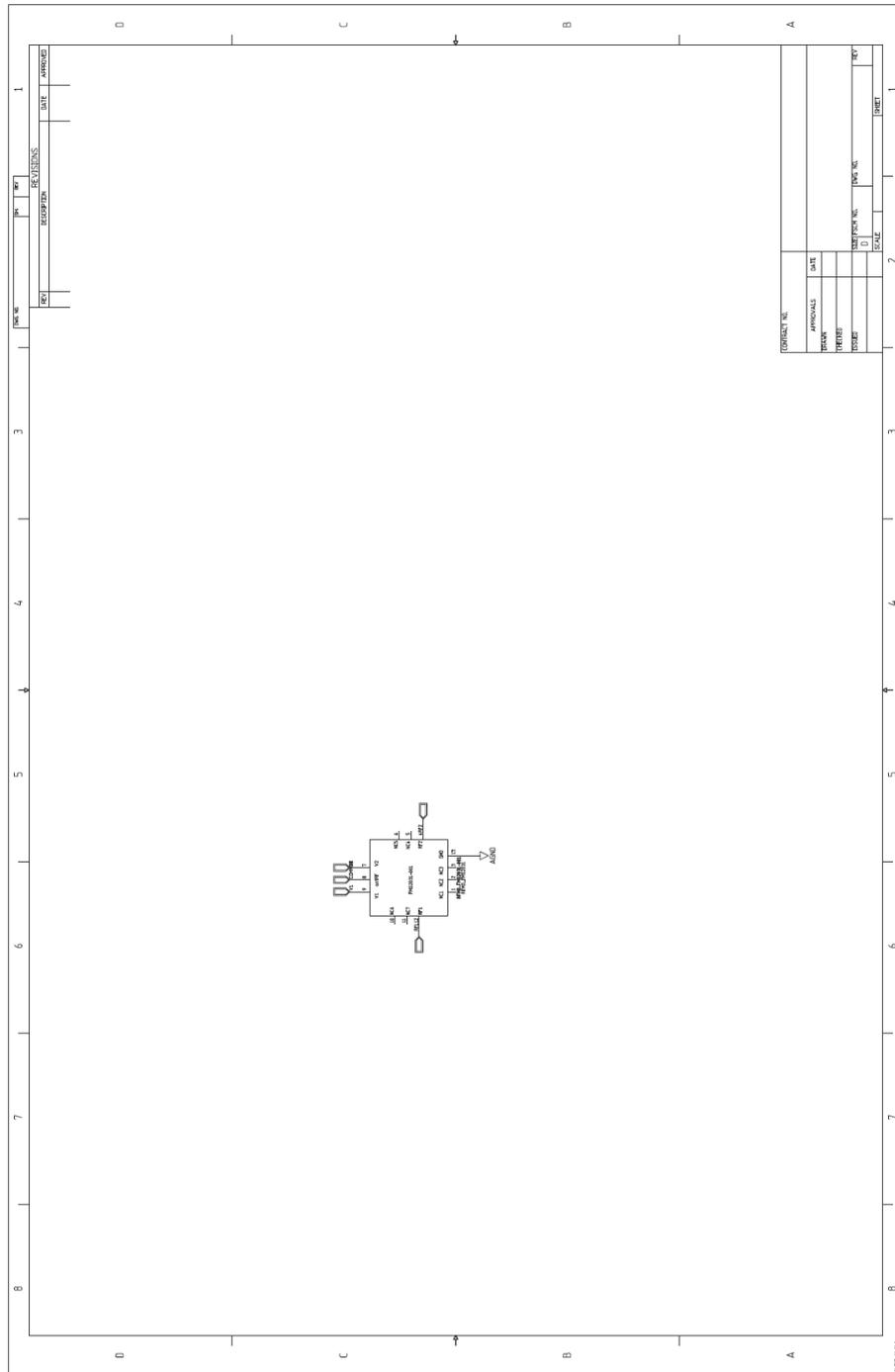


Figura 6.19. Schema elettrico FMS2031-001

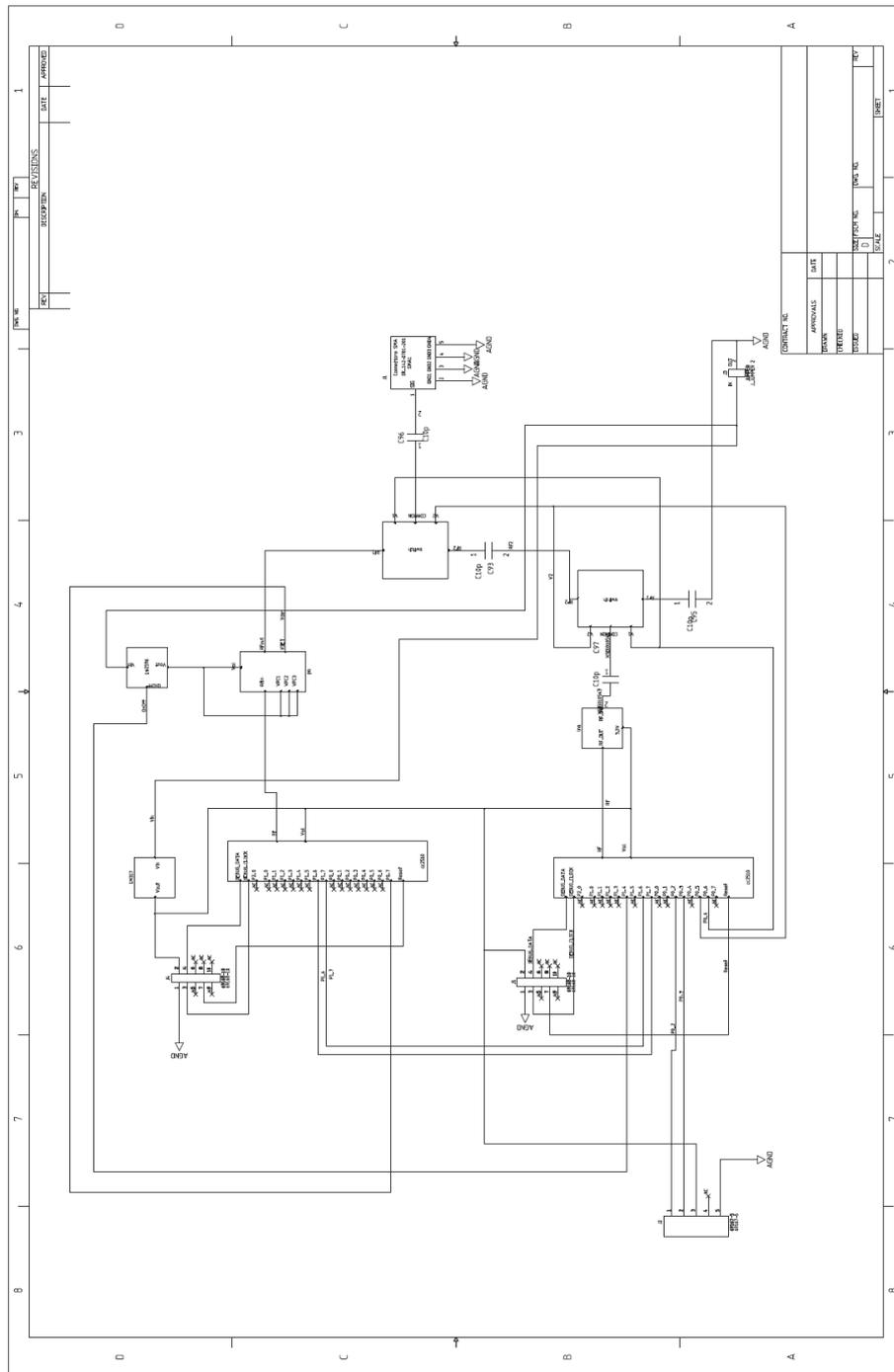


Figura 6.20. Schema SchedaTxRx

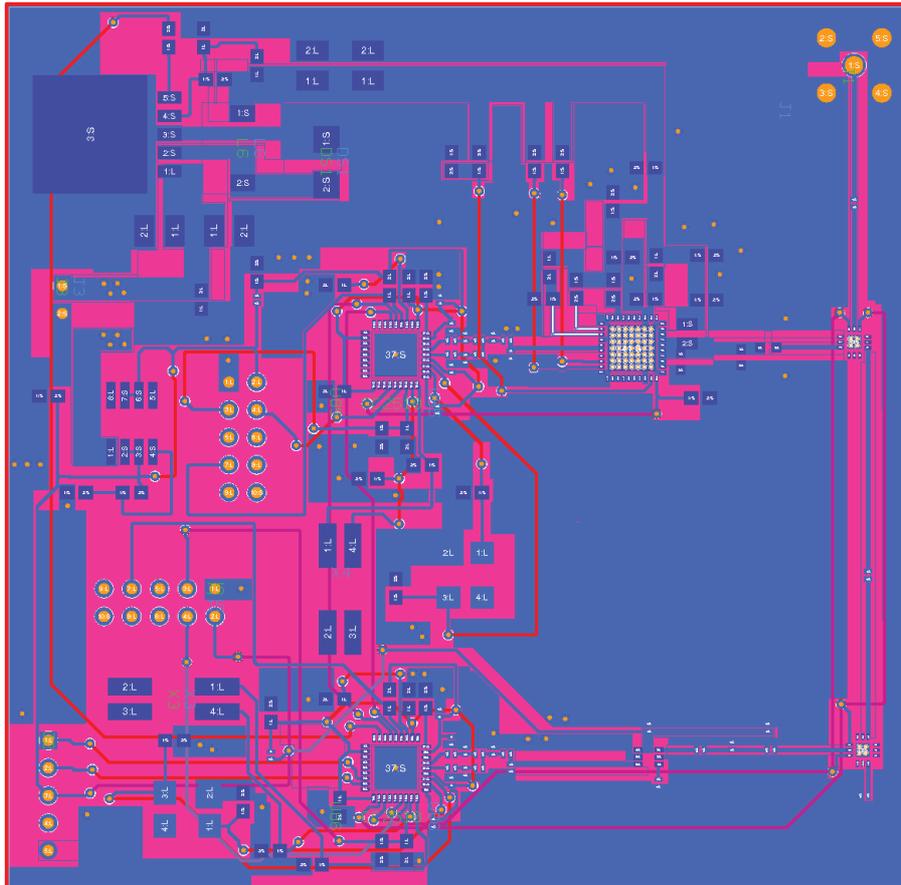


Figura 6.21. PCB schedaTxRx

# Capitolo 7

## Conclusioni

Il lavoro di tesi svolto nel corso di circa due anni è consistito nell'analisi, la sintesi e la realizzazione, software e hardware, della parte del progetto che concerne uno dei tre moduli di cui si compone il satellite Aramis (Architettura Modulare per Satelliti): il modulo di comunicazione Telecommunication Tile che lavora alla frequenza di 2.4 GHz.

Per quanto concerne la parte software, le problematiche sono state affrontate elaborando dei sistemi di codice di trasmissione e ricezione dati da parte del transceiver.

A tale fase di ricerca è seguito un lavoro di verifica dell'efficacia dei software elaborati che ha comportato una concentrazione altrettanto ampia sulla parte pratica del progetto, la quale mi ha fornito l'opportunità di testare praticamente la parte di elaborazione dati all'interno di un complesso progetto di ricerca.

Tutto ciò ha costituito la base per l'inizio di un lavoro sulla parte hardware che è approdato alla realizzazione, sia progettuale che fisica, di un amplificatore in grado di fornire potenza utile in trasmissione durante una comunicazione in ambito spaziale.

Dopo la creazione del sistema di trasmissione mi è stato possibile mettere insieme le diverse parti del lavoro precedente ed attualizzarlo nella progettazione dell'intero sistema di comunicazione ideando un modello di scheda da noi denominata *SchedaTxRx*.

Oltre al valore scientifico dell'esperienza, collaborare ad un progetto di questo tipo ha aggiunto un aspetto formativo legato al lavoro effettuato in team a cui ho potuto prendere parte.

Nello specifico, il progetto si compone dello sviluppo software di trasmissione e ricezione di dati via radio, con l'analisi di opportuni casi d'errore, utilizzando come ambiente di sviluppo l'IAR Embedded Workbench Ide.

Questa fase ha comportato la conoscenza e lo studio delle funzionalità comportamentali di un transceiver, oltre all'analisi dei fenomeni di interferenza tipici nelle

comunicazioni wireless.

Sono stati effettuati dei test che hanno permesso di acquisire pratica nell'utilizzo di apparecchiature e nell'impiego del pacchetto software *Labwindows*, per creare interfacce grafiche e visualizzare i risultati ottenuti.

Il software di comunicazione via radio è stato completato ma la versione definitiva dovrà essere anche in grado di permettere la comunicazione tra il transceiver e l'OBC *On Board Processor*, il cui protocollo è ancora in fase di implementazione; esso dovrà inoltre misurare una serie di parametri di comunicazione.

Il dimensionamento del link budget, effettuato al fine di valutare i parametri caratterizzanti un sistema di comunicazione quali la potenza in trasmissione e in ricezione, ha consentito l'applicazione dei principi di telecomunicazione.

Il lavoro che ho effettuato su questo progetto consta anche dello sviluppo della parte che riguarda l'hardware, dalla realizzazione dell'amplificatore di potenza alla progettazione dell'intero sistema di comunicazione.

Alla base di ciò è stata utile l'esperienza conseguita nella comprensione di datasheet di componenti e nella realizzazione di circuiti stampati, mediante l'utilizzo dei tool della suite di Mentor Graphics, con un'accurata analisi sia nel campo dell'elettronica di potenza sia nel campo dell'elettronica per la radiofrequenza.

In particolare la realizzazione fisica dell'amplificatore di potenza ha consentito l'apprendimento della pratica nella fase di saldatura di componenti SMD e through, e in quella di test di funzionamento di una scheda elettronica.

La schedaTxRx non è ancora stata realizzata fisicamente a causa di tempistiche e decisioni realizzative del progetto in quanto è tuttora in fase di definizione la dimensione che essa dovrebbe avere all'interno del satellite.

# Appendice A

## Il CD-ROM allegato

Il software sviluppato nel lavoro di tesi è contenuto nel cd allegato ed è suddiviso nei seguenti file:

- *firmware del programmatore* contenente il codice delle istruzioni da scrivere nella memoria flash del CC2510 seguito dal file.hex contenente la codifica in esadecimale di tali istruzioni e dalle relative librerie:
  - debug.c
  - debug.h
  - XMODEM.c
  - XMODEM.h
  - cc2510reg.h
  - commands.h
- *software del programmatore*, che realizza l'interfaccia grafica per la visualizzazione del set di operazioni inerenti al CC2510, con le relative librerie:
  - commands.c
  - define.h
  - ProgramCC2510.c
  - serial.c
- *software di trasmissione e ricezione di pacchetti di dati* presente sia in versione debug sia in versione release con le relative librerie:
  - rf.c
  - RF.h

- uart.c
  - uart.h
  - timer1.c
  - timer1.h
  - radio.h
  - hal.h
- *software dell'interfaccia grafica*, che permette la visualizzazione del funzionamento dei comandi di trasmissione e ricezione con la possibilità di configurare dall'esterno i registri del CC2510, con le relative librerie:
    - cmd.c
    - cmd.h
    - gnd437.c
    - gnd437.h
  - *software del programma di test per il calcolo della probabilità d'errore* in ricezione di un alto numero di pacchetti trasmessi con la possibilità di fornire dall'esterno i parametri di valutazione e vederne il risultato conseguito, con le relative librerie:
    - config\_panel.h
    - hal.h
    - main.c
    - main.h
    - rf.c
    - RF.c
    - uart.c
    - uart.h
    - timer1.c
    - timer1.h
    - radio.h

# Bibliografia

- [1] Lars Alminde, Morten Bisgaard, Dennis Vinther, Tor Viscor, and Kasper Østergard, “Educational value and lessons learned from the AAU-CUBESAT project,” *Recent Advances in Space Technologies, 2003. RAST '03. International Conference on. Proceedings of*, 2003. [www.studentspace.aau.dk/publications/RAST2003.pdf](http://www.studentspace.aau.dk/publications/RAST2003.pdf).
- [2] Stephen Waydo, Daniel Henry, Mark Campbell, “CubeSat Design for LEO-Based Earth Science Missions,” *Aerospace Conference Proceedings, 2002*, 2002. [www.cds.caltech.edu/~waydo/papers/IEEE2002.pdf](http://www.cds.caltech.edu/~waydo/papers/IEEE2002.pdf).
- [3] S.Speretta, L.Reyneri, C.Sansoè, C.Passerone, “PICPOT: A Nanosatellite From Turin Polytechnic,” *Student Space Science and Technology Symposium (Patras (GR)) ottobre 2006*, [http://polimage.polito.it/group/papers/picpot\\_sts2006.pdf](http://polimage.polito.it/group/papers/picpot_sts2006.pdf).
- [4] Politecnico di Torino, “La Ricezione della Telemetria di Picpot”, Maggio 2006 [http://polimage.polito.it/picpot/documentazione/La\\_Telemetria\\_di\\_Picpot.pdf](http://polimage.polito.it/picpot/documentazione/La_Telemetria_di_Picpot.pdf)
- [5] Sebastiano Tirrò “Satellite Communication Systems Design,” 1993 Plenum Press, New York.
- [6] Leon W., Couch, 2, “Digital and Analog Communication Systems, Seventh Edition,” 2007 by Pearson Education.
- [7] CC2510F32, “Low-Power SoC (System-on-Chip) with MCU, memory, 2.4 GHz RF Transceiver, and USB, (Rev.F), 11 Jul 2008”, <http://focus.ti.com/lit/ds/symlink/cc2510f32.pdf>.
- [8] Chipcon Products from Texas Instrument, “CC1110/CC2430/CC2510, Debug and Programming Interface Specification, 3 May 2007, Rev.1.2”, <http://focus.ti.com/lit/ds/symlink/cc2510f32.pdf>.
- [9] CC2510.[Online]. <http://focus.ti.com/docs/prod/folders/print/cc2510f32.html>.
- [10] Chipcon Products from Texas Instrument, ‘Quick Start Instructions CC1010DK Development Kit, 30 Jan 2006,’ <http://focus.ti.com/lit/ug/swru056/swru056.pdf>.
- [11] Microcontroller technology, “the 68HC 11 di Peter Spasov,” Prentice-Hall, 1993.

- [12] Chuck Forsberg, "XMODEM/YMODEM PROTOCOL REFERENCE", *XMODEM and YMODEM, File Transfer Protocols, 10-2-88*, <http://moscova.inria.fr/~doligez/zmodem/ymodem.txt>.
- [13] Microsym Computers Inc. "Hexadecimal Object File Format Specification, Revision A, January 6, 1988", <http://microsym.com/editor/assets/intelhex.pdf>.
- [14] Sirenza Microdevices, "SZM-2166Z, 2.3 GHz-2.7 GHz, 2 W Power Amplifier," <http://www.alldatasheet.com/datasheet-pdf/pdf/265520/SIRENZA/SZM-2166Z.html>.
- [15] Renato Orta, "Teoria delle Linee di Trasmissione," Febbraio 2002, CLUT.
- [16] Giorgio Taricco , "Comunicazioni elettriche con elementi di Teoria dell'Informazione", CLUT Seconda Edizione 2002.
- [17] Mario Orefice, "Propagazione delle onde elettromagnetiche,"
- [18] Irfan Ali, Member, IEEE, Naofal Al-Dhahir, Member, IEEE, and John E. Hershey, Senior Member, IEEE, "Doppler Characterization for LEO Satellites," *IEEE Transactions on communications, Vol. 46, No. 3, March 1998*.
- [19] James F. Kurose, Keith W. Ross, "Internet e Reti di Calcolatori," McGraw-Hill, Seconda Edizione, 2003.
- [20] V.Carl Hamacher, Zvonko G. Vranesic, Safwat G- Zaky, "Introduzione all'architettura dei Calcolatori," McGraw-Hill 1996.
- [21] Linghang Fan, Haitham Cruickshank, Zhile Sun, "Ip Networking over, Next Generation, Satellite Systems," International Workshop, Budapest, July 2007.
- [22] Adrian Hooke and Richard Desjardins, Member, IEEE, "Communication Protocol Standards for Space Data Systems" *Proceedings of the IEEE, Vol. 78, No. 7, July 1990*.
- [23] Aldo Roveri, "Comunicazioni mobili 1", Univ. di Roma "La Sapienza" ,2008-2009.
- [24] David Halliday, Robert Resnick, "Fisica 1," Terza Edizione.
- [25] Maxim, "2.4 GHz SiGe, High IP3 Low-Noise Amplifier, 19-1786; Rev1, 11 Marzo," <http://datasheets.maxim-ic.com/en/ds/MAX2644.pdf>
- [26] National Semiconductor, "LM2596, Simple Switcher Power Converter 150 KHz 3A Step-Down Voltage Regulator, May 2002," <http://www.national.com/ds/LM/LM2596.pdf>.
- [27] Filtronic, "FMS2031-001, 10 Watt, GaAs Wide Band SPDT Switch," <http://pdf1.alldatasheet.com/datasheet-pdf/view/225003/FILTRONIC/FMS2031-001.html>.
- [28] National Semiconductor, "LM317DT,3 Terminal Adjustable Regulator, July 2004," <http://pdf1.alldatasheet.com/datasheet-pdf/view/8616/NSC/LM317DT.html>.
- [29] Politecnico di torino, "Appunti relativi al corso di Elettronica," *Dalle Lezioni del Prof. Vincenzo Pozzolo*.

- [30] Mario Pent, “Dispense del corso di Ponti radio e Satelliti,” Torino, 2003.
- [31] Roberto Garelo, “ Lucidi del corso di Trasmissione Numerica,”.
- [32] Nico Grilloni, “Gli alimentatori stabilizzati. Progetto e calcolo,” 2005.
- [33] Franco Maddaleno, “Alimentatori a commutazione,” Politeko, Torino, 2007.
- [34] Politecnico di Milano, “Microsoft PowerPoint, Dimensionamento di una microstriscia.ppt,” Dal corso di microonde, 2004-2005, [http://home.dei.polimi.it/macchiar/microondeI/lucidi/Esercitazione11\\_05\\_2005.pdf](http://home.dei.polimi.it/macchiar/microondeI/lucidi/Esercitazione11_05_2005.pdf).
- [35] Carlo Naldi, “ Lucidi del corso di Dispositivi alle alte frequenze e optoelettronici,” 2002.
- [36] G.Ghione, M.Piroli, “Elettronica delle microonde,” Parte Seconda, Otto editore, 2002.
- [37] Maria Caldaralo, Maurizio Tranchero, “DocuMentor, come realizzare circuiti stampati con L’Expedition Flow di Mentor Graphics,” 2007.