FRONTESPIZIO

Sommario

Questo elaborato vuole fornire tutti gli strumenti necessari per progettare un ricetrasmettitore sul satellite Atmocube; fornisce specifiche, funzioni, modalità di progettazione e altre variabili che un progettista deve considerare per questa tipologia di componenti. S'è cercato di toccare tutti i punti in maniera approfondita, dato che il satellite è ancora in fase di sviluppo e , verosimilmente, alcuni dati che ora abbiamo considerato non saranno gli stessi utilizzati per l'elaborazione finale, cosicché chi dovrà in futuro riprendere in mano la parte del ricetrasmettitore potrà sfruttare il lavoro già fatto senza troppe difficoltà. L'impegno è quello di produrre un documento che sia il più esaustivo e completo possibile.

Indice

ELENCO TABELLE…….....7

ELENCO FIGURE ……..8

1. **INTRODUZIONE**……...10

1.1 Il satellite Atmocube……...……..10

1.2 La stazione a terra……...……..11

1.3 La frequenza e la modulazione…….11

1.4 *Link Budget*…….....15

1.4.1 *Uplink*……...15

1.4.2 *Downlink*……..18

2. **SPECIFICHE DI PROGETTO**……..21

2.1 Gli obiettivi……...21

2.2 Dati di progetto……22

2.2.1 Parte di trasmissione…….....….....22

2.2.2 Parte di ricezione……..23

2.2.3 Parte di controllo……..23

2.2.4 Parte di alimentazione…….……24

2.3 Scelte……26

2.3.1 ADF7020…….27

2.3.2 CC1100……....30

2.3.3 Comparazione fra ADF7020 e CC1100…….....36

2.3.4 Motivazioni della scelta del CC1100…….....36

2.3.5 Piano di lavoro…….....37

3. **IL PROGETTO**…….....38

3.1 Schema a blocchi…….....38

3.2 Parte di alimentazione…….....40

3.3 Circuito di connessione al *Power Amplifier* …….....40

3.4 Circuito di connessione al *Low Noise Amplifier*…….....41

3.5 Schema elettrico completo…….....43

3.6 Specifiche di controllo…….....44
3.7 Il prototipo…….....51
3.8 Misure…….....58
3.8.1 Misura 1…….....59
3.8.2 Misura 2…….....63
3.8.3 Misura 3…….....68
3.8.4 Conclusioni sulle misure…….....69

4. RF STUDIO…….....71

5. CONCLUSIONI.....76
5.1 Obiettivi raggiunti…….....76
5.2 Per il futuro…….....77

6. APPENDICE…….....78

7. BIBLIOGRAFIA.....86

RINGRAZIAMENTI…….....87

Elenco Tabelle

Tabella 1: f e λ in *downlink* e *uplink*…….....12
Tabella 2: *uplink*…….....16
Tabella 3: *downlink*…….....19
Tabella 4: valori componenti esterne ADF7020…….....29
Tabella 5: caratteristiche ADF7020…….....30
Tabella 6: caratteristiche CC1100…….....33
Tabella 7: CC1100 vs. ADF7020…….....36
Tabella 8: valori componenti esterni CC1100 a 433 MHz…….....44
Tabella 9: esito misura 1…….....62
Tabella 10: esito misura 2…….....67
Tabella 11: esito misura 3…….....68

Elenco figure

Figura 1: schema a blocchi satellite Atmocube……..……...10
Figura 2: diagramma di *scattering* per una BPSK…….....13
Figura 3: *bit rate uplink* vs. angolo di elevazione…….....17
Figura 4: *bit rate downlink* vs. angolo di elevazione…….....20
Figura 5: sezione Tx e Rx del satellite…….....21
Figura 6: sistema di alimentazione del ricetrasmettitore…….....25
Figura 7: struttura interna dell'ADF7020…….....27
Figura 8: schema a blocchi completo del ricetrasmettitore utilizzando l'ADF7020.....29
Figura 9: principali blocchi del CC1100…….....31
Figura 10: configurazione consigliata per CC1100 a 433 MHz…….....32
Figura 11: struttura interna del CC1100 …….....38
Figura 12: *top-view* dei pin del CC1100……..... 39
Figura 13: dettaglio dei principali blocchi d'alimentazione del ricetrasmettitore…….....40
Figura 14: schema a blocchi circuito di connessione al PA…….....41
Figura 15: schema elettrico completo del CC1100…….....43
Figura 16: *automatic power-on reset*…….46
Figura 17: *manual reset*…….....47
Figura 18: struttura pacchetto…….....48
Figura 19: schema di funzionamento del *whitening*…….....50
Figura 20: schema di funzionamento dell'*interleaving*…….....51
Figura 21: CC1100 montato su *evaluation board* con configurazione a 433 MHz….....52
Figura 22: scheda con µC…….....53
Figura 23: schema elettrico traslatore di livello…….....54
Figura 24: traslatore di livello…….....55
Figura 25: sistema completo del ricetrasmettitore numero 1…….....56
Figura 26: sistema completo del ricetrasmettitore numero 2…….....57
Figura 27: schema a blocchi sistema di misura…….....57
Figura 28: sistema sviluppato in laboratorio…….....58
Figura 29: attenuazione del sistema completo…….....59
Figura 30: attenuazione cavo coassiale…….....60
Figura 31: attenuatore variabile …….....61
Figura 32: particolare dell'attenuatore fisso montato sull'attenuatore variabile…….....61

Figura 33: oscilloscopio e analizzatore di modulazione…….....63
Figura 34: sequenza di preambolo…….....64
Figura 35: analizzatore di modulazione con valori per 34.4 kbps…….....64
Figura 36: spettro segnale di preambolo a 433 MHz, *bit rate* 34.4 kbps, scala 50 kHz…….....65
Figura 37: analizzatore di modulazione con valori per 10 kbps…….....66
Figura 38: spettro segnale di preambolo a 433 MHz, *bit rate* 10 kbps, scala 25 kHz…….....66
Figura 39: spettro segnale di preambolo a 433 MHz, *bit rate* 10 kbps, scala 50 kHz…….....67
Figura 40: spettro segnale di preambolo a 433 MHz, *bit rate* 10 kbps, scala 500 kHz…….....67
Figura 41: *screen-shot* RF studio, *normal view*……71
Figura 42: dettaglio impostazioni RF Studio…….....72
Figura 43: *screen-shot* RF studio, *register view*……...73

1. Introduzione

1.1 *Il satellite Atmocube*

Il satellite Atmocube è un progetto sperimentale dell'Università degli Studi di Trieste, in cui gli studenti hanno il compito di progettare un nano satellite che sia in grado di effettuare alcune misure nell'atmosfera. Il satellite ha le dimensioni di 13 x 13 x 13 cm, è il suo scopo è quello di rilevare a vari livelli dell'atmosfera terrestre i valori della densità del flusso magnetico, dell'intensità di radiazione del vento solare e della densità dell'alta atmosfera. Per mandarlo in orbita si sfrutterà una spedizione spaziale russa, la quale lo rilascerà nell'atmosfera; da quel momento inizierà la vita dell'Atmocube. Il sistema sarà composto da sottosistemi collegati e comunicanti tra loro, figura 1. C'è la sezione dedicata alla strumentazione di bordo per compiere le misure, quella per l'alimentazione del satellite, quella dedicata alla ricezione e alla trasmissione dati, e quella dedicata al microcontrollore.

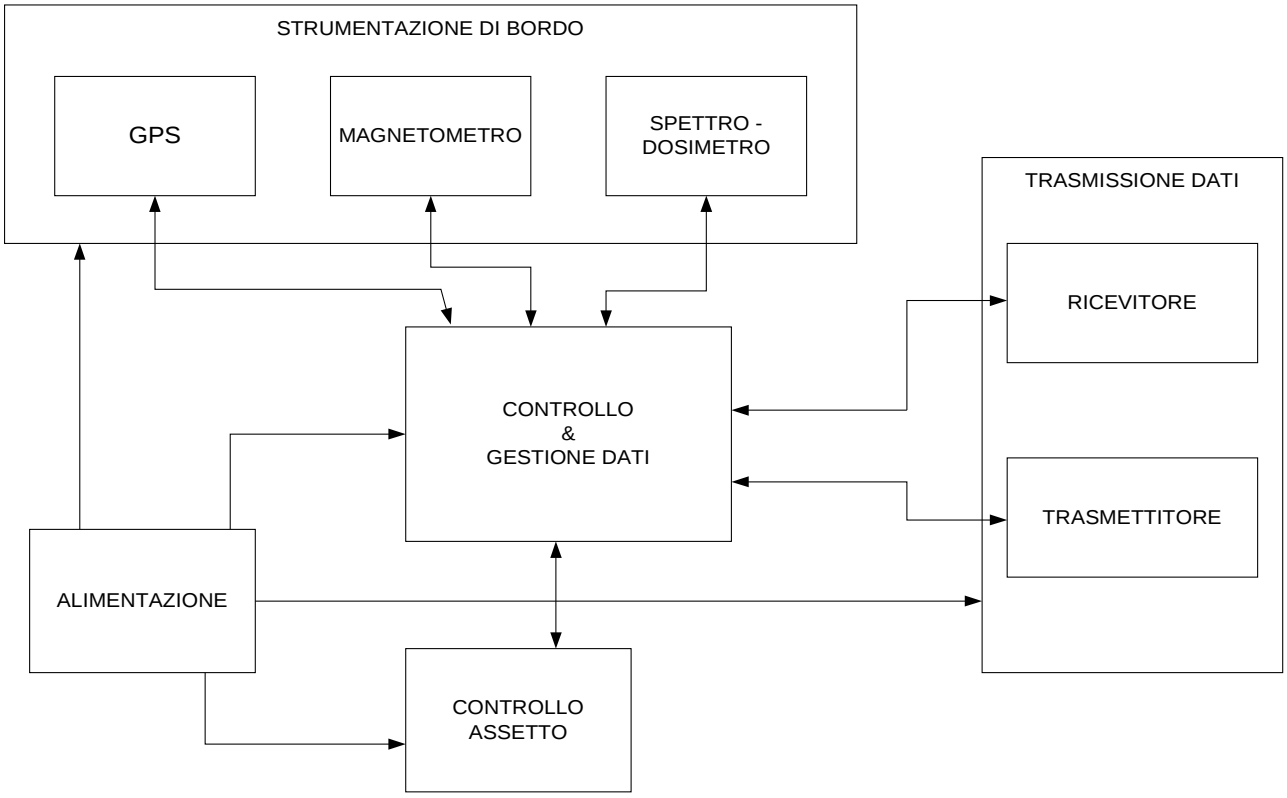

Figura 1: schema a blocchi satellite Atmocube

Lo scopo di questo studio è analizzare ogni passo della progettazione del ricetrasmettitore per un nano satellite; esso dovrà tener conto delle specifiche del sistema, quali possono essere per esempio la potenza trasmessa, la frequenza RF di lavoro, il tipo di modulazione e dovrà tener conto anche

delle componenti già esistenti come la parte dell'alimentazione, il µC ma anche di un fattore importante come la base di ricezione dei dati a terra.

1.2 *La stazione a terra*

Al fine della progettazione è fondamentale stabilire quali caratteristiche abbia la stazione a terra. E' stata scelta come sede per la stazione l'Osservatorio Astronomico di Basovizza, dato che è già stato utilizzato per questi scopi nei primi anni '90, in occasione della missione ItalSat. E'necessario considerare le caratteristiche di queste apparecchiature per scegliere il ricetrasmettitore del satellite più adatto; si deve tener conto della sensibilità, della sensitività , della larghezza di banda (430- 440 MHz) e della potenza in uscita dall'apparecchio della stazione di terra. Importante è il tipo di modulazione supportato; di per se la stazione a terra è dotata di un TNC (*Terminal Mode Controller*) che si occupa dell'incapsulamento dei dati nei pacchetti, con modem interno AFSK (*Audio Frequency Shift Keying*). Nelle trasmissioni satellitari è preferibile adoperare una PSK (*Phase Shift Keying*), quindi si è deciso che il modem interno del TNC venga disabilitato, e che venga utilizzato in sostituzione un modem esterno BPSK (*Binary Phase Shift Keying*)), cioè una modulazione binaria di fase in banda traslata, che è vantaggiosa in termini di rapporto segnale rumore, banda occupata e utilizzo di un alfabeto binario. E' chiaro quindi che, nel merito della scelta dei componenti per il ricevere e trasmettere dal satellite alla stazione di terra, queste variabili dovranno essere necessariamente considerate.
Procedendo con lo studio si vedrà che il circuito integrato da noi scelto adotterà una modulazione di frequenza, FSK (Frequency Shift Keying), non compatibile con la stazione di terra di Basovizza, pertanto sarà necessario progettare una stazione ricevente adeguata alla nostra componente.

1.3 *Le frequenze e la modulazione*

Il satellite Atmocube lavora in banda UHF, utilizzata sia in trasmissione dati che in ricezione. La scelta è caduta su questa gamma di frequenza data la necessità di operare a valori maggiori di 100 MHz per riuscire a penetrare gli strati atmosferici, e anche perché sono frequenze sfruttate nel campo radio-amatoriale, quindi gratuite, peculiarità importante dato il nostro scopo didattico. I valori reali non sono precisamente quelli riportati in tabella 1. ma è stata utilizzata una approssimazione accettabile. Inizialmente era stato deciso d'utilizzare la banda VHF per l'utilizzo del *beacon*, ipotesi

successivamente scartata per motivi pratici di progettazione, dato che l'implementazione risulta più agevole ed inoltre non è necessario rispettare alcuno standard particolare.

	Banda UHF [70 cm]	
	Downlink	Uplink
f [MHz]	437,490	437,847
λ [m]	0,68	0,68

Tabella 1: f e λ in *downlink* e *uplink*

Come già accennato nel paragrafo precedente è possibile considerare la modulazione in uso alla stazione di Basovizza, cioè una BPSK, oppure considerare come stazione ricevente a terra una strumentazione che supporti la BFSK (*Binary Frequency Shift Keying*).
La modulazione B-PSK presuppone un vocabolario composto da due simboli, ognuno dei quali verrà assegnato ad una forma d'onda che sarà utilizzata per trasmettere l'informazione attraverso il canale. Caratteristica principale della modulazione è che l'informazione è trasmessa nella fase del segnale, il quale avrà la generica forma:

$$u(t) = A\cos\left(2\pi f_C\ t + \varphi(t)\right)$$

con f_C frequenza della portante e $\varphi(t)$ fase del segnale modulato.
Applicando una semplice formula si può vedere che questo tipo di modulazione assegna ai propri simboli la medesima energia per simbolo (nel nostro caso per bit):

$$E_m = \int_0^T u_m{}^2\ (t)dt = \int_0^T s_m^2(t)\cos^2(2\pi f_c t)dt = \frac{1}{2}\int_0^T s_m^2(t)dt + \frac{1}{2}\int_0^T s_m^2(t)\cos(4\pi f_c t)dt = \frac{1}{2}\int_0^T s_m^2(t)dt = E_s$$

infatti l'energia media per simbolo coincide con l'energia del simbolo.
Alla luce di questo fatto, analizzando il diagramma di *scattering*, figura 2, di una generica B-PSK vediamo che ha la stessa forma di una 2-PAM antipodale, cioè una modulazione d'ampiezza in banda base; questa osservazione è utile per considerare la probabilità d'errore sul bit, dato che si considera la distanza tra due bit successivi :

$$P_e = Q\left(\frac{d}{\sqrt{2N_o}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

con *d* distanza tra i due simboli dell'alfabeto binario, N_o rumore gaussiano presente e E_b energia per bit.

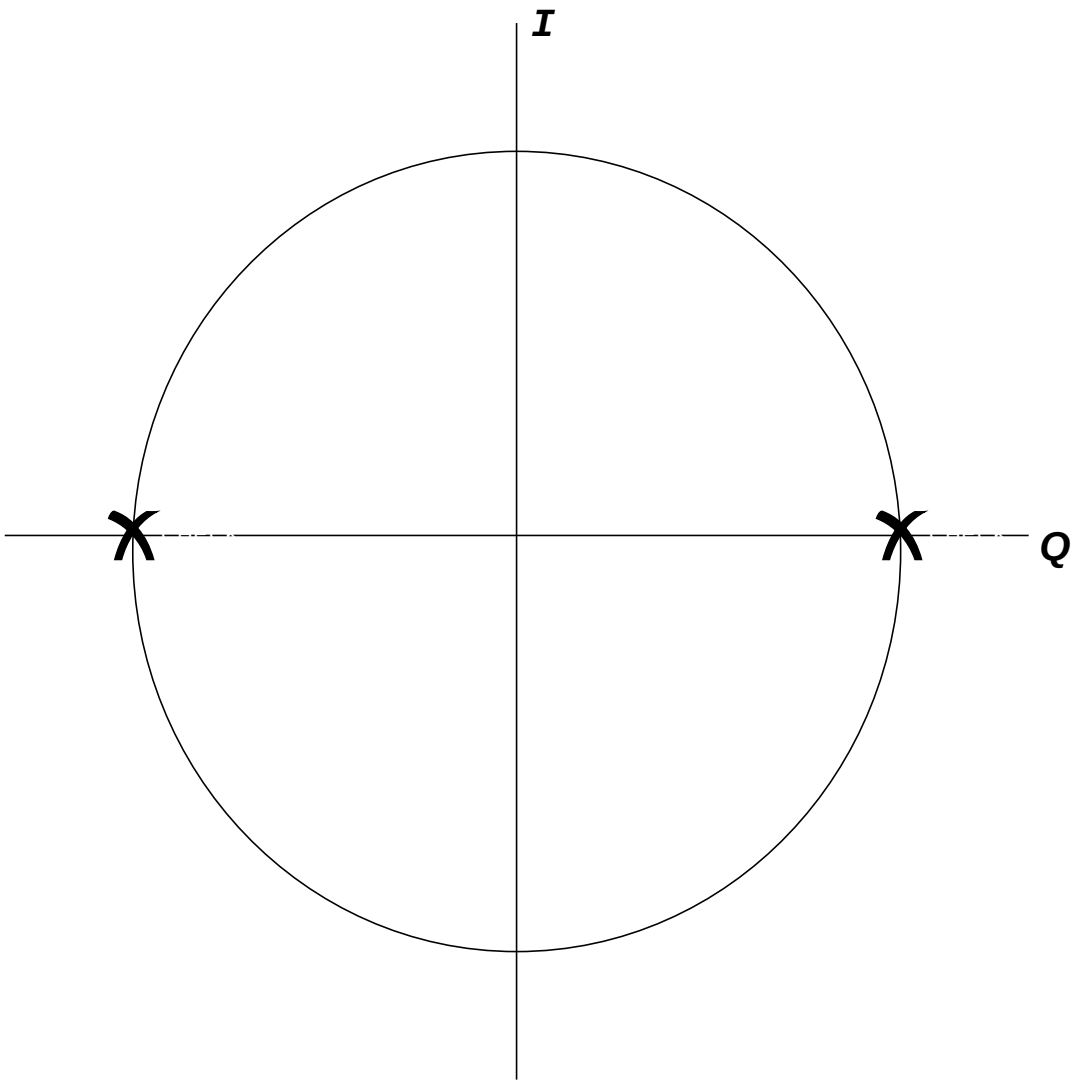

Figura 2: diagramma di *scattering* per una B-PSK

E'possibile una semplice analisi delle prestazioni in termini di banda e d'efficienza spettrale; la modulazione di fase ha buone prestazioni in termini di banda, ma in termini d'efficienza spettrale perde qualcosa rispetto ad altre modulazioni quali potrebbero essere quelle per segnali ortogonali (FSK, PPM), ma questo nel caso il numero di bit fosse >2. Nel nostro caso la modulazione B-PSK va più che bene, perché limita la banda e il rapporto segnale rumore è accettabile per l'uso che ne dobbiamo fare.

Per quanto riguarda la FSK, l'informazione viene portata dalla frequenza del segnale modulato. Un generico segnale FSK ha la forma:

$$u_m(t) = \sqrt{\frac{2E_S}{T}} \cos 2\pi(f_C + m\Delta f)\ t$$

e con Δf intervallo tra due frequenze successive , E_S energia per simbolo e T intervallo di segnalazione.

Anche in questa modulazione, come per la BPSK, ogni simbolo ha la stessa energia, e alla luce di questo possiamo considerare lo stesso diagramma di *scattering*. Una caratteristica interessante è porre una condizione sul valore minimo che Δf può assumere, in modo da avere forme d'onda ortogonali, quindi incorrelate:

$$\Delta f_{\min} = \frac{1}{2T} \quad [\text{Hz}]$$

Questa condizione di incorrelazione è utile in quanto, se un bit è errato, non influenza i bit successivi. La probabilità d'errore di una modulazione ortogonale binaria di fase è :

$$P_e = Q\left(\sqrt{\frac{E_S}{N_0}}\right)$$

con E_S energia per simbolo(che in una modulazione binaria coincide con energia per bit) e N_0 rumore gaussiano.

Non è ottimale in termini di banda, *W*, la FSK dato che:

$$W \cong \frac{M}{2T} \quad [\text{Hz}]$$

con M numero di simboli utilizzati. Da questo si capisce che all'aumentare del numero di forme d'onda anche la banda aumenta; invece in termini d'efficienza spettrale:

$$r \cong \frac{R_s}{W} \quad [(\text{bit\simbolo})\backslash \text{ Herz}]$$

con Rs *simbol rate*, cioè numero di bit per simbolo. Da queste considerazioni capiamo che all'aumentare del numero di forme d'onda si ha un uso efficiente dell'energia a disposizione; questa modulazione è ideale per sistemi a potenza limitata che hanno una banda sufficientemente larga per consentire un elevato numero di segnali.

1.4 *Link budget*

Il *link budget* è la somma di tutti i guadagni e perdite nella tratta dal trasmettitore, attraverso il canale, al ricevitore in un sistema di telecomunicazioni. Vengono considerati sia le attenuazioni del segnale trasmesso, dovute alla propagazione in un canale, sia i guadagni (o le perdite) dovute all'antenna ricevente. Pertanto, parlando più specificatamente del nostro progetto, il *link budget*, [8], definisce le caratteristiche in termini di potenza, attenuazioni e guadagni della modalità in *downlink* (dal satellite alla terra) e della modalità in *uplink* (dalla terra al satellite).
Il *link budget* permette di determinare i valori della potenza in trasmissione e del guadagno dell'antenna e, più in generale, permette di definire tutti gli aspetti che erano stati definiti in modo approssimato. É quindi uno degli elementi chiave nel progetto di un sistema spaziale, e permette di evidenziare molte caratteristiche delle prestazioni generali del sistema. In questa fase, inoltre, è possibile effettuare variazioni sulle caratteristiche dei componenti in modo da rendere il bilancio migliore possibile.
Ovviamente, i bilanci da fare sono due: uno per la tratta in *uplink* e uno per la tratta in *downlink*, perché, anche se si tratta dello stesso collegamento, i sistemi trasmittente e ricevente del satellite e delle stazione sono profondamente diversi.

1.4.1 *Uplink*

L'*uplink* è il bilancio riguardante la tratta terra-satellite; da un certo punto di vista questa stima è meno critica rispetto al *downlink* dato che la base a terra è ben conosciuta, si hanno dei dati certi sulle sue prestazioni ed eventuali guasti sarebbero prontamente corretti avendo la strumentazione a portata di mano. Per valutare l' *uplink* è necessario considerare tre variabili in particolare:

- Potenza trasmissibile del ricetrasmettitore Kenwood relativo alla banda UHF con modulazione FM
- Attenuazione della linea di trasmissione collegante il nostro ricetrasmettitore all' antenna direttiva situati nella base a terra
- Guadagno dell'antenna

E' chiaro che diversi valori saranno già noti come parametri tipici della strumentazione, in più si è lavorato sulle tre condizioni sopra citate per elaborare un *link budget* [8] che soddisfasse una BER di 10^{-5} e un rapporto segnale rumore che fosse ≈ 13,3 dB.

Di seguito , in tabella 2, troviamo nel dettaglio i valori relativi al *uplink* del nostro progetto.

NOME	SIMBOLO	UNITÀ	SORGENTE	VALORE
Potenza trasmessa	P	W	parametro	40
Potenza trasmessa	P	dBW	10log(P)	16,02
Attenuazione linea	L_l	dB	parametro	-2,7
Guadagno massimo antenna tx	G_{pt}	dB	Tabella4.7	15
Ampiezza cono a -3 dB	θ_t	$^\circ$	parametro	15
Offset di puntamento	e_t	$^\circ$	10% di θ	1,5
Perdita di puntamento	L_{pt}	dB	Eq. (4.7)	-0,12
Guadagno effettivo antenna tx	G_t	dB	Eq. (4.14)	14,88
EIRP	$EIRP$	dBW	Eq. (4.3)	28,20
Attenuazione di spazio libero	L_s	dB	Tabella 4.4	-148,2
Perdite di propagazione	L_a	dB		0
Guadagno massimo antenna rx	G_{pr}	dB	parametro	2
Ampiezza cono a -3 dB	θ	$^\circ$	parametro	78
Offset di puntamento	e	$^\circ$	parametro	15
Perdita di puntamento	L_{pr}	dB	Eq. (4.7)	-0,44
Guadagno effettivo antenna rx	G_r	dB	$G_{pr} + L_{pr}$	1,56
Temperatura di sistema	T_s	K	Eq. (4.13)	350,72
Perdita di implementazione (3)	—	dB	stima	-2
Data Rate	R	bit/s	parametro	9600
Banda del ricevitore	B	kHz	datasheet	58
E_b/N_0 (1)	E_b/N_0	dB	Eq. (4.2)	44,89
Carrier to Noise density ratio	C/N_0	dB $\cdot$ Hz	Eq. (4.15)	84,71
Received Isotropic Power	RIP	dBW	$EIRP - L_s$	-120
Portante	C	dBW	Eq. (4.16)	-100,81
Portante	C	W		$8,3 \cdot 10^{-11}$
Ampiezza del segnale		µV	$V = \sqrt{P \cdot R}$	64,42
BER richiesta	BER	—	parametro	10^{-5}
E_b/N_0 richiesto (2)	Req E_b/N_0	dB		13,3
Margine di sistema		dB	(1)-(2)+(3)	29,59

Tabella 2: *Uplink*

In figura 3 troviamo il grafico di come varia la *bit rate* in funzione dell'angolo di elevazione del satellite.

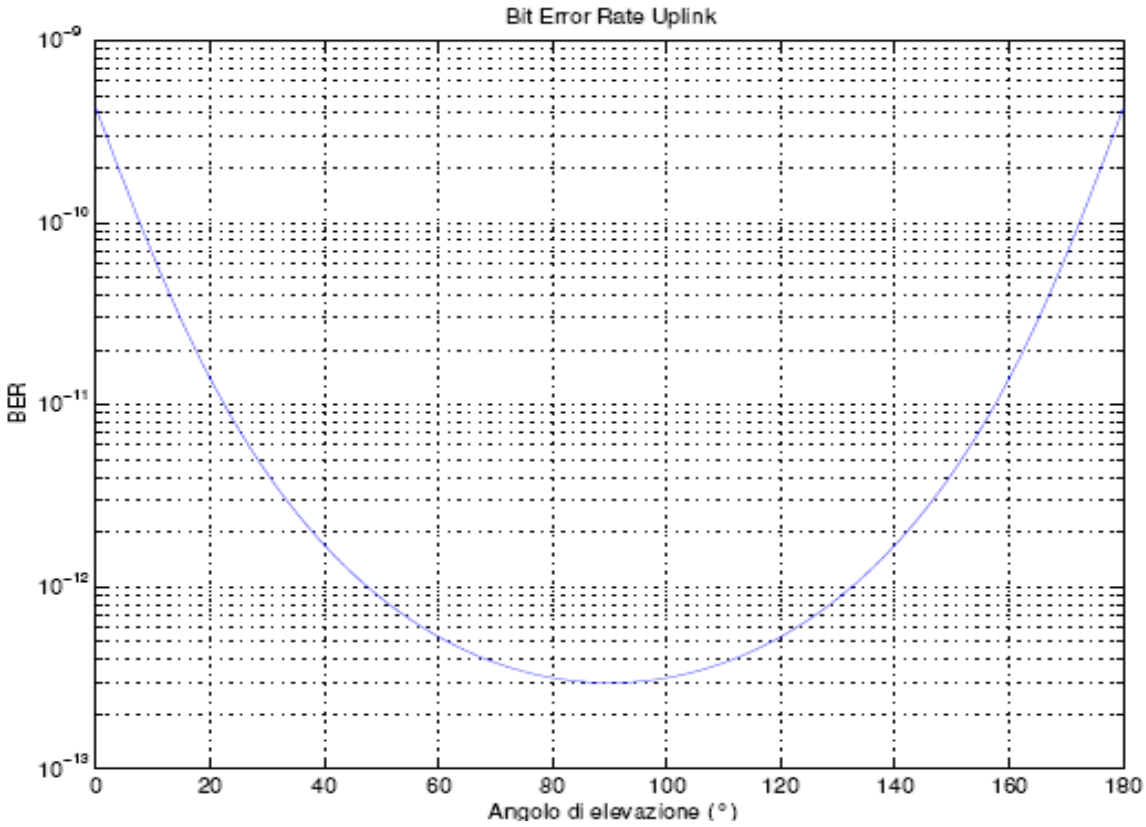

Figura : *Bit Rate Uplink* vs. angolo di elevazione

E' inoltre possibile calcolare la probabilità d'errore sul bit, considerando come modulazione la BFSK, utilizzando il rapporto $\frac{E_b}{N_0}$ .Come si legge dalla tabella 3, il rapporto $\frac{E_b}{N_0}$ in *uplink* è pari a 44,89 dB, pertanto la probabilità d'errore, considerando rispettate le condizioni d'ortogonalità di questa modulazione, sarà:

$$
\begin{aligned}
P_e &= Q\sqrt{\frac{E_b}{N_0}} \\
&= \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{2N_0}}\right) \\
&= \frac{1}{2} erfc\left(\sqrt{\frac{44{,}89}{2}}\right) \\
&= 1{,}04 \bullet 10^{-11}
\end{aligned}
$$

Da questo risultato si vede che in *uplink* la probabilità d'errore minima è pienamente rispettata, dato che ci si aspetta una P_e minore o uguale a 10^{-5}.

1.4.2 *Downlink*

E’ questo collegamento che determina le vere prestazioni del sistema. Questa tratta è la più critica, essendo quella dal satellite alla terra, proprio per l’impossibilità di valutare il comportamento del satellite una volta nell’atmosfera, sperando non subisca danneggiamenti nella fase d’espulsione e nel conseguente cammino. Oltre a quelli già descritti nell’ *uplink* ci sono ulteriori fattori da prendere in considerazione in questa fase del progetto. Dato che è presente un amplificatore di potenza in uscita dal nostro ricetrasmettitore, per amplificare i segnali che dovranno raggiungere la terra, è necessario prendere in considerazione anche questo ulteriore stadio nel calcolo del *downlink*. Dunque i fattori che devono venire considerati sono:

- Amplificazione di potenza in uscita all’integrato CC1100 che funge da ricetrasmettitore nel satellite.
- L’attenuazione della linea di trasmissione è trascurabile dato che la linea collegante il CC1100 e il PA è molto breve.
- Temperatura di sistema
- Margini di sistema, è la differenza tra rapporto segnale rumore ottenibile dal sistema e quello necessario per avere una determinata probabilità d’errore, naturalmente questa differenza deve essere positiva.

Nella pagina successiva, in tabella 3, troviamo nel dettaglio i valori del *downlink* relativi al nostro progetto.

NOME	SIMBOLO	UNITÀ	SORGENTE	VALORE
Potenza trasmessa	P	dBm	parametro	10
Potenza trasmessa	P	dBW	10log(P)	-20
Guadagno PA	G_{pa}	dB	parametro	20
Attenuazione linea	L_l	dB	parametro	0
Guadagno massimo antenna tx	G_{pt}	dB	parametro	2
Ampiezza cono a -3 dB	θ	°	parametro	78
Offset di puntamento	e	°	parametro	15
Perdita di puntamento	L_{pt}	dB	Eq. (4.7)	-0,44
Guadagno effettivo antenna tx	G_t	dB	Eq. (4.14)	1,56
EIRP	$EIRP$	dBW	Eq. (4.3)	1,56
Attenuazione di spazio libero	L_s	dB	Tabella 4.4	-148,2
Perdite di propagazione	L_a	dB	parametro	0
Guadagno massimo antenna rx	G_{pr}	dB	Tabella4.7	15
Ampiezza cono a -3 dB	θ	°	parametro	15
Offset di puntamento	e	°	10% di θ	1,5
Perdita di puntamento	L_{pr}	dB	Eq. (4.7)	-0,12
Guadagno effettivo antenna rx	G_r	dB	Eq. (4.14)	14,88
Temperatura di sistema	T_s	K	Eq. (4.13)	7499,41
Perdita di implementazione (3)	—	dB	stima	-2
Data Rate	R	bit/s	parametro	9600
Banda del ricevitore	B	kHz	datasheet	10
E_b/N_0 (1)	E_b/N_0	dB	Eq. (4.2)	18,26
Carrier to Noise density ratio	C/N_0	dB · Hz	Eq. (4.15)	58,09
Received Isotropic Power	RIP	dBW	$EIRP - L_s$	-146,64
Portante	C	dBW	Eq. (4.16)	-121,76
Portante	C	W		$6,66 \cdot 10^{-13}$
Ampiezza del segnale		µV	$V = \sqrt{P \cdot R}$	5,77
BER richiesta	BER	—	parametro	10^{-5}
E_b/N_0 richiesto (2)	Req E_b/N_0	dB		13,3
Margine di sistema		dB	(1)-(2)+(3)	2,96

Tabella 3: ***Downlink***

In figura 4 troviamo il grafico di come varia la *bit rate* in funzione dell'angolo di elevazione del satellite.

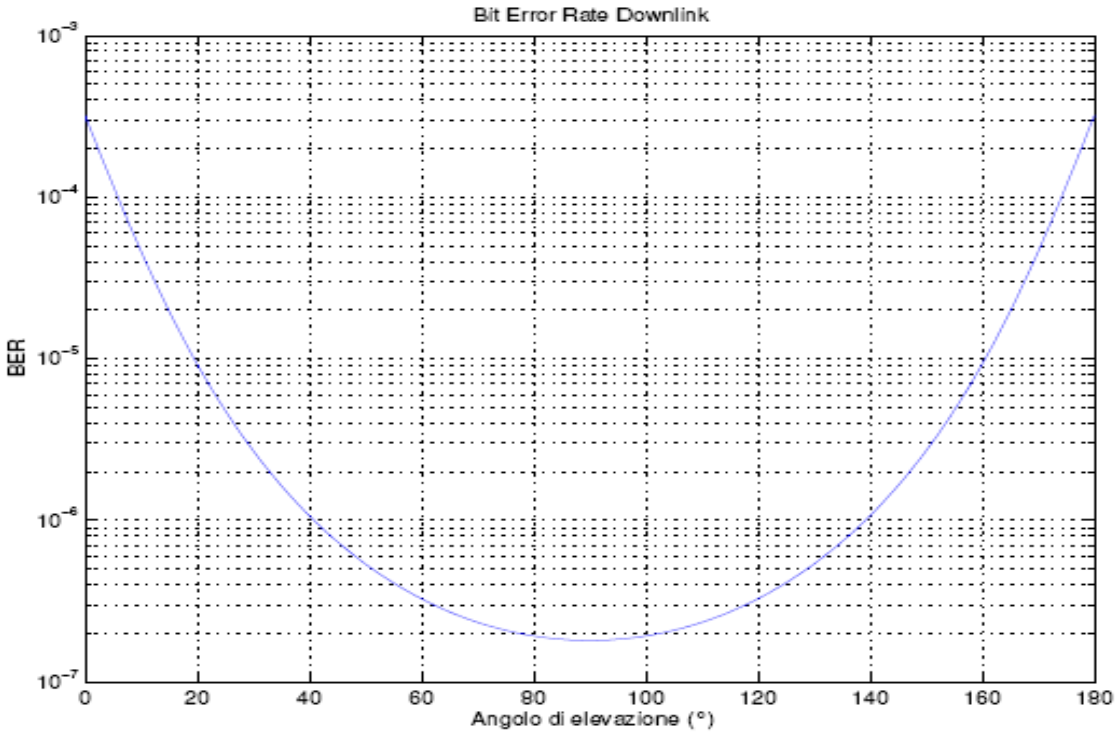

Figura 4: *Bit Rate Downlink* vs. angolo di elevazione

Anche in *downlink* è possibile calcolare la P_e in funzione del $\frac{E_b}{N_0}$. Come si legge dalla tabella 4, il rapporto $\frac{E_b}{N_0}$ in *downlink* , è pari a 17,52 dB, pertanto la probabilità d'errore, considerate rispettate le condizioni d'ortogonalità di questa modulazione, sarà:

$$P_e = Q\sqrt{\frac{E_b}{N_0}}$$
$$= \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{2N_0}}\right)$$
$$= \frac{1}{2} erfc\left(\sqrt{\frac{17{,}52}{2}}\right)$$
$$= 1{,}42 \bullet 10^{-5}$$

in linea con le specifiche di sistema che richiedono una probabilità d'errore uguale o inferiore a 10^{-5}. .

2. Specifiche di progetto

2.1 *Gli obiettivi*

L'obiettivo che ci siamo posti è progettare un ricetrasmettitore, da installare nel satellite, che soddisfi i requisiti di sistema, che in linea generale si possono riassumere in:

- Limitati consumi d'energia
- Lavori alle frequenze richieste di 437 MHz
- Alta sensibilità, cioè la capacità di ricevere segnali deboli
- Non necessiti di particolari componenti aggiuntive
- Possibilmente supporti la modulazione della stazione di terra
- Sia una componente *low-cost*

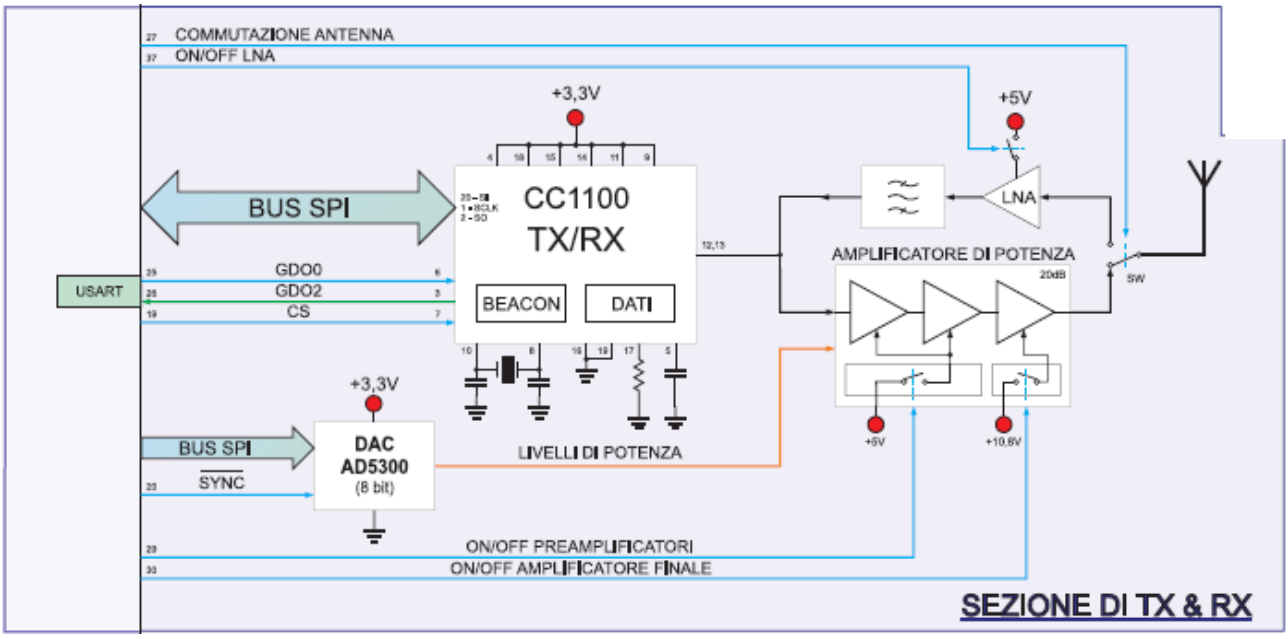

Figura 5: sezione Tx e Rx del satellite

L'integrato è connesso ad una PA (*power amplifier*, amplificatore di potenza) e ad un LNA (*low noise amplifier*, amplificatore a basso rumore) i quali sono selezionati dal microcontrollore tramite uno *switch*,collegato ad un'antenna, in base al fatto che si sia, rispettivamente, in modalità di trasmissione o ricezione. Nello schema a blocchi della sezione di Tx/Rx è stato preso in considerazione l'integrato CC1100 della Chipcon, le motivazioni le vedremo in seguito; l'integrato ha la funzio-

ne di trasmettere i dati di *beacon*, che forniscono delle indicazioni alla stazione di terra riguardo la posizione del satellite, il nome del satellite e altre informazioni utili,inoltre trasmette i dati raccolti nell'atmosfera, e allo stesso tempo deve essere in grado anche di ricevere i segnali dalla terra, ed elaborarli per essere in grado di segnalare quando sarà pronto alla trasmissione. L'obiettivo del progettista del ricetrasmettitore è quello di testare le prestazioni dell'integrato scelto nelle situazioni che potrebbero verificarsi nell'atmosfera, dalle più generali alle più critiche, raccogliere delle misure indicative del comportamento del ricetrasmettitore e fornire i mezzi per impostare la configurazione che più valorizzi le prestazioni a cui siamo interessati.

2.2 *Dati di Progetto*

Il primo obiettivo è quello di scegliere una componente che soddisfi le caratteristiche elencate nel paragrafo 2.1., fatto questo è necessario stilare un elenco degli argomenti principali riguardanti il ricetrasmettitore. Basandosi sul *link budget* e sui lavori fin qua svolti dagli studenti partecipanti al progetto, gli ambiti da sviluppare sono:

- LA TRASMISSIONE
- LA RICEZIONE
- IL CONTROLLO DELL'INTEGRATO
- L'ALIMENTAZIONE

Il compito del progettista è considerare e rispettare le specifiche introdotte dai singoli ambiti, facendo in modo che esse siano compatibili col tipo di sistema su cui si sta lavorando. In pratica, l'obiettivo è configurare l'integrato in termini di ricezione, trasmissione, controllo e alimentazione in modo tale che esso sia compatibile col sistema Atmocube.

2.2.1 *Parte di trasmissione*

In modalità di trasmissione la potenza uscente dal nostro integrato ,secondo il *link budget* [8], è di circa 10dBm; successivamente questo segnale passerà nel PA, dove subirà un aumento di potenza, fino a raggiungere almeno 1 W (pari a 30dBm). Il consumo, variabile in base alla *bit rate* adottata,

può andare da 13, mA a 28,9 mA. Le modulazioni considerate , come visto nel paragrafo1.3 , sono la BFSK, modulazione binaria di frequenza in banda traslata, oppure la BPSK, modulazione binaria di fase in banda traslata. Vedremo successivamente che, in base all'integrato scelto, la modulazione che ci interessa sarà quella in frequenza. Considerando che si utilizza la banda UHF è fondamentale che il nostro integrato lavori alla frequenza di 437 MHz. Nei capitoli relativi al *downlink* e al *uplink* è stata calcolata la probabilità d'errore relativa alla modulazione BFSK che, si vedrà in seguito, sarà la modulazione prescelta.

2.2.2 *Parte di ricezione*

Il satellite, una volta nell'atmosfera, deve essere in grado di ricevere i segnali che dalla terra gli vengono inviati. Questa operazione è utile per sapere quando il Cubesat è in grado di trasmettere i dati. Il consumo stimato in fase di ricezione va da 14,3 mA a 16,5 mA, lavorando a 433 MHz. In fase di progettazione del sistema [8] è stata scelta come BER massima accettabile 10^{-5} e come rapporto segnale\rumore richiesto $\approx$ 13,3 dB. Dato che il segnale emesso dalla stazione a terra ha una potenza di 40 W, si stima che il segnale trasmesso da terra verso il satellite sia $\approx$ 28,2 dB; prendendo in considerazione le varie attenuazioni presenti si stabilisce che il segnale si assesti all'ingresso del ricevitore a $\approx$ - 88,44 dB. Detto questo il ricetrasmettitore in dotazione al Cubesat deve avere una sensibilità $\leq$ - 88,44 dB.
Sensibilità e probabilità d'errore sono strettamente collegate; il nostro obiettivo è mantenere una BER di 10^{-5} utilizzando una bit rate a 9600 kbps (sfruttando così le frequenze radio amatoriali). Un altro obbiettivo del progettista, se è possibile, è implementare nel ricetrasmettitore un filtraggio del rumore o una codifica dei dati in modo da limitare l'*ISI* (interferenza intersimbolica, tipico dei disturbo presente nei canali a banda limitata).

2.2.3 *Parte di controllo*

Il satellite è dotato di un μC, è lui che determina quali operazioni ogni singolo sottosistema deve effettuare; lo studio di questo componente è stato sostenuto dallo studente Montagnana. Si è arrivati alla conclusione che è necessario l'uso di due differenti microcontrollori, il principale, chiamato *master* ha la responsabilità del controllo del satellite, nonché della trasmissione a terra dei dati, mentre il secondo chiamato *slave* ha il compito di prelevare i dati strumentali. La scelta è caduta su

due prodotti della Microchip, il microcontrollore P18F452 per il *master* mentre il P18F252 è lo *slave* ; il nostro integrato comunica col *master* tramite il *bus* SPI, il quale è ovviamente condiviso con altri sottosistemi, per cui sono state impostate delle regole affinché non si verifichino collisioni. Sarà il nostro microcontrollore a costruire i pacchetti secondo l' AX.25, che è il protocollo scelto, visto che è usato frequentemente per le trasmissioni radio amatoriali, per la comunicazione a livello *data link* . Il μC è in grado di settare l'integrato in modalità ricezione o trasmissione tramite i pin che sono dedicati a questa funzione, di conseguenza dovrà anche regolare lo *switch* che ha il compito di fornire il segnale ricevuto all'integrato in modalità ricezione e fornire il segnale da trasmettere all'antenna in modalità trasmissione. Quindi il μC provvederà a configurare prontamente il nostro integrato a seconda che sia necessario ricevere o trasmettere. Pertanto un ulteriore obiettivo del progettista del ricetrasmettitore satellitare è definire i comandi, e per mezzo di quali linee, l'integrato comunichi col μC.

2.2.4 *Parte di alimentazione*

Considerando lo studio fatto precedentemente dallo studente Caharija[9], si deve considerare il ricetrasmettitore nella sua interezza, studiando i vari sistemi che lo compongono e che dovranno essere alimentati, i quali sono:

- Amplificatore di potenza PA
- Amplificatore a basso rumore LNA
- Il ricetrasmettitore
- Commutatore d'antenna

L'amplificatore di potenza del trasmettitore (PA) è un amplificatore con stadio finale funzionante in classe C; il suo compito è portare a 2-3 W la potenza del segnale ad RF. Il rendimento è stimato attorno al 40-50% . L'alimentazione richiesta è di 5V. Non avendo avuto dati sufficienti al momento dello svolgimento del lavoro di progettazione dell'alimentatore si è deciso di considerare la potenza massima richiesta dall'amplificatore ad 8 W (condizione peggiore).
L'amplificatore a basso rumore (LNA) è il primo stadio della ricezione; esso è in fase di sviluppo ma si può affermare con certezza che non consumerà più di 20-30 mA. Anche questo dispositivo è alimentato con una tensione da 5V. La potenza massima richiesta , nel peggiore dei casi, si aggira attorno ai 200 mW.

Per quanto riguarda il ricetrasmettitore , da considerarsi dotato di oscillatore interno, è necessaria una tensione di alimentazione molto stabile. La tensione va dà 1,8 – 3,6 V, con una I_{max} che oscilla tra i 28 mA e i 30 mA , dati considerati lavorando alla frequenza di 437 MHz.

Per quanto riguarda il commutatore d'antenna non c'è ancora una soluzione definitiva; le opzioni considerate sono : commutatore RF a diodi pin, *relè*, *switch* integrati, ecc. Comunque sia l'alimentazione è sempre di 5 V; nel caso si scelgano diodi pin si dovrà disporre di una tensione negativa, in ogni modo il dispositivo non consumerà più di 100 mW.

Nella figura 6 troviamo lo schema generale dell'alimentazione del sistema del ricetrasmettitore; per convenienza si è scelto l'integrato CC1100 della Chipcon, che procedendo con l'elaborato, risulterà esser la componente prescelta.

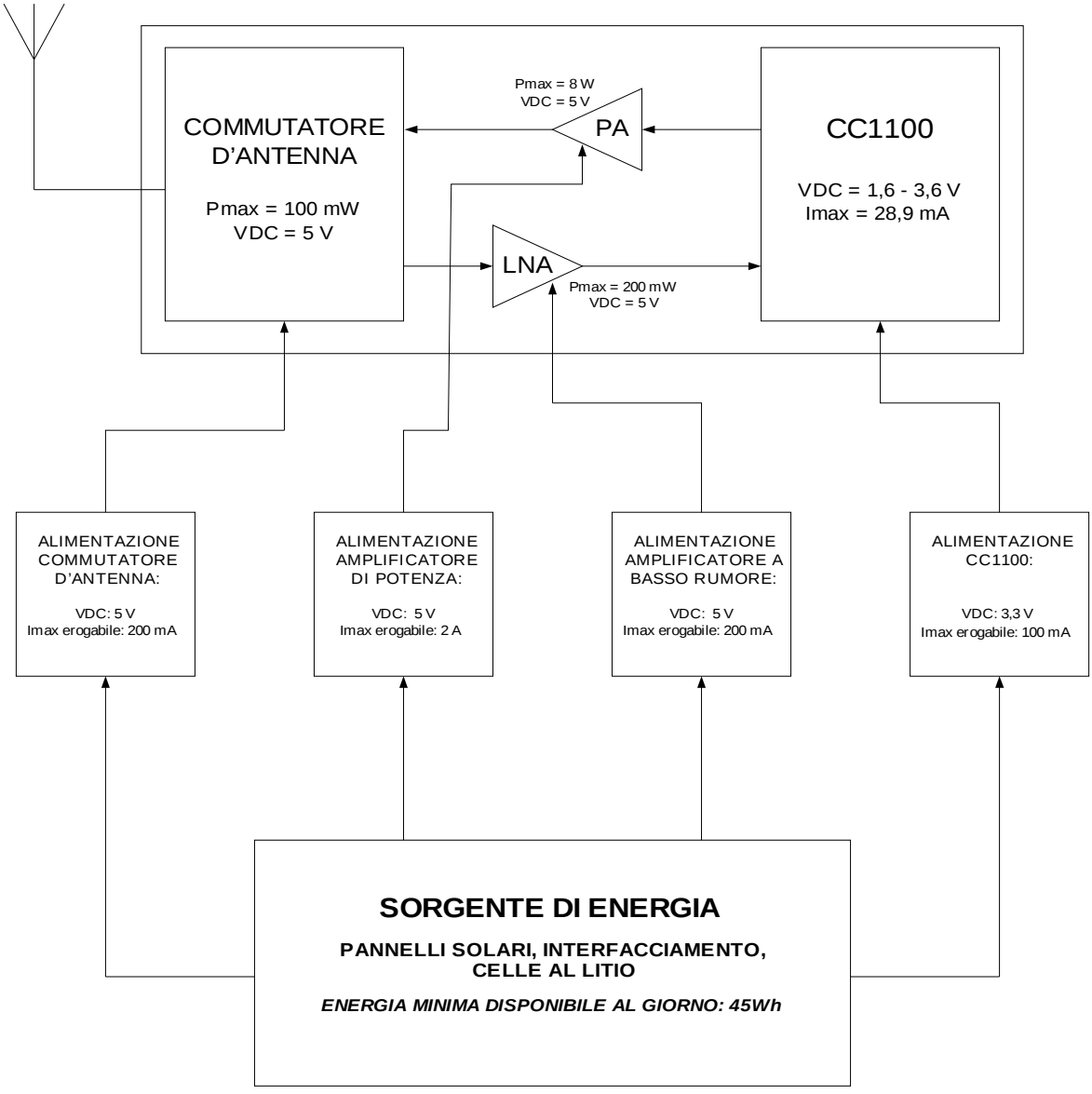

Figura 6: sistema di alimentazione del ricetrasmettitore

2.3 *Scelte*

Una volta determinate le caratteristiche che il ricetrasmettitore deve avere è compito del progettista trovare una componente che svolga le funzioni utili e soddisfi i requisiti di sistema. La scelta è caduta su due componenti, l'ADF7020 della Analog Device e sul CC1100 della Chipcon. Inizialmente s'era discusso se adottare due componenti separate per la ricezione e per la trasmissione oppure un singolo pezzo che svolgesse entrambe le funzioni; per questioni di spazio, costi e maggiore facilità d'implementazione s'è optato per la seconda soluzione. Infatti i nostri due integrati appartengono alla famiglia dei semiconduttori *transceiver*, abili a svolgere sia trasmissione che ricezione in un unico *chip*. Altro punto essenziale della ricerca è stato il basso consumo, ed infatti i nostri due integrati consumano poco, l'ADF7020 per +10 dBm di potenza in uscita consuma 28,5 mA che può scendere a 26,8 mA nel caso si adatti separatamente il PA interno all'integrato con uno *switch* esterno, mentre in ricezione consuma dai 19 mA ai 21 mA. L'integrato della Chipcon per trasmettere a +10 dBm consuma circa 28,9 mA in trasmissione mentre in ricezione oscilla, a seconda della *bit rate*, tra i 14,3 mA e i 16,5 mA. Altro fattore fondamentale è la banda di lavoro che, ricordiamolo, è di 437 MHz; entrambi la soddisfano avendo l'ADF7020 un *range* utile che è 431-478 MHz, mentre per il CC1100 sarà 400-464 MHz, ed è da aggiungere che quest'ultimo è dotato di un filtro digitale di canale, programmabile, che spazia da una frequenza di 58 KHz a 812 KHz (valori ottenuti considerando un cristallo che lavora a 26 MHz, essendo i valori del filtro dipendenti dal cristallo. Altra caratteristica presa in considerazione e di vitale importanza è la sensibilità dell'integrato che deve essere minore o uguale a -84,4 dBm, ed entrambi gli integrati soddisfano perfettamente questo vincolo. Un occhio di riguardo è necessario dedicarlo al tipo di modulazione supportata; si è riscontrata una certa difficoltà a trovare in commercio integrati supportanti la modulazione binaria di fase, di fatto ci si è posti come obiettivo la ricerca di una modulazione la più comune possibile per poi poter eventualmente progettare una semplice stazione di ricezione a terra; a questo proposito vediamo che il CC1100 supporta la FSK (*Frequency Shift Keying*) e la GFSK (*Gaussian- Frequency Shift Keying*), differente dalla precedente per il fatto che gli impulsi utilizzati dalla modulazione sono funzioni gaussiane il che migliora l'occupazione di banda del segnale, mentre l' ADF7020 supporta anch'esso la FSK e la GFSK con l'aggiunta della modulazione ASK (*Amplitude Shift Keying*, modulazione d'ampiezza in banda traslata). Una possibilità da non trascurare nella scelta degli integrati è la possibilità di introdurre delle codifiche di canale per eliminare l'*ISI* , cioè l'interferenza intersimbolica, o la possibilità di sfruttare qualche filtraggio per limitare il più possibile il rumore che il canale

introduce. Ultimo punto, ma non meno importante, è verificare la possibilità d'approvvigionamento dei pezzi; essendo un progetto *low cost* più si limitano le spese meglio è. A questo riguardo s'è approfittato delle numerose conoscenze del professore Fragiacomo con le aziende fornitrici per verificare la disponibilità e i costi dei pezzi.

2.3.1 *ADF7020*

L'ADF7020 è un integrato prodotto dalla Analog Device è una componente a bassi consumi che può lavorare a due differenti bande (431- 478 MHz \ 862 – 956 MHz). Ha la possibilità di modulare in diverse modalità che sono FSK, GFSK, ASK, OOK (On-Off Keying), GOOK (Gaussian *On-Off Keying*), ed è possibile programmare diverse *bit rate* che per la FSK\GFSK può raggiungere i 200 kbps. La tensione necessaria per alimentare l'integrato soddisfa le richieste e si assesta nell'intervallo 2,3 – 3,6 V. La potenza d'uscita è programmabile e và dai – 16 dBm a + 13 dBm con passi di 0,3 dBm. La sensibilità in ricezione varia a seconda della modulazione e della *bit rate* , e si può vedere che per 9,6 kbps ci si aggira intorno ai –122 dBm. I consumi, come già anticipato sopra, sono esigui, circa 19 mA in ricezione e circa 26,8 mA in trasmissione (con + 10 dBm di potenza in uscita). C'è la possibilità di controllare visivamente in maniera digitale il livello dell' *RSSI* (è un indicatore del livello di potenza del segnale presente nel canale utilizzato per la comunicazione)

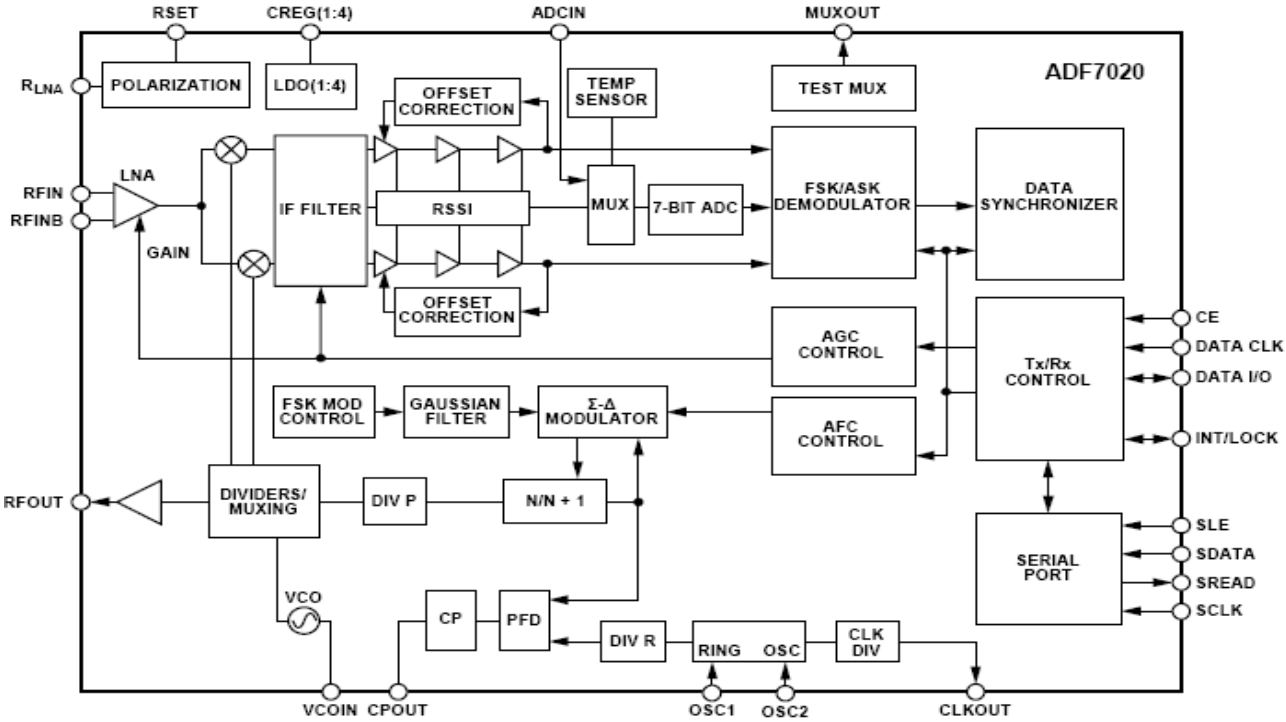

Figura 7: struttura interna del ADF7020

Osservando lo schema a blocchi interno , figura 7, si possono individuare le parti dedicate alla modulazione e alla demodulazione che, come precedentemente detto , possono essere la FSK o la ASK. Vediamo che sia in fase di demodulazione che in fase di demodulazione ci sono dei filtraggi di prassi nell'integrato. In fase di demodulazione troviamo un filtro IF (a frequenze intermedie, per l'appunto quelle ricavate dal segnale ricevuto dopo il suo passaggio dal *mixer*) che è utile per filtrare il rumore accumulato dal segnale lungo il passaggio nel canale; in fase di modulazione potremmo utilizzare un filtro gaussiano, ottenendo così una GFSK, che ci permetterebbe di risparmiare un po'di banda. Una particolarità di questo integrato, che và a suo favore, è la presenza di due blocchi di controllo delle prestazioni, l'AGC *control* e l'AFC *control* ; il primo blocco è collegato all'LNA, ed è usato come blocco di controllo del guadagno. Se l'*RSSI*, che è il valore della potenza stimata nel canale d'interesse, si avvicina alla soglia di controllo dell'AGC lo stesso riduce il guadagno e viceversa se l' *RSSI* è troppo basso rispetto la soglia. Il secondo blocco, ovvero l'AFC *control* , è utile visto che controlla e rimuove gli errori in frequenza che possono nascere da fraintendimenti tra l'oscillatore in trasmissione e l'oscillatore in ricezione. Il blocco denominato MUX controlla i regolatori di tensione dato che nell'integrato ci sono quattro dispositivi che alimentano delle specifiche parti del circuito integrato; per l'appunto questo dispositivo controlla che le tensioni d'alimentazione rispettino le specifiche e ad ogni sezione venga fornita la giusta tensione. Il blocco denominato *serial port* si occupa della comunicazione col µC, e servono ad immagazzinare le configurazioni dei progettisti.
Nella pagina successiva, figura 8, vediamo lo schema a blocchi completo del ricetrasmettitore utilizzando l'ADF7020.

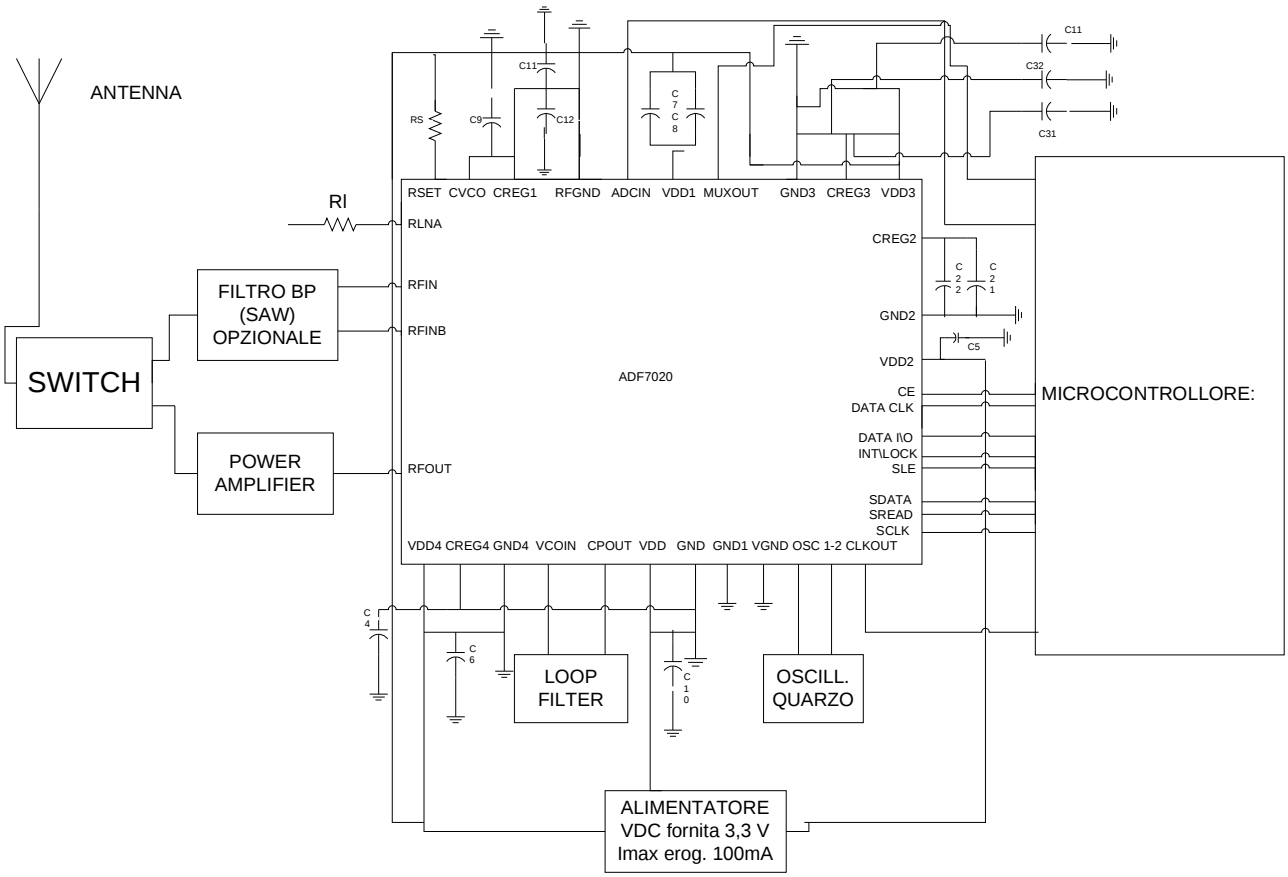

Figura 8 : schema a blocchi completo del ricetrasmettitore utilizzando l'ADF7020

Componente	Valore
C11	100 nF
C12	5,1 pF
C21	100 nF
C22	5,1 pF
C31	100 nF
C32	5,1 pF
C4	100 nF
C5	10 nF
C6	10 nF
C7	1 µF
C8	10 pF
C9	22 nF
C10	0,01 µF
C11	0,01 µF
RI	1,1 Ω
Rs	3,6 Ω

Tabella 4: valori componenti esterni ADF7020

Nel *data sheet* dell'ADF7020 [6] sono richiesti una serie di induttori, capacità e resistenze in prossimità di alcuni pin dell'integrato, per garantire le migliori prestazioni d'ogni blocco funzionale. E'in-

teressante considerare una possibile applicazione dell'integrato; l'ADF7020 ha ottime *performance* in termini di sensibilità, potenza trasmessa e consumo di corrente solamente se le porte RF d'ingresso e d'uscita sono adattati all'impedenza dell'antenna. L'Analog Device fornisce due soluzioni, un adattamento con una rete esterna oppure sfruttando le componenti dell'integrato senza troppi pezzi da aggiungere, e quindi limitando le spese c'è la possibilità di progettare una rete d'adattamento interna.

	ADF 7020
Sensibilità	≈ -101dBm per 9.6 kbps
Consumi	28,5 mA in trasmissione; 19- 21 mA in ricezione in base alla modalità configurata.
Potenza massima	+ 10 dBm
Tensione d'alimentazione	2,3 – 3,6 Volt
Modulazione	FSK, GFSK, ASK, OOK, GOOK
Bit Rate possibili	da 0,15 kbps a 200 kbps con FSK
Frequenze di lavoro	da 431 MHz a 478 MHz; da 862 MHz a 956 MHz

Tabella 5: caratteristiche ADF7020

In conclusione si può dire che l'ADF7020 soddisfi pienamente le nostre esigenze essendo facilmente implementabile, soddisfando tutti i parametri di progetto e garantendo buone prestazioni; unica nota è che non ci sono particolari funzioni per perfezionare la qualità di trasmissione andando quindi a migliorare la BER, non è presente infatti alcuna codifica di canale.

2.3.2 *CC1100*

Il CC1100 è un *transceiver* della Chipcon d'ultima generazione a basso costo e a bassa potenza; è adoperato ed è stato progettato per ISM (*Industrial, Scientific and Medical*). Le bande di frequenza a cui può essere configurato sono 315 MHz, 433 MHz, 868 MHz e 915 MHz, ma è possibile programmare anche ad altre frequenze nelle bande 300-348 MHz, 400-464 MHz e 800-928 MHz. Il nostro integrato è molto piccolo, appena 4 mm per 4 mm, con soli 20 *pins*. Ha un'alta sensibilità, arrivando a – 110 dBm con una *bit rate* di 1,2 kbps e solo 1% d'errore sui pacchetti ricevuti. In trasmissione, qualsiasi sia la modulazione, può avere una potenza in uscita di + 10 dBm. I consumi sono ridotti in ricezione, circa 15,4 mA alla frequenza di lavoro di 433 MHz con 1,2 kbps di *bit*

rate. Questa componente ha integrato un *modem* in banda base estremamente configurabile. Supporta le modulazioni FSK, GFSK, MSK, OOK e ASK ed è configurabile ad una *data rate* che può raggiungere i 500 kbps. Le componenti richieste esternamente sono davvero poche ; ci sono due configurazioni possibili a seconda che il CC1100 lavori a 315\433 MHz oppure a 868\915 MHz. Anche il CC1100, come l'ADF7020, è dotato del blocco *AGC control*, utile al controllo del guadagno dell'LNA interno all'integrato. Questa componente ha diverse possibilità per migliorare le prestazioni; si può abilitare l'opzione *Forward Error Correction* con *Interleaving*, utile ad evitare gli errori di *burst*, cioè quel tipo d'errore che colpisce in una determinata sequenza di informazione diversi bit consecutivi. Inoltre riscontriamo anche in questo integrato il blocco AFC che può essere usato per allineare la frequenza del sintetizzatore alla frequenza centrale del segnale ricevuto; è possibile introdurre una codifica di canale (*whitening*) ma anche la funzione PQI (*Preamble Quality Indicator*) che serve a rilevare i preamboli e proteggere la sincronizzazione delle parole dal rumore. Inoltre supporta la tecnologia CCA (*Clear Channel Assessment*), che consiste nell'"ascolto" del canale prima di trasmettere le informazioni per assodare che esso non sia già occupato (*listen-before-talking*).

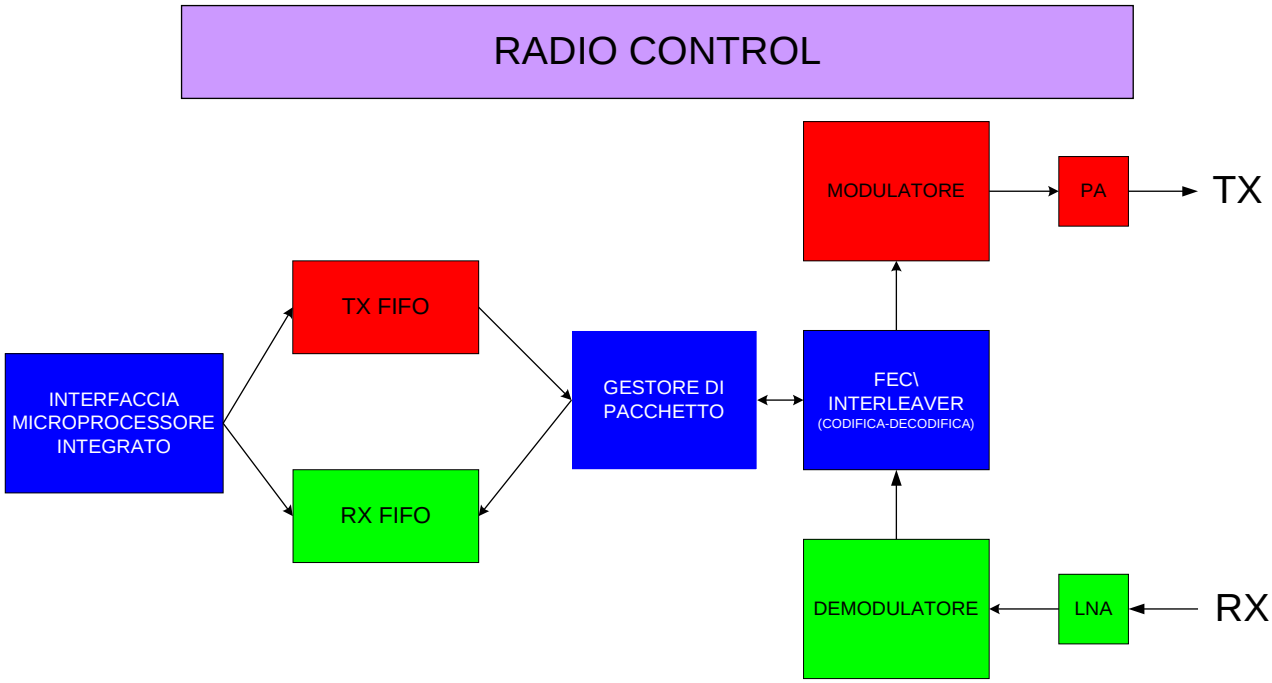

Figura 9: principali blocchi del CC1100

Facendo riferimento allo schema a blocchi di figura 9 osserviamo a livello generale i principali blocchi funzionali del CC1100. In ricezione il segnale passa nell'LNA interno, l'amplificatore a basso rumore di cui è dotato l'integrato, alle frequenze RF, seguitamente verrà convertito alle frequenze IF per esser poi introdotto in un convertitore analogico\digitale prima di entrare nel blocco

del demodulatore. In fase di trasmissione invece il segnale modulato viene direttamente sintetizzato alle RF, per poi passare nel PA interno, amplificatore di potenza, dell'integrato. Connessi ai blocchi del demodulatore e del modulatore troviamo i blocchi del *FEC/INTERLEAVER* che introduce la codifica di canale (opzionale) e il *PACKET HANDLER*, che svolge la funzione di gestore dei pacchetti di comunicazione, di cui approfondiremo successivamente le caratteristiche. Individuiamo anche altri due blocchi, uno dedicato alla ricezione dal nome RX FIFO e uno dedicato alla trasmissione TX FIFO; entrambi sono FIFO (*first-in-first-out*, filtro digitale) con 64-byte di memoria per i dati. L'integrato è controllato digitalmente da un microcontrollore tramite un'interfaccia digitale SPI a 4 fili. Altro blocco è il *RADIO CONTROL*, che è una macchina a stati integrata nel nostro dispositivo, che ha la funzione di configurare di volta in volta, in base alle esigenze del progetto, il nostro integrato nelle varie modalità possibili.

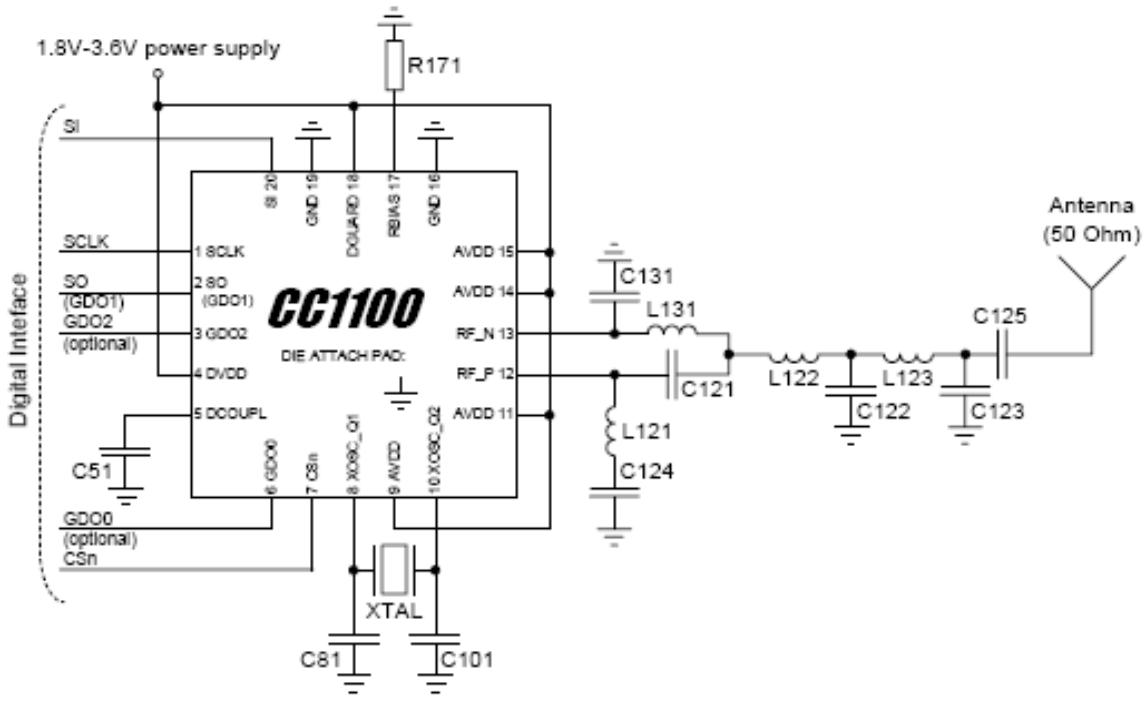

Figura 10 : configurazione consigliata per CC1100 a 433 MHz

In figura 10 abbiamo la configurazione consigliata per le componenti esterne al nostro CC1100. Si può vedere che le componenti, per l'integrato che lavora a 433 MHz sono poche, economiche e di facile reperibilità. E' importante specificare che le capacità di disaccoppiamento , utili a bloccare la corrente continua fonte di interferenze sui segnali, non sono presenti nello schema; la progettazione e il dimensionamento del circuito definitivo sarà approfondito in fase di progettazione, nel capitolo successivo.

	CC1100
Sensibilità	≈ -111dBm per 1.2 kbps
Consumi	28,9 mA in trasmissione; 14.3-16.5mA in ricezione in base alla bit rate programmata.
Potenza massima	+ 10 dBm
Tensione d'alimentazione	1,8 – 3,6 Volt
Modulazione	FSK, GFSK, ASK, OOK, MSK
Bit Rate possibili	da 1,2 kbps a 500 kbps con FSK
Frequenze di lavoro	da 300 MHz a 348 MHz; da 400 MHz a 464 MHz, da 800 MHz a 928 MHz

Tabella 6: caratteristiche CC1100

(i valori nella tabella non sono omogenei rispetto alla tabella dell'ADF7020, perché non abbiamo trovato i dati della sensibilità alla *bit rate* e modulazione a noi utile). E'interessante vedere più approfonditamente gli aspetti peculiari di questa componente:

1. *RECEIVER CHANNEL FILTER BANDWIDTH*
2. *DATA RATE PROGRAMMING*
3. *PACKET HANDLING*
4. *RECEIVED SIGNAL QUALIFIER E LINK QUALITY INFORMATION*
5. *FEC CON INTERLEAVING*

1. In ricezione è possibile usufruire di un filtro programmabile, configurando I registri MDMCFG4.CHANBW_E e MDMCFG4.CHANBW_M in maniera opportuna secondo la tabella presente nel *data sheet* [tabella 20 pagina 39], in modo che il filtro riduca la larghezza del canale in ricezione. La larghezza di banda del filtro è dipendente e proporzionale alla frequenza degli oscillatori. E' preferibile che il segnale occupi una banda ≤ all'80% della larghezza di banda del filtro di canale, in questa maniera sarà possibile avere le prestazioni migliori.

2. Ci sono due registri che ci permettono di programmare la *data rate* voluta, i quali sono MDMCFG3.DRATE_M e MDMCFG4.DRATE_M. E' possibile considerare la *data rate* dipendente dai valori assegnati a questi registri grazie alla formula

$$R_{DATA} = \frac{\left(256 + DRATE_M\right) \cdot 2^{DRATE_E}}{2^{28}} \cdot f_{XOSC}$$

Da questa considerazione posso trovare, data una certa *data rate* i valori assegnabili ai due registri:

$$DRATE_E = \ln\left(\frac{R_{DATA}\cdot 2^{20}}{f_{XOSC}}\right)$$

$$DRATE_M = \frac{R_{DATA}\cdot 2^{28}}{f_{XOSC}\cdot 2^{DRATE_E}} - 256$$

L'integrato supporta data rate che vanno da 1,2 kbps fino a 500 kbps.

3. Il PACKET HANDLING è un supporto hardware per la gestione dell'orientazione di pacchetto dei protocolli radio presente nel nostro integrato, esso svolge una funzione importantissima; lo si può sfruttare sia in trasmissione che in fase di ricezione. Per quanto riguarda la fase trasmissiva :
    - Si possono programmare un certo numero di *byte* di preambolo
    - Può essere aggiunta una parola di 2 *byte* per la sincronizzazione (che possono essere duplicati per ottenere una parola di sincronizzazione di 4-*byte*)
    - E' possibile utilizzare la funzione *whitening* , che introduce dei *bit* di ridondanza in trasmissione per proteggere le informazioni utili dalle impurità di canale, *bit* che verranno eliminati poi in ricezione
    - Opzionalmente si può sfruttare il FEC (*Forward Error Correction*) con *Interleaving*, che è una codifica di canale sfruttando un codice convoluzionale
    - Opzionalmente si può sfruttare il CRC, che è un codice a controllo di parità
    - Controlla e gestisce la configurazione a 4-*byte* del preambolo e la parola di sincronizzazione di 4-*byte* di lunghezza

    Il *PACKET HANDLER* può aggiungere in trasmissione tutti gli elementi sopraccitati nei pacchetti immagazzinati nel TX FIFO. Allo stesso modo, prendendo i pacchetti dal RX FIFO in fase di ricezione, li destrutturerà considerando:
    - La rivelazione del preambolo
    - Rivelazione della parola di sincronismo
    - Opzionalmente controllerà un *byte* dedicato all'indirizzo di pacchetto
    - Opzionalmente calcola e controlla il CRC

- Opzionalmente appende 2-*byte* di stato con il valore dell'RSSI , del *LINK QUALITY INDICATION* e lo stato CRC

I registri che controllano il *PACKET HANDLER* possono essere modificati solo quando l'integrato sarà in modalità *IDLE* (in modalità inattiva).

4. Per *RECEIVED SIGNAL QUALIFIERS* e *LINK QUALITY INFORMATION* intendiamo quella serie di controlli di qualità dell'informazione che il nostro integrato può supporta re, i quali hanno come scopo quello di aumentare la verosimiglianza che una parola di sincronizzazione valida sia rivelata, in altre parole che si riesca ad interpretare il messaggio ricevuto il più verosimilmente possibile rispetto l'originale. Le funzioni a nostra disposizione sono:
   - PQT, *Preamble Quality Thresold*: il CC1100 ha un contatore interno che, a seconda che si riceva un bit di preambolo uguale (o diverso) al bit precedente, incrementa (o decrementa) il valore del contatore. Se le soglie vengono rispettate la "parola" ricevuta e' sincronizzata.
   - RSSI: è possibile , tramite questa funzione, visualizzare il livello di potenza del segnale nel canale richiesto. In fase di ricezione è possibile leggere di continuo il livello RSSI dall'opportuno registro. Questo livello è in dB con una risoluzione di 1\2 dB
   - CS, *Carrier Sense*: anche questa opzione è utile a determinare la qualità della parola di sincronizzazione ma può essere anche usata contro le interferenze generate da altre sorgenti RF nella banda ISM
   - CCA, *Clear Channel Assesment*: è una funzione usata per segnalare se il canale interessato è libero o occupato.

5. FEC (*Forward Error Correction*) con *Interleaving*: è una codifica di canale atta a ridurre la BER quando l'integrato opera in prossimità dei limiti di sensibilità. Sfrutta una codifica convoluzionale, di cui si può calcolare il PER (*Packet Error Rate*), partendo dalla BER:

$$PER = 1 - (1 - BER)^{packet_lenght}$$

2.3.3 *Comparazione fra ADF 7020 e CC1100*

Dopo avere studiato le caratteristiche peculiari dei due integrati è venuto il momento di confrontarli per vagliare quale fosse più appropriato alle nostre esigenze; in tabella 7 riassumiamo le caratteristi-

che più generali, notando che i due integrati sono piuttosto simili sia per quanto si parla di consumi, sia per la potenza, come per l'alimentazione, le modulazioni, la bit rate e le frequenze di lavoro.

	CC1100	ADF 7020
Sensibilità	≈ -111dBm per 1.2 kbps	≈ -101dBm per 9.6 kbps
Consumi	28,9 mA in trasmissione; 14.3-16.5mA in ricezione in base alla bit rate programmata.	28,5 mA in trasmissione; 19- 21 mA in ricezione in base alla modalità configurata.
Potenza massima	+ 10 dBm	+ 10 dBm
Tensione d'alimentazione	1,8 – 3,6 Volt	2,3 – 3,6 Volt
Modulazione	FSK, GFSK, ASK, OOK, MSK	FSK, GFSK, ASK, OOK, GOOK
Bit Rate possibili	da 1,2 kbps a 500 kbps con FSK	da 0,15 kbps a 200 kbps con FSK
Frequenze di lavoro	da 300 MHz a 348 MHz; da 400 MHz a 464 MHz, da 800 MHz a 928 MHz	da 431 MHz a 478 MHz; da 862 MHz a 956 MHz

Tabella 7: CC1100 vs. ADF7020

Anche per quanto riguarda le applicazioni esterne da aggiungere si è notato che la differenza è assai minima necessitando entrambe di pochissimi componenti esterni (capacità, resistenze e induttanze). Una differenza degna di nota è che il CC1100 supporta varie opzione per migliorare la qualità della trasmissione mentre l'ADF7020 ne è sprovvisto. Entrambi gli integrati sono *low cost*; altra variabile da considerare è la possibilità d'approvvigionamento, e la possibilità di trovare componenti (vedi scheda madre e microcontrollore) e strumenti per testare l'integrato.

2.3.4 *Motivazione della scelta del CC1100*

Successivamente al confronto fra integrati, valutando tutte le voci, si può constatare che per quanto riguarda consumi, modulazioni e sensibilità, cioè le caratteristiche che per prime devono essere vagliate, i nostri due integrati sono pressoché equivalenti. In linea generale anche le componenti esterne richieste sono le stesse, cioè una serie di induttanze, condensatori e resistenze , più un cristallo esterno usato dall'oscillatore degli integrati, con l'eccezione dell'ADF7020 che richiede in aggiunta la progettazione di un *loop filter* ,che varia a seconda della modulazione adottata, il quale svolge la funzione di integrare gli impulsi di corrente forniti dal circuito *charge pump* dell'ADF7020, in modo da formare una tensione che sintonizzi l'uscita del VCO dell'integrato alla frequenza desiderata (inoltre attenua le interferenze spurie generate dal PLL). Una nota importante a favore del

CC1100 è la possibilità d'agire sui dati grazie alle tecniche di controllo della qualità dell'informazione e alla codifica di canale, che l'ADF7020 non supporta, pertanto l'integrato della Chipcon garantisce prestazioni migliori. Nella scelta degli integrati siamo stati fortunati, dato che la componente della Analog Device era facilmente ottenibile per i contatti di lavoro tra l'Enteos e la stessa; mentre per la componente della Chipcon la situazione è stata ancor più favorevole; una società amica dell'Enteos, azienda che ha supportato dall'inizio alla fine lo svolgimento di questo progetto, la Elcon, s'è offerta di comprare e fornirci l'integrato CC1100 in cambio dello studio sulle prestazioni e sul possibile uso della componente.

2.3.5 *Piano di lavoro*

Dopo avere scelto l'integrato da utilizzare è necessario testarlo e collegarlo alle componenti che il progettista del sistema ha stabilito. Successivamente sarà nostro dovere simulare le condizioni che il ricetrasmettitore dovrà affrontare nell'atmosfera e registrare il suo comportamento (con le dovute approssimazioni). Nei paragrafi successivi andremo quindi ad esplicare ognuna delle fasi del progetto, che sono:

- Schema a blocchi in dettaglio
- Progetto parte di alimentazione
- Progetto della parte di connessione verso l'amplificatore di potenza
- Progetto della parte di connessione verso l'LNA
- Calcolo delle prestazioni in ricezione e trasmissione
- Schema elettrico completo
- Stesura delle specifiche di controllo
- Realizzazione del prototipo
- Misure del prototipo

3. Il progetto

Andiamo ora ad analizzare nel particolare ogni singola fase di progettazione del ricetrasmettitore che verrà poi installato nel satellite ATMOCUBE.

3.1 *Schema a blocchi*

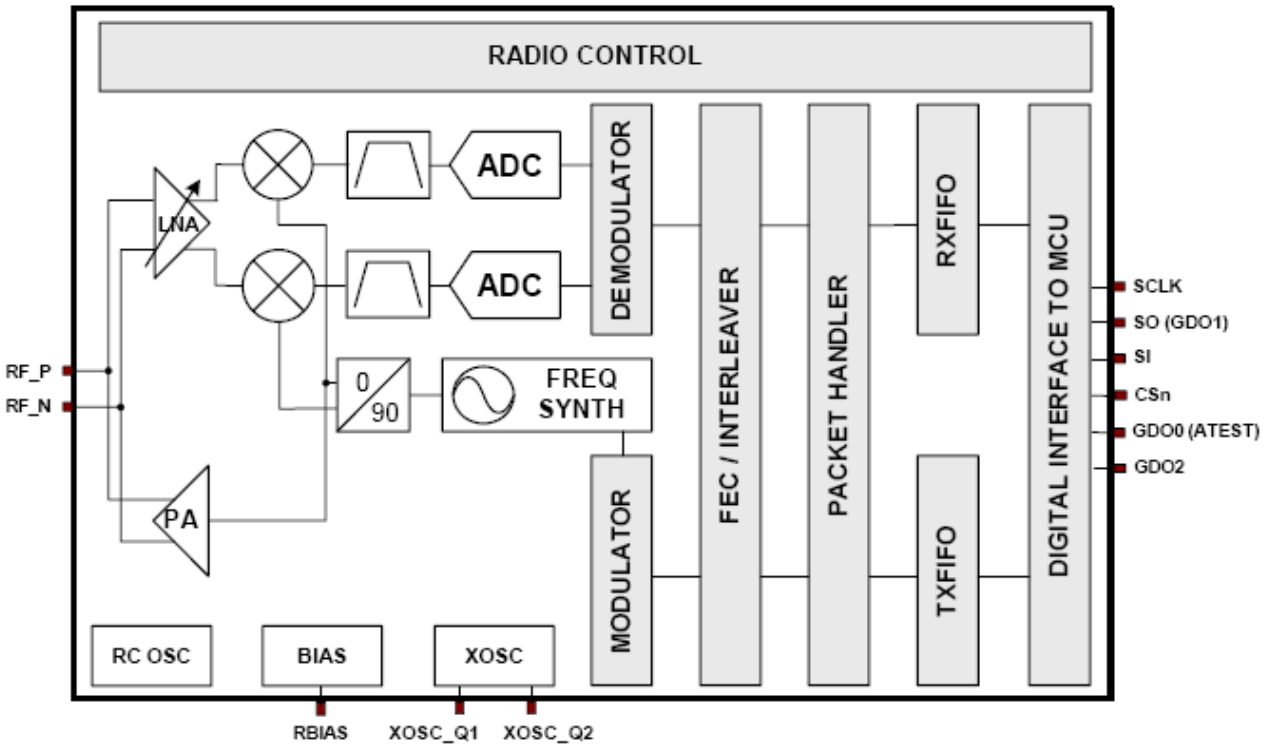

Figura 11: struttura interna del CC1100

Lo schema a blocchi di figura 11 mette in evidenza in maniera più esaustiva ,rispetto figura 9, quali siano e secondo che disposizione, i blocchi funzionali presenti nell'integrato.TX FIFO e RX FIFO rappresentano le memorie da 64-*byte*, che hanno come scopo quello di immagazzinare i dati , rispettivamente, in trasmissione e in ricezione; connesso a questi due blocchi individuiamo il *PACKET HANDLER* che ha il compito della gestione dei pacchetti, dall'aggiunta di bit al preambolo, alle varie altre applicazioni per migliorare le prestazioni lavorando sul pacchetto. Il blocco *FEC\INTERLEAVER* introduce la codifica di canale, programmabile, che sfrutta una codice convoluzionale ,cioè un codice con memoria dato che i blocchi di codice dell'informazione corrente sono determinati anche dai bit precedenti. Infine i blocchi dedicati alla modulazione, *MODULATOR*, e alla demodulazione, *DEMODULATOR*, con il blocco del sintetizzatore che produce localmente le frequen-

ze per demodulare e modulare, i quali blocchi sono connessi rispettivamente al LNA, posto dopo il pin di ricezione dei dati e dopo il blocco di demodulazione, e il PA posto dopo il blocco del modulatore e prima del pin d'uscita dei dati modulati.

Le linee da collegare all'integrato sono venti; quattro linee sono dedicate alla configurazione seriale che avviene tramite il microcontrollore (SO; SI; CSn, SCLK), due linee sono dedicate alla messa a terra dell'integrato (GND[16-19]), altre quattro linee sono dedicate all'alimentazione analogica dell'integrato , visto che ogni linea è dedicata ad una particolare zona del ricetrasmettitore(AVDD[9-11-14-15]). Due linee sono dedicate al collegamento con il cristallo esterno alla componente(XOSC_Q1,XOSC_Q2 [8-10]); tre linee dedicate all'alimentazione, di tipo digitale questa volta, di cui un dedicata alla sezione digitale I\O e al nucleo digitale dei regolatori di tensione dell'integrato (DVDD 4) uno dedicato al disaccoppiamento (DCOUPL 5)e uno per fornire la tensione adatta all'isolamento del rumore. Altri due pin sono dedicati a test generali sullo stato dell'integrato (GDO2 e GDO0, 3 e 6), un pin è dedicato al collegamento con una resistenza per la corrente di riferimento (EXTERNAL BIAS RESISTOR,RBIAS 17). Infine abbiamo due linee dedicate al I\O per valori positivi o negativi, sia in ingresso che in uscita, dei segnali ad RF (RF_P e RF_N, 12-13).

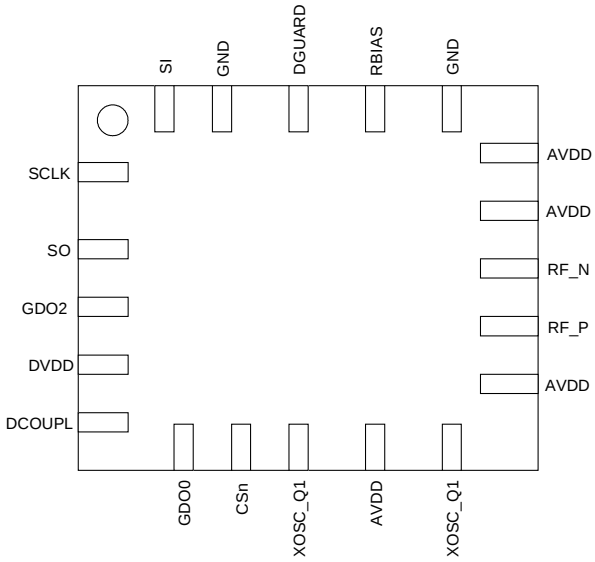

Figura 12: *top-view* dei pin del CC1100

3.2 *Parte di alimentazione*

Come già analizzato nel paragrafo 2.2.4, è necessario considerare il ricetrasmettitore nella sua totalità, quindi analizzare l'alimentazione di ogni sistema che lo compone. Nel nostro caso è particolarmente cruciale considerare l'alimentazione del PA, dell'LNA e dello stesso CC1100, figura 13.

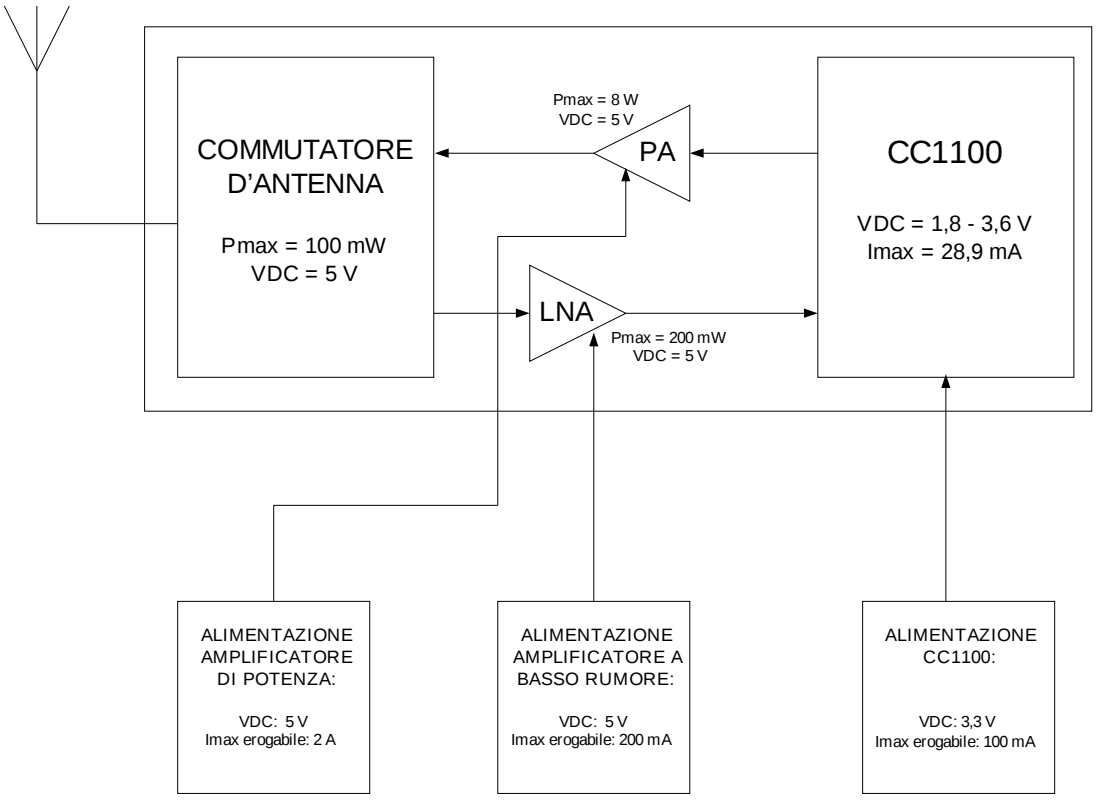

Figura 13: dettaglio dei principali blocchi d'alimentazione del ricetrasmettitore

3.3 *Circuito di connessione al Power Amplifier*

L'amplificatore di potenza per il trasmettitore AtmoCube è già stato progettato [10], e consiste in un amplificatore a radio frequenza, che riceve il segnale modulato da trasmettere e lo amplifichi, per esser poi irradiato dall'antenna di bordo; il segnale generato deve avere un livello di potenza adeguati per essere ricevuto correttamente dalla stazione a terra; nello studio fatto dallo studente Popesso[10] si considera come integrato l'ADF7020, ma vengono utilizzati valori compatibili anche con il CC1100 visto che in termini di frequenza di lavoro, potenza fornita in ingresso all'amplificatore e impedenza d'uscita (50 Ω) del trasmettitore le due componenti si equivalgono.
Si è scelto di dividere il sistema in una parte amplificatrice ed una parte di controllo della potenza.

Il blocco amplificatore deve produrre un segnale che deve essere ricevuto a terra, è necessario quindi portare il segnale emesso dal trasmettitore da 5-10 mW (7-10 dB) ad una potenza superiore ai 2 W (>33 dB), pertanto il guadagno dell'amplificatore dovrà essere almeno di:

$$G_{tot_dB} = P_{out_dBm} - P_{in_dBm} = 33\ dBm - 7\ dBm = 26\ dB$$

con P_{out_dBm} potenza in uscita espressa in dBm e P_{in_dBm} potenza in ingresso espressa in dBm.
Per raggiungere questo obiettivo è necessario dividere in più blocchi lo stadio di amplificazione. E' stata suppostala divisione, da accertare, in due parti, driver e stadio finale di potenza. Il blocco di controllo deve stabilizzare il livello di potenza in uscita, tramite la variazione di tensione di alimentazione del *driver*, inoltre attiva e disattiva l'intero amplificatore nei momenti di non utilizzo.
In figura 14 lo schema a blocchi dello stadio amplificatore.

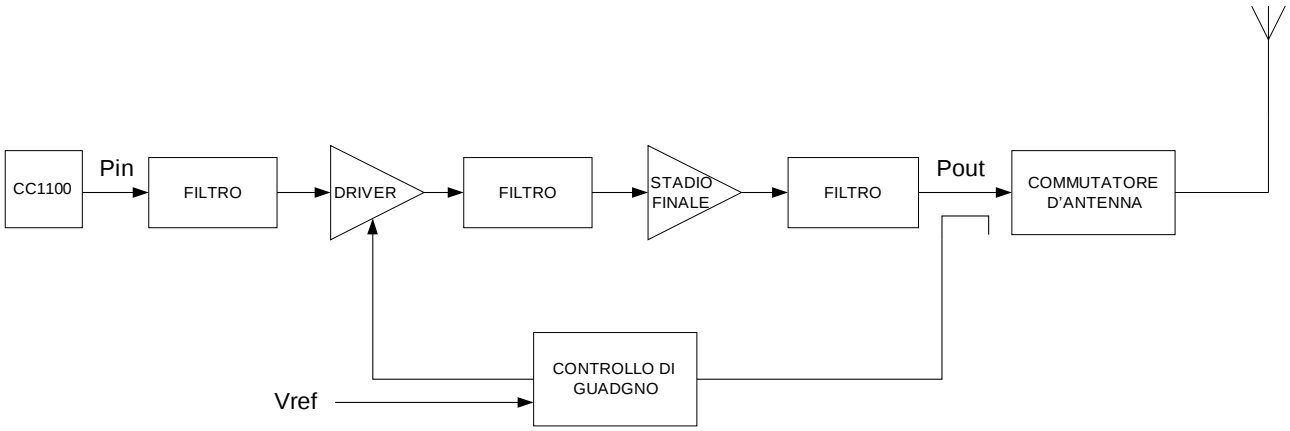

Figura 14: schema a blocchi circuito di connessione al PA

I filtri utilizzati attenuano le frequenze non desiderate e adattano le impedenze di ingresso e di uscita dei vari blocchi.

3.4 *Circuito di connessione al Low Noise Amplifier*

Al momento la progettazione di questa parte amplificatrice è in fase di studio, pertanto possiamo dire che non deve consumare più di 20-30 mA, che l'alimentazione necessaria è di 5 V e che la potenza massima richiesta si aggira intorno ai 200 mW.
Inoltre bisogna verificare se questo collegamento è realmente necessario; dalla tabella 2 del *uplink* si vede che la sensibilità richiesta per una trasmissione accettabile e di – 84,4 dB e, se andiamo a guardare le tabelle relative alla sezione in ricezione del CC1100 troveremo che, tralasciando le mo-

dulazioni a cui sono riferite, la sensibilità va da -111 dBm con una 1,2 kbps fino a – 88 dBm con 500 kbps di bit rate, quindi ben entro i limiti posti dallo studio del sistema Atmocube.

3.5 *Schema elettrico completo*

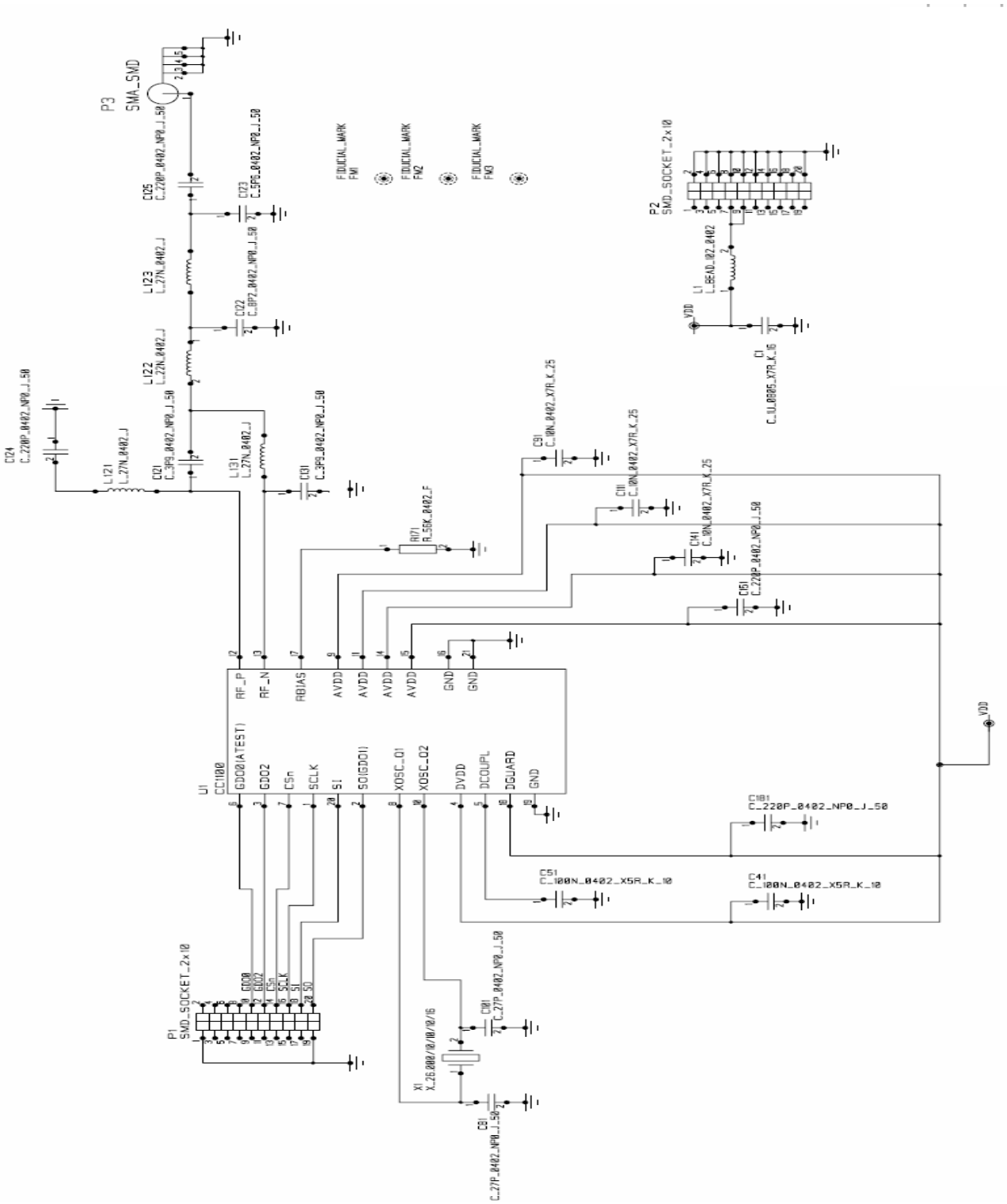

Figura 15: schema elettrico completo del CC1100

Lo schema rappresentato in figura 15 è stato fornito dal produttore della componente e progettato specificatamente per un integrato che lavori a 433 MHz. In tabella 8 troviamo i valori e lo scopo delle singole componenti collegate all'integrato.

Componenti	Descrizione	Valore
C51	Capacità di disaccoppiamento per i regolatori di tensione	100nF±10%
C81\C101	Capacità di carico del cristallo	27pF±5%
C121\C131	Capacità relative al circuito d'adattamento	3.9pF±0.25pF
C122	Capacità del circuito d'adattamento	8.2pF±0.5pF
C123	Capacità del circuito d'adattamento	5.6pF±0.5pF
C124	Capacità di blocco della corrente continua	220pF±5%
C125	Capacità di blocco della corrente continua	220pF±5%
L121\L131	Induttore relativo al circuito d'adattamento	27nH±5%
L122	Induttore relativo al circuito d'adattamento	22nH±5%
L123	Induttore relativo al circuito d'adattamento	27nH±5%
R171	Resistenza per definire la corrente di polarizzazione	56kΩ±1%
XTAL	Cristallo	26 - 27 MHz

Tabella 8: valori componenti esterni CC1100 a 433 MHz

L121,L131,C121 e C131 formano un circuito d'adattamento per il segnale ad RF.

3.6 *Specifiche di controllo*

Il CC1100 può essere configurato per ottenere le migliori prestazioni possibili in base all'uso che ne deve esser fatto; è possibile farlo tramite interfaccia SPI e può esser programmato per:

- Modalità di trasmissione\ricezione
- Accensione\spegnimento dell'oscillatore
- Selezione del canale RF
- *Data rate*
- Modulazione
- Banda del filtro di canale in RF
- Potenza in uscita
- Resettaggio del *chip*
- Supporto *hardware* dei pacchetti
- FEC e *Interleaving*

- *Data Whitening*

Le caratteristiche delle modalità da noi prese in considerazione sono state descritte nel paragrafo 2.3.2 relativo al circuito integrato CC1100. Ora verranno descritte le impostazioni dei registri per usufruire delle modalità inerenti al progetto.

La configurazione del CC1100 è fatta programmando registri da 8 bit; la configurazione adatta alle caratteristiche di sistema è facilmente trovabile usando il *software* SmartRF Studio, che descriveremo nel paragrafo successivo. Ci sono tre tipologie di registri differenti: 14 registri per i *Command Strobe*, che servono ad attivare le varie modalità del *chip*. Il *Command Strobe* può essere visto come un singolo *byte* di istruzione, ed il suo uso è generalmente per disabilitare il cristallo, abilitare la modalità di ricezione, abilitare la modalità di trasmissione; quarantasette registri da 8-*bit* sono di configurazione, la maggior parte dei quali utilizzati per *test* sul circuito integrato, e non devono essere scritti per le normali operazioni del CC1100; ci sono inoltre dodici registri di stato che sono di sola letture e contengono informazioni riguardo lo stato del *chip*.

Per quanto riguarda l'impostazione della *data rate* si rimanda al paragrafo 2.3.2. relativo al circuito integrato CC1100.

Il CC1100 ha tre modalità di stato differenti:

- Modalità ricezione
- Modalità trasmissione
- Modalità *IDLE* (Inattivo)

Le prime due modalità, chiamate *active modes*, sono attivabili dal MCU usando i *Command Strobes* SRX o STX, oppure automaticamente grazie al WOR, *Wake On Radio*, cioè una funzione opzionabile che permette di attivare l'integrato e porlo in ascolto di eventuali pacchetti senza che sia utilizzato l'MCU. Per queste modalità è necessario configurare il registro MCSM1, considerando RXOFF_MODE[1:0] per la parte in ricezione e TXOFF_MODE[1:0] per la parte in trasmissione; tramite il settaggio di questi *bit* è possibile scegliere cosa succede nel caso venga ricevuto o spedito un pacchetto. Le possibilità a nostra disposizione sono: passare allo stato IDLE(00), passare allo stato FSTXON (01) che consiste,sia in ricezione che in trasmissione, ad uno stato d'attesa in cui il sintetizzatore dell'integrato continua a lavorare, rimanere o passare, a seconda dello stato in cui ci si trovi, alla modalità di trasmissione (10) e rimanere o passare alla modalità di ricezione (11).

La modulazione è settabile nel registro MDMCFG2.MOD_FORMAT; andando a considerare i bit 6,5,4 di questo registro si può selezionare la modulazione voluta, così con 6[0] 5[0] 4[0] ho la FSK, con 6[0] 5[0] 4[1] ho la GFSK, con 6[0] 5[1] 4[1] ho ASK\OOK e con 6[1] 5[1] 4[1] una MSK.

Per programmare la deviazione in frequenza si considerano i valori DEVIATION_M e DEVIATION_E nel registro DEVIATN. Per calcolare la f_{dev} si utilizza la formula:

$$f_{dev} = \frac{f_{XOSC}}{2^{17}}(8 + DEVIATION_M) \cdot 2^{DEVIATION_E}$$

Considerando il fatto riscontrato durante le misure in laboratorio, cioè la necessità di dover resettare il sistema del ricetrasmettitore dopo ogni misura per non incorrere in errori di trasmissione, abbiamo due opzioni:

1) *Automatic power-on reset*

2) Risettaggio manuale

La prima opzione sfrutta il circuito di *power-on reset* situato all'interno del CC1100. Ogni qualvolta si fornisce di potenza l'integrato è necessario, successivamente, resettare, e riportarlo quindi alla configurazione di *default*. Se il ciclo di *reset* è completo il chip sarà nello stato IDLE e sia l'oscillatore interno all'integrato sia i regolatori di tensione saranno stabilizzati e staranno lavorando; se il *chip* ha avuto i tempo di stabilizzare i regolatori di tensione e gli oscillatori allora il pin SO andrà a "0" dopo aver posto a "0" anche CSn , nel caso il tempo non sia sufficiente il pin So andrà ad "1", comunicando che il cristallo e i regolatori di tensione non si sono stabilizzati, figura 16.

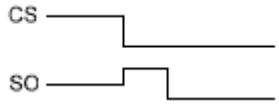

Figura 16: ***automatic power-on reset***

Il settaggio manuale sfrutta il *Command Strobe* SRES, e la configurazione necessaria è:

- Settare SCLK=1 e SI=0 per non permettere possibili problemi con la modalità di controllo dei pin (vedi sezione 11.3 pag.29 del *data sheet* [5])
- Porre *Command Strobe* CSn *low\high*
- Mantenere CSn ad "1" per almeno 40µs per portare CSn a "0"
- Porre CSn "0" e aspettare che anche SO vada a "0"
- Impostare SRES sulla linea SI
- Dopo che SO torna nuovamente a "0" il *reset* è completo

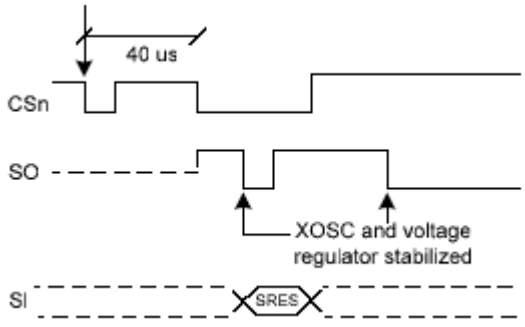

Figura 17: *manual reset*

Per quanto riguarda l'oscillatore , può essere acceso o controllato automaticamente settando il *bit* numero 0, XOSC_FORCE_ON, del registro MCSMO. Nella modalità di controllo automatico l'oscillatore viene controllato coi *Command Strobe* SXOFF o SPWD; nel primo caso viene spento mentre nel secondo va in modalità SLEEP, cioè minimo consumo possibile, è è da tener conto che è necessario esser in modalità IDLE.

Per la configurazione del filtro di canale si considerano i registri MDMCFG4.CHANBW_E e MDMCFG4.CHANBW_M. Il rapporto tra valori da attribuire ai registri e banda del filtro è data:

$$BW_{channel} = \frac{f_{XOSC}}{8 \cdot (4 + CHANBW_M) \cdot 2^{CHANBW_E}}$$

Per ottenere le migliori prestazioni possibili è necessario che la banda del filtro sia scelta in modo che il segnale occupi al massimo l'80% della banda del filtro di canale.

La configurazione per la potenza in uscita è gestita grazie alla PATABLE, per selezionare il settaggio del PA interno in dotazione al CC1100. La PATABLE è una tabella da 8-*byte* che definisce otto valori di potenza del PA selezionabili dal valore dato dai 3-*bit* denominati PA_POWER nel registro FREND0.

Il registro FREQ2, in particolare i *bit* denominati FREQ[21:16] permettono di selezionare la frequenza della portante desiderata. Nel nostro caso è necessario assegnare a FREQ2_FREQ[21:16] il valore 16, per avere la frequenza della portante a 436 MHz, considerando un *range* di frequenza compreso tra 436 MHz e 463,8 MHz.

Il formato dei pacchetti, figura 18, può essere configurato e le parti su cui si può lavorare sono:

- Preambolo
- Parola di sincronizzazione
- *Byte* di lunghezza o pacchetto di lunghezza costante programmabile

- Opzione *byte* di indirizzo
- *Payload*
- Opzione 2-*byte* CRC

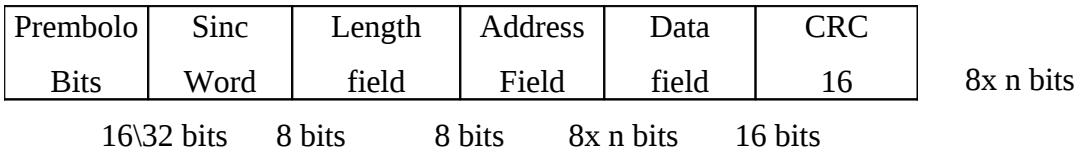

Figura 18: struttura pacchetto

Il settore del preambolo è una sequenza alternata di (0101010…), la lunghezza minima del preambolo è programmabile. Quando viene abilitata la trasmissione il modulatore inizia a trasmettere il preambolo. Quando il numero di bites di preambolo è stato spedito, viene spedita la parola di sincronizzazione e poi i dati dal TX FIFO se ci sono dati disponibili. Nel caso TX FIFO sia vuoto, il modulatore continuerà a spedire bites di preambolo fino a quando non viene scritto il primo byte nel TX FIFO. Quindi il modulatore spedisce la parola di sincronizzazione e poi i bytes dei dati. Il numero dei bites di preambolo è programmato con il valore MDMCFG1-NUM-PREAMBLE. La parola di sincronizzazione è un valore di due byte settati nei registri SYNC1 e SYNC0. La parola di sincronizzazione provvede, appunto, alla sincronizzazione dei pacchetti ricevuti.

CC 1100 supporta sia protocolli con pacchetti di lunghezza costante, sia protocolli con pacchetti di lunghezza variabile. Pacchetti di lunghezza variabile o fissi possono essere programmati fino a 255 bytes; per pacchetti più lunghi la modalità di lunghezza dei pacchetti che deve essere usata è la modalità lunghezza infinita. La modalità di pacchetti a lunghezza finita è selezionato settando PKTCTRL.LENGTH_CONFIG=0 . La lunghezza di pacchetto è settabile nel registro PKTLEN.

Nella modalità di lunghezza di pacchetto variabile, PKTCTRLO.LENGTH_CONFIG=1, la lunghezza del pacchetto è configurata dal primo byte dopo la parola di sincronizzazione. La lunghezza del pacchetto è definita dai dati del *payload*, escludendo il byte di lunghezza e i bit opzionali del CRC automatico. Il registro PKTLEN è usato per settare la lunghezza di pacchetto massima permessa in ricezione. Qualsiasi pacchetto ricevuto con lunghezza maggiore di quella impostata è scartato.

Con PKTCTRLO.LENGTH_CONFIG==2 la lunghezza del pacchetto è settata all'infinito e sia la trasmissione che la ricezione continueranno finchè non li si farà finire manualmente. La modalità infinita può essere spenta mentre un pacchetto è sul punto d'esser trasmesso o ricevuto. Questa configurazione permette l'uso di pacchetti con lunghezze differenti che il CC1100 supporta. La mini-

ma lunghezza di pacchetto supportata è un byte di *payload*. Il settore della lunghezza nel pacchetto può essere riprogrammata durante la ricezione e la trasmissione. Questo permette di avere una configurazione della lunghezza che supporta pacchetti a lunghezza variabile. All'inizio della ricezione, la lunghezza del pacchetto è settata su un valore grande. L' MCU (Microprocessore) legge abbastanza bytes per interpretare la lunghezza del campo nel pacchetto. Poi il valore di PKTLON è programmato in base a questo valore. La fine del pacchetto sarà rilevata quando il contatore dei byte nel gestore di pacchetto è uguale al registro PKTLEN (quindi quando avrà lo stesso valore). A questo punto la MCU deve essere adatta a programmare la lunghezza corretta, prima che il contatore interno raggiunga la lunghezza del pacchetto.

Per utilizzare l'opzione "pacchetto a lunghezza infinita", sono permessi pacchetti di lunghezza arbitraria. Prima di poterne usufruire , la modalità infinita deve essere attivata. Dalla parte del trasmettitore, il registro PKTLEN è settato a mod (length,256). Dalla parte dell' ricevitore l' MCU scorre abbastanza bytes per interpretare la lunghezza del campo nel pacchetto, e viene settato il registro PKTLEN in mod(length,256). Quando sono rimasti meno di 256 bytes del pacchetto l'MCU disabilita la lunghezza infinita del pacchetto e attiva la modalità pacchetti di lunghezza finita. Quando il contatore di byte interno raggiunge il valore PKTLRN, la trasmissione (o la ricezione) finisce. Può essere usato il CRC automatico, settando il registro PKTCTRL0.CRC_EN = 1. Facciamo un esempio della modalità sopra descritta, utilizzando un pacchetto di 600-byte da trasmettere,pertanto l' MCU dovrà eseguire le seguenti operazioni:

- settare PKTCRLO.LENGHTH.CONFIG=2 (10);
- pre-programmare il registro PKTCRL a mod(600,256)=88
- si trasmetta per esempio 345 *bytes*, riempiendo il TX-FIFO da 64 *byte* 6 volte (384 *bytes* trasmessi);
- settare PRTCRLO.LENGTH.CONFIG=0 (00);
- la trasmissione finisce quando il contatore di pacchetto raggiunge quota 88. un totale di 600 *bytes* sono stati trasmessi.

In sostanza, viene attivata la configurazione di pacchetto a lunghezza infinita,e in base alla lunghezza del pacchetto da trasmettere viene fissato il valore di PKTLEN. Vengono contati i byte, arrivati a 255 si riparte da 0. Quando il contatore rileva che mancano meno di 256 byte da trasmettere, scatta l'attivazione della modalità a pacchetto con lunghezza finita che invierà byte fino a raggiungere il valore di PKTLEN.

Nel mondo reale spesso abbiamo lunghe sequenze di "1" e di "0". Le performance possono essere migliorate codificando (*whitening*) i dati prima di trasmettere decodificandoli successivamente in ricezione. Nel CC1100 può essere fatto automaticamente settando il byte

PKTCTRLO_WHITE_DATA=1. Tutti i dati, eccetto il preambolo e la parola di sincronizzazione, sono poi fatti passare in una porta logica xor generando una sequenza *random* da 9-bit prima di essere trasmesse. Al ricevitore i dati sono nuovamente fatti passare in porte xor permettendoci di ricostruire così il segnale originale. Questa è la decodifica che ci permette di ricevere i dati spediti.

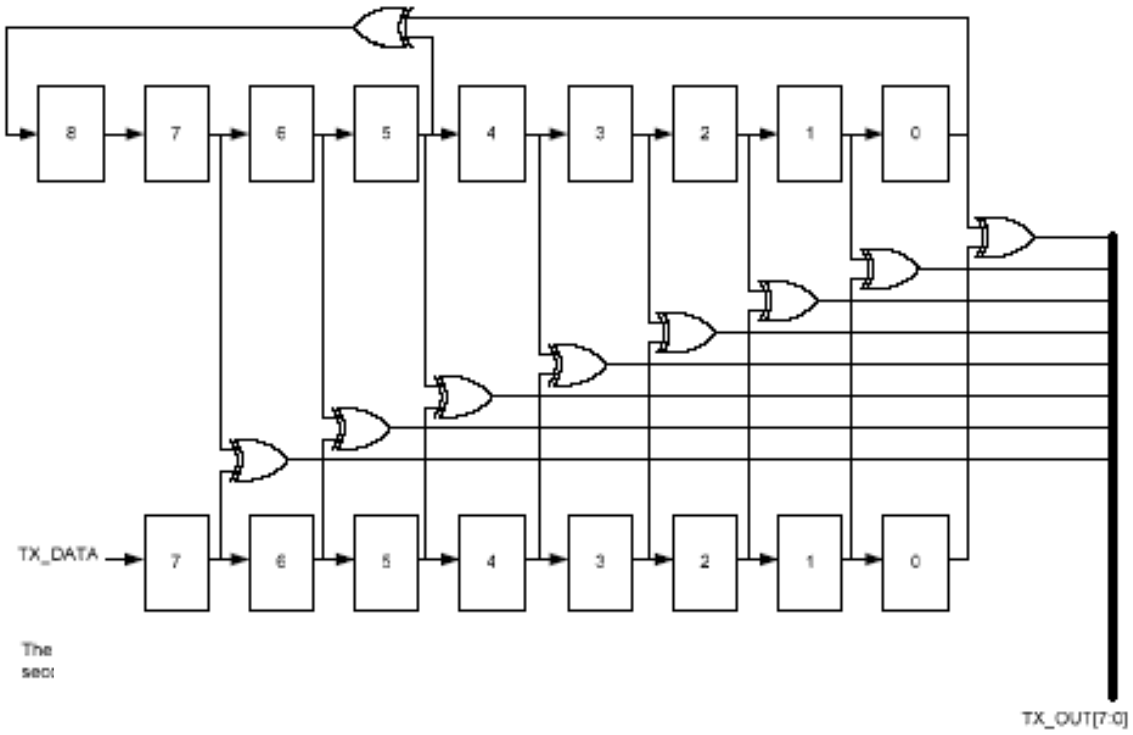

Figura 19: schema di funzionamento del *whitening*

La FEC è una correzione d'errori a posteriori ed è supportata dal CC1100; per abilitare questa opzione, assieme all'*interleaving*, bisogna settare MDMCFG1.FEC_EN=1, nel caso lo si voglia disabilitare è necessario porre uno 0. Il FEC è supportato solo nella modalità di pacchetto a lunghezza fissa. FEC è impiegato nel campo dei dati e nella parola di CRC per ridurre la BER quando si opera nei pressi dei limiti di sensibilità. E'introdotta ridondanza in trasmissione in considerazione del fatto che il ricevitore può recuperare i dati originali in presenza di alcuni bit d'errore. L'uso del FEC permette la corretta ricezione dei dati ad un basso SNR, in questo modo si amplia la portata delle comunicazione se la banda in ricezione rimane costante. Alternativamente, per un dato SNR, se si usa il FEC la BER decresce. Il PER (*packet error rate*) è in relazione col BER tramite l'equazione :

$$PER = 1-(1-BER)^{packet_lenght}$$

quindi un BER più basso può essere usato per pacchetti più lunghi, oppure una più alta percentuale di pacchetti di una certa lunghezza possono essere trasmessi con successo. Inoltre, nel caso reale, fenomeni transitori e tempo varianti possono produrre errori occasionali sebbene si sia nelle condi-

zioni di buona ricezione. Il FEC può proteggere degli errori combinato con l'*interleaving* dei dati codificati. Lo schema FEC adottato per il CC1100 è un codice convoluzionale nel quale n bits sono generati in base ai k bits in ingresso e agli m bits più recenti formando un flusso di codice (code stream) che è capace di contrastare un certo numero di bits d'errore tra ogni stato di codifica. Il codificatore convoluzionale ha un tasso di codice Rc = 1\2 con una lunghezza di vincolo costante di m=4; sostanzialmente il codificatore ha in ingresso un bit e ne fa uscire 2. Quindi richiederà una banda di ricezione più grande e sarà ridotta la sensibilità. Con il FEC si migliora la ricezione ma peggiora la sensibilità in relazione all'allargamento della banda.

I dati ricevuti attraverso un canale radio spesso saranno soggetti ad errori di *burst* dovuti a segnali interferenti e tempo invarianti. Per aumentare la robustezza delle informazioni contro gli errori si adotta l'interleaving combinato con il FEC.

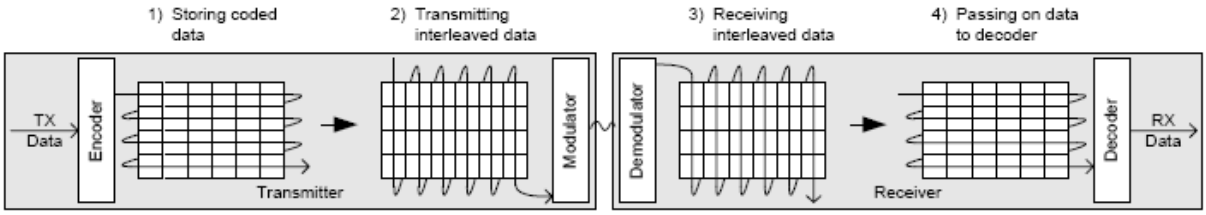

Figura 20: schema di funzionamento dell'*interleaving*

In figura 20 troviamo la matrice di *interleaving*. L'*interleaving* e il *de-interleaving* sono rappresentabili con matrici 4x4. Nel trasmettitore i *bits* di informazione sono scritti nelle righe della matrice dal momento che la sequenza di *bit* da trasmettere è letta partendo dalle colonne della matrice. Nel ricevitore, invece, i simboli ricevuti sono scritti nella colonna della matrice e dopo che i dati sono passati nel decodificatore convoluzionale (fase di *de-interleaving*) le informazioni sono lette dalle righe della matrice. Quando FEC e *interleaving* sono usati, un extra *byte* è richiesto a chiusura dello schema a traliccio.

3.7 *Il prototipo*

La ditta Elcon ha messo a nostra disposizione due integrati, i quali verranno usati alternativamente come trasmettitore e ricevitore per testare la qualità di trasmissione. Il chip CC1100 è montato su una *evaluation board* CC1100EM, adottando la configurazione descritta nei paragrafi precedenti per la frequenza a 433 MHz.

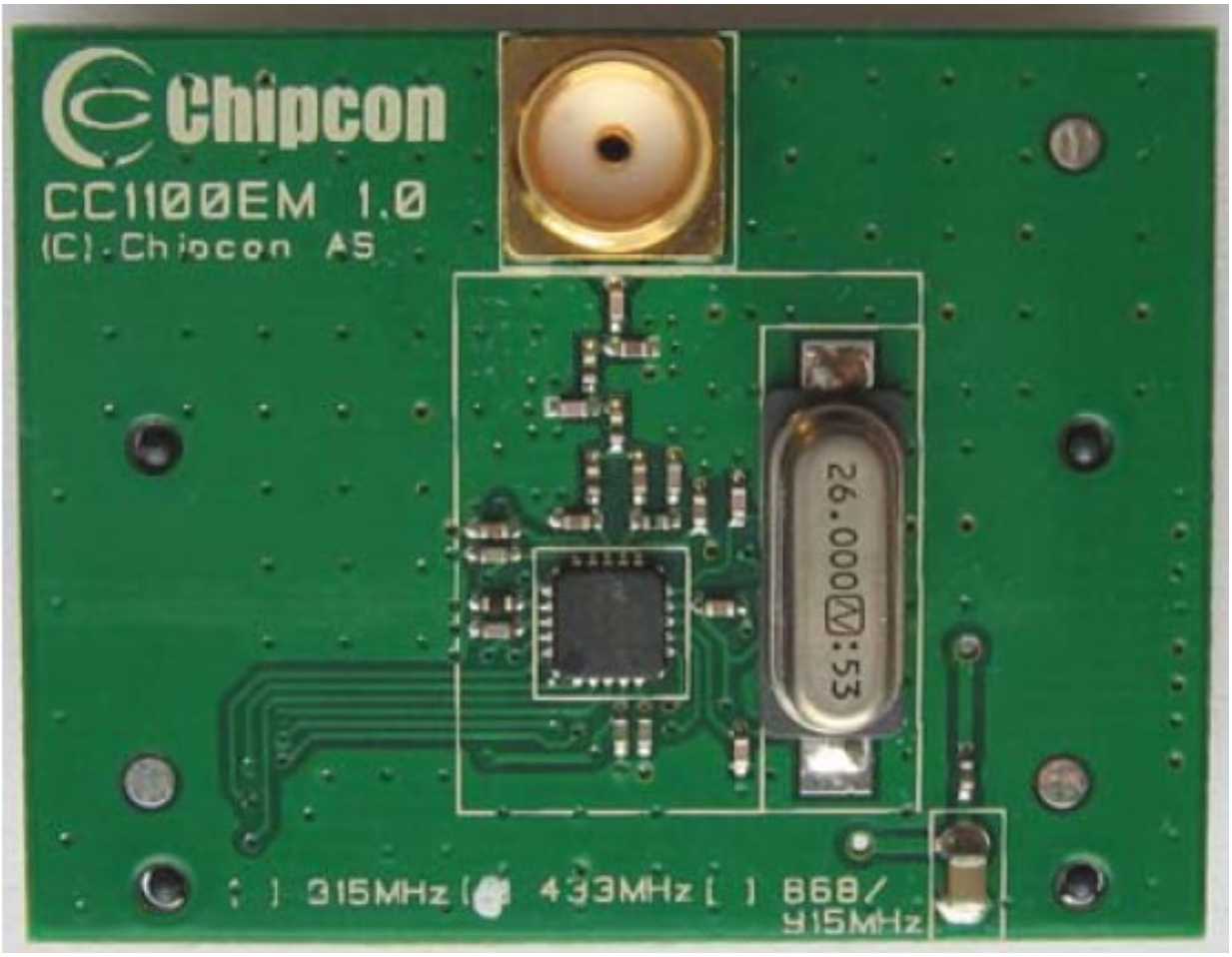

Figura 21: CC1100 montato su *evaluation board* con configurazione a 433 MHz

La scheda è controllata da un microcontrollore. E' stato utilizzato, grazie alla fornitura di Elcon, un µC montato su una scheda guasta di un cambio automatico per trattori. Questa scheda viene usata per i *test*; è stato verificato quale fosse il guasto del µC, dopodiché è stato configurato in maniera compatibile al nostro integrato, utilizzando per il collegamento dei fili connessi direttamente ai pin disponibili e non guasti del µC.

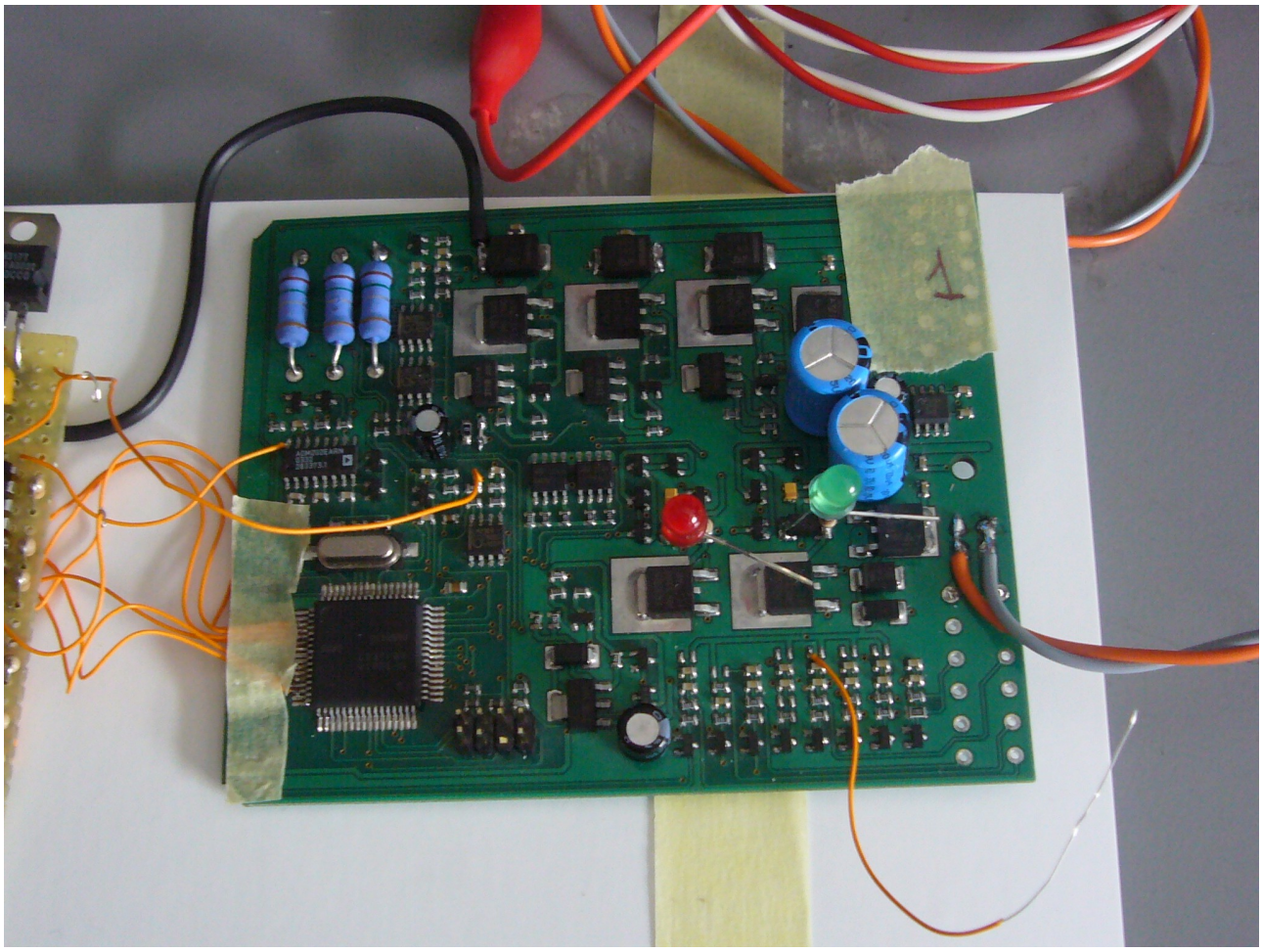

Figura 22 : scheda con µC

La Elcon ha costruito inoltre uno traslatore di livello, lo schema elettrico in figura 23, da porre tra il µC e l'integrato. Il suo compito è quello di convertire i livelli di tensione tra µC e il CC1100, visto che il primo lavora ad una tensione di 5 V mentre il secondo lavora a 3,3 V. Il blocco LM317 è un alimentatore variabile, cioè un alimentatore che in base ai carichi connessi eroga una certa tensione. I blocchi IC1 e IC2 sono dei buffer converters, invertitori di fase, della famiglia HCF4049 prodotti dalla Philips; uno , quello nella parte destra, IC2 è dedicato al lato CC1100 mentre il blocco IC1 e dedicato al lato del µP.

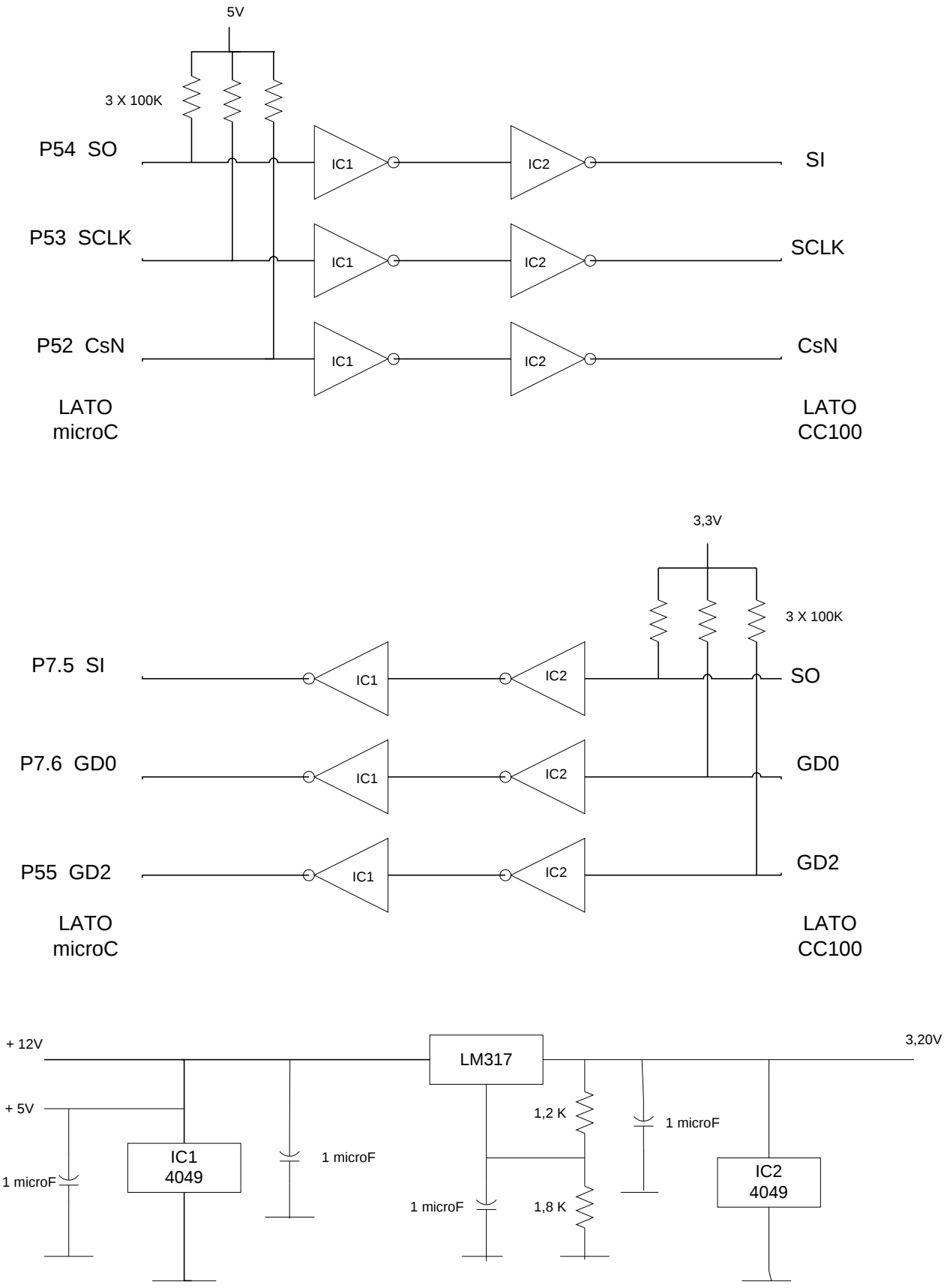

Figura 23: Schema elettrico traslatore di livello

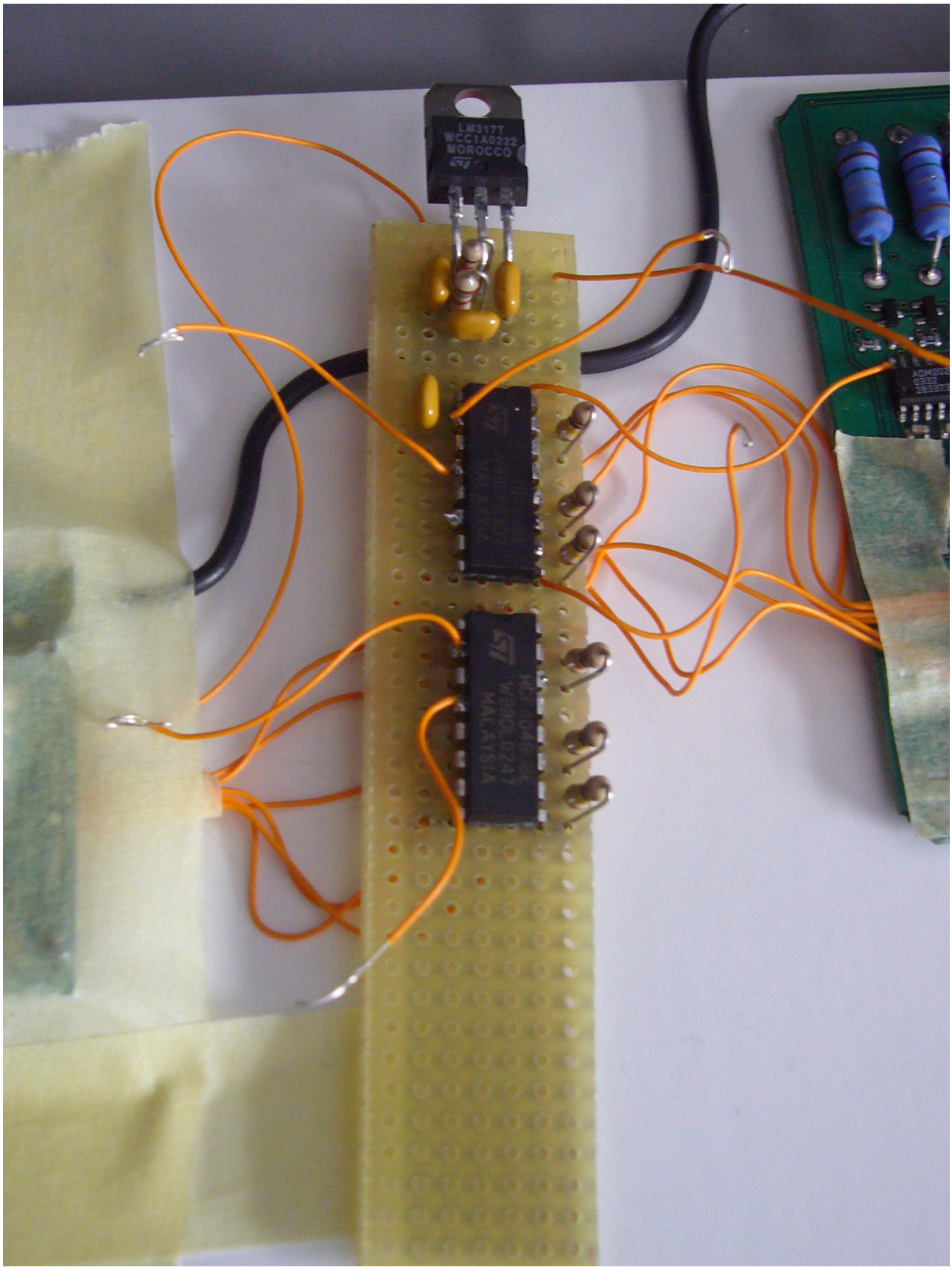

Figura 24: traslatore di livello

Come si vede in figura 25 , le varie componenti che andranno a formare il ricetrasmettitore possono avere questo tipo di struttura. Come già detto in precedenza, avendo due CC1100, entrambi sono stati utilizzati, cosicché si potesse valutare la qualità della trasmissione. Un ricetrasmettitore è stato denominato “1” ,figura 25 ,ed è stato utilizzato in prima istanza per la ricezione e successivamente per la trasmissione, mentre l’altro ricetrasmettitore, denominato “2”, figura 26, è stato utilizzato per la trasmissione e poi per la ricezione.

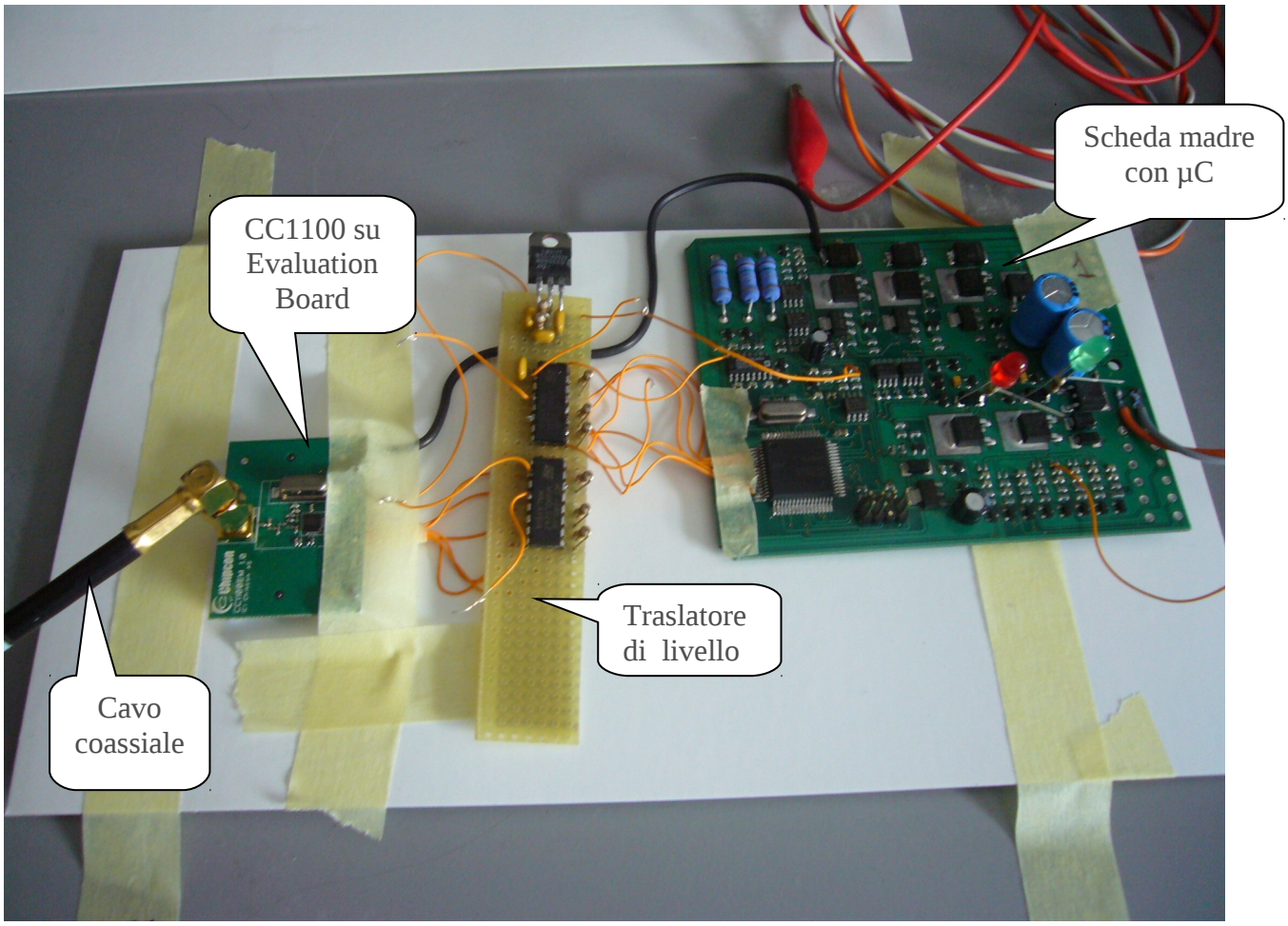

Figura 25: sistema completo del ricetrasmettitore numero 1

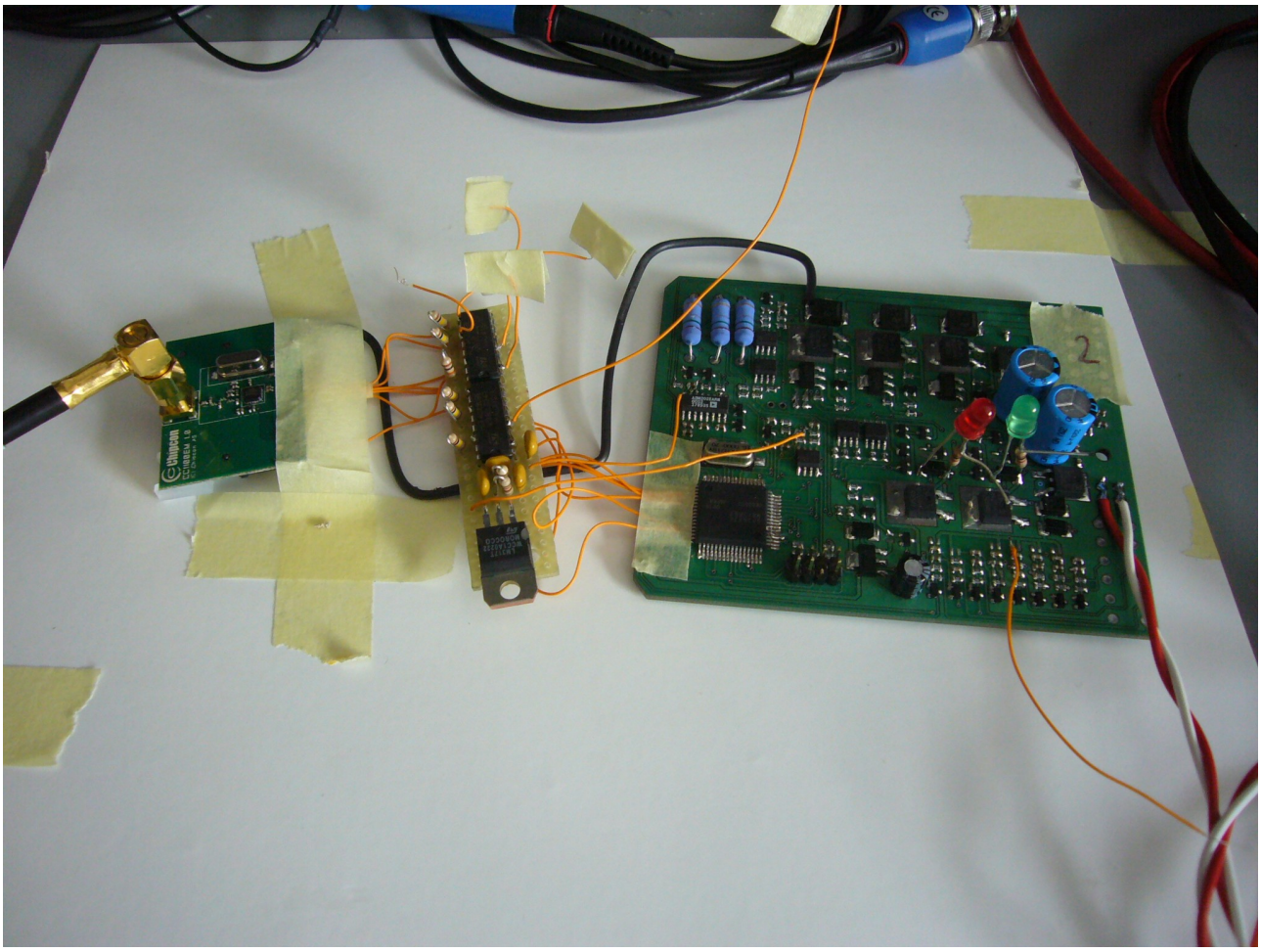

Figura 26: sistema completo del ricetrasmettitore numero 2

Lo schema a blocchi del sistema da noi costruito, in figura 27.

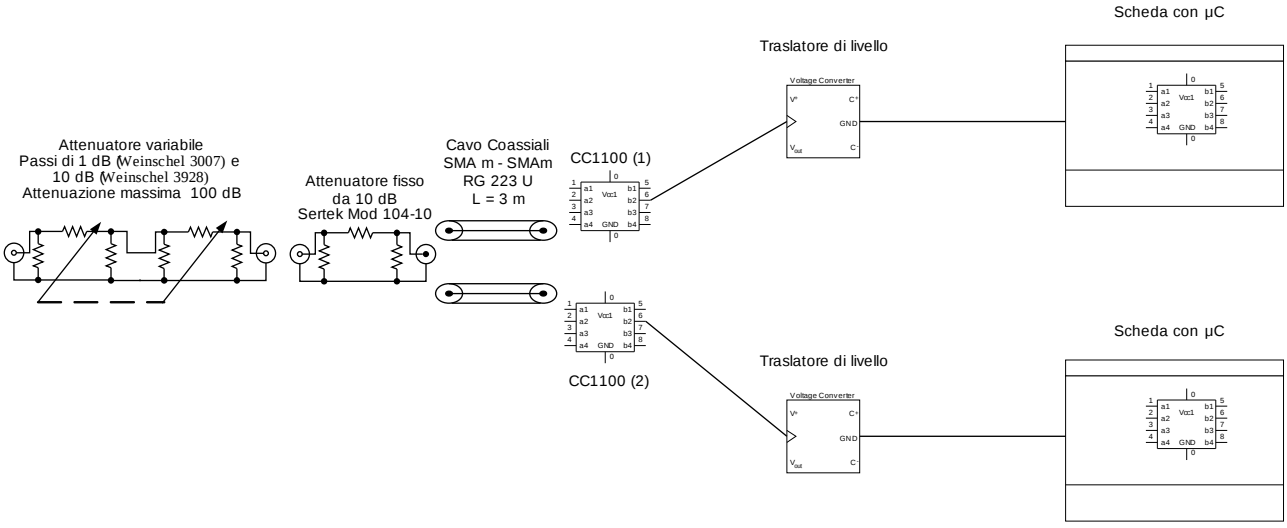

Figura 27: schema a blocchi sistema di misura

E in figura 28 la foto del sistema costruito in laboratorio

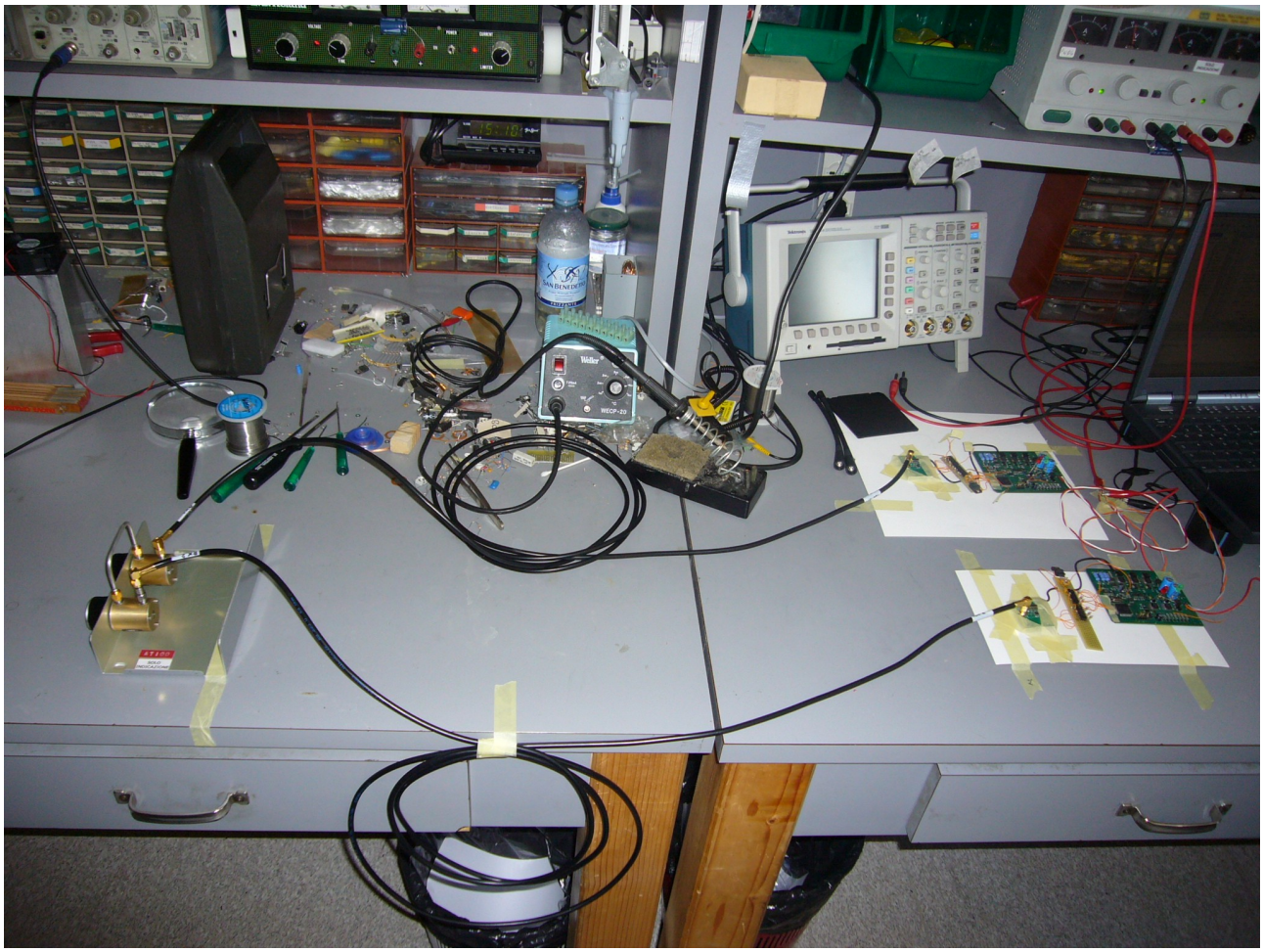

Figura 28: sistema sviluppato in laboratorio

3.8 *Misure*

Dopo avere assodato com'è strutturata effettivamente la configurazione del sistema, elenchiamo brevemente le tipologie di misure e test da effettuare per vagliare la bontà delle componenti:

1. Misura della trasmissione di pacchetto con attenuazione del canale
2. Misura dell'indice di modulazione e della deviazione di frequenza
3. Controllo dello spettro di potenza del segnale di preambolo

3.8.1 *Misura 1*

La misura numero 1 consiste nel simulare una attenuazione tra l'apparecchio trasmittente e quello ricevente, per verificare fino a quale attenuazione la comunicazione non viene corrotta. Per simulare, approssimativamente, la trasmissione su un canale è stato costruito dal professor Fragiacomo, con l'appoggio della Telit, un attenuatore (50 Ohm) variabile composto da due attenuatori; un attenuatore variabile da 0 a 10 dB con incrementi di 1 dB, (Weinschel 3007.), l'altro un attenuatore variabile da 0 a 90 dB con incrementi di 10 dB, (Weinschel 3928). I due attenuatori sono interconnessi con una cavo coassiale rigido della lunghezza di 12 cm e presentano connettori SMAf.
L'attenuatore è collegato tramite due cavi coassiali da 50 Ohm RG 223 U terminati SMA m – SMA m della lunghezza di 3 m alle schede dei nostri integrati. Dopo le prime misure è stato aggiunto un attenuatore fisso da 10 dB (Sertek Mod 104-10) tra il cavo e l'attenuatore da 90 dB per permettere una più ampia scala di valutazione.
La misura dell'attenuazione è stata fatta con una analizzatore di reti vettoriale (VNA) della HP modello 8714 B. Lo strumento è stato calibrato in modo trasmesso con cavo coassiale RG316 SMAm – SMAm della lunghezza di 50 cm e con due adattatori Nm – SMAf.

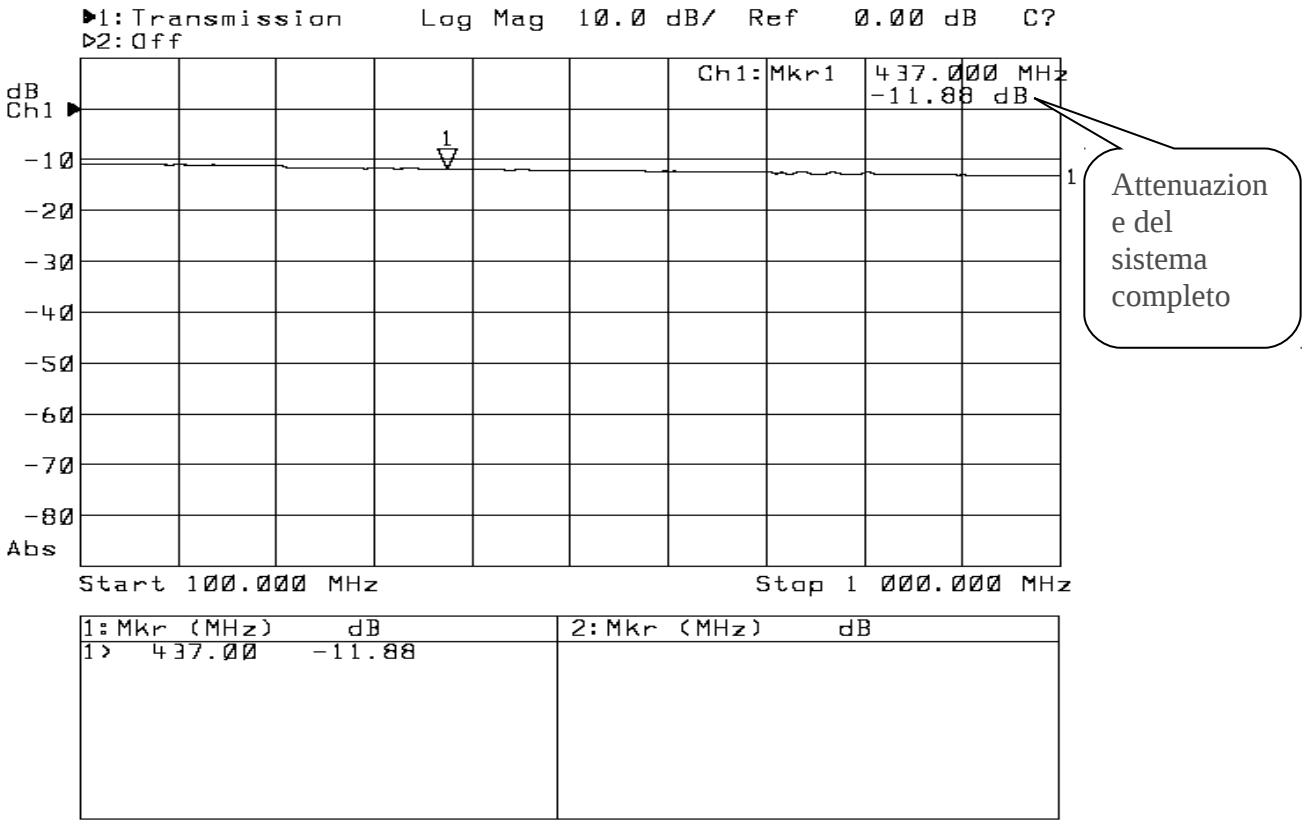

Figura 29: attenuazione del sistema completo

Come vediamo in figura 29 l'attenuazione del sistema attenuatori più cavi, con gli attenuatori nella posizione 0 dB è di 1, 88 dB. Di conseguenza al valore indicato sugli attenuatori si dovrà aggiungere 1,88 dB.

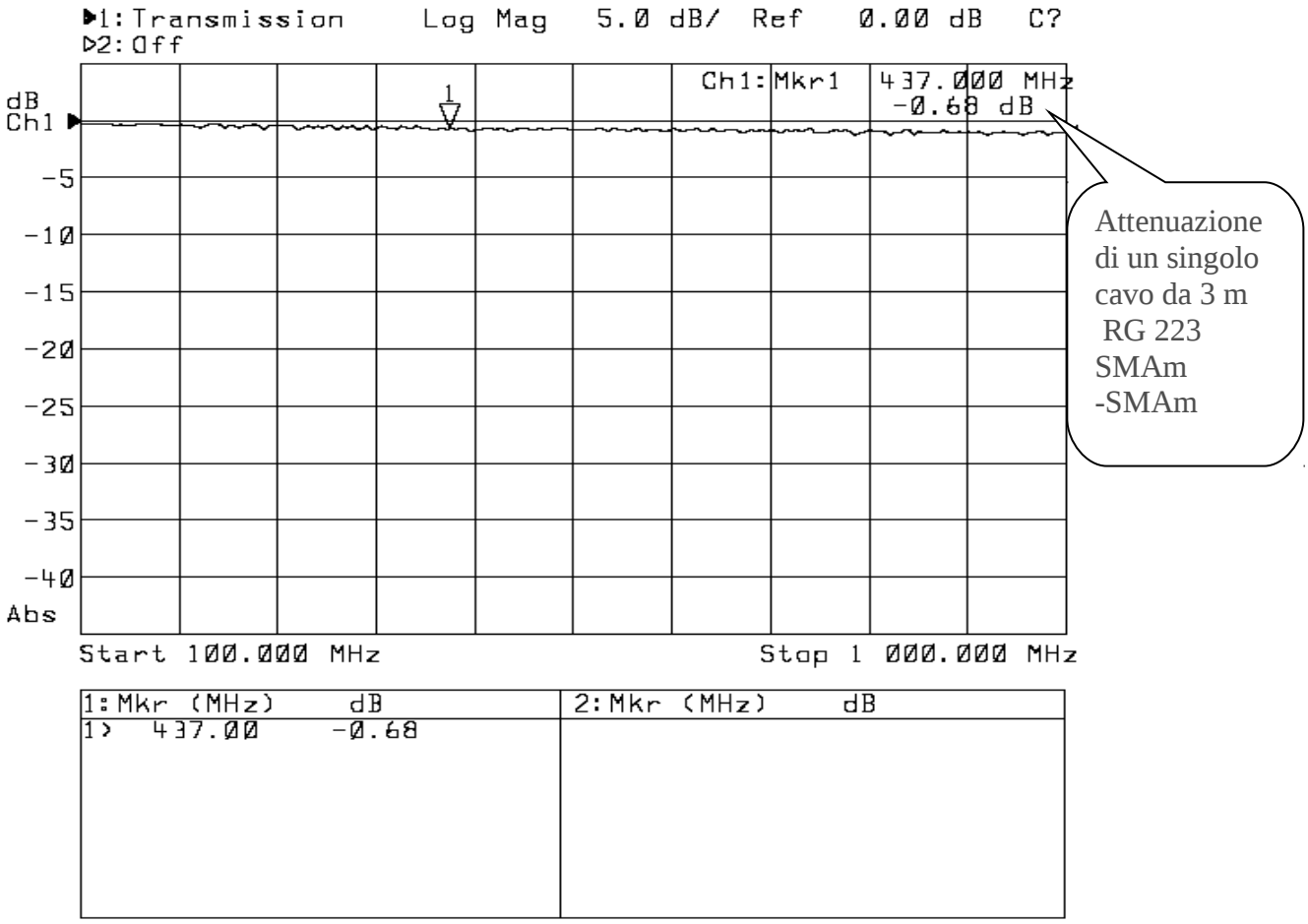

Figura 30: attenuazione cavo coassiale

In figura 30 invece vediamo l'attenuazione di uno dei due cavi coassiali 50 Ohm RG 223 U terminati SMA m – SMA m della lunghezza di 3 m .

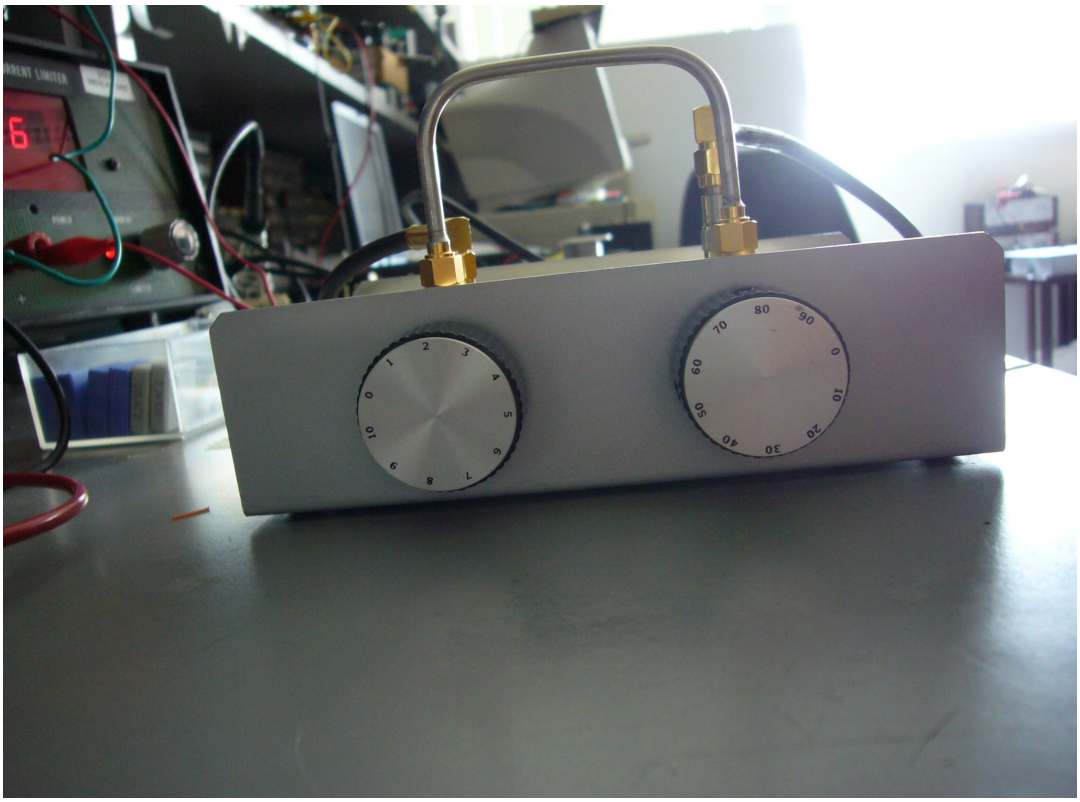

Figura 31: attenuatore variabile

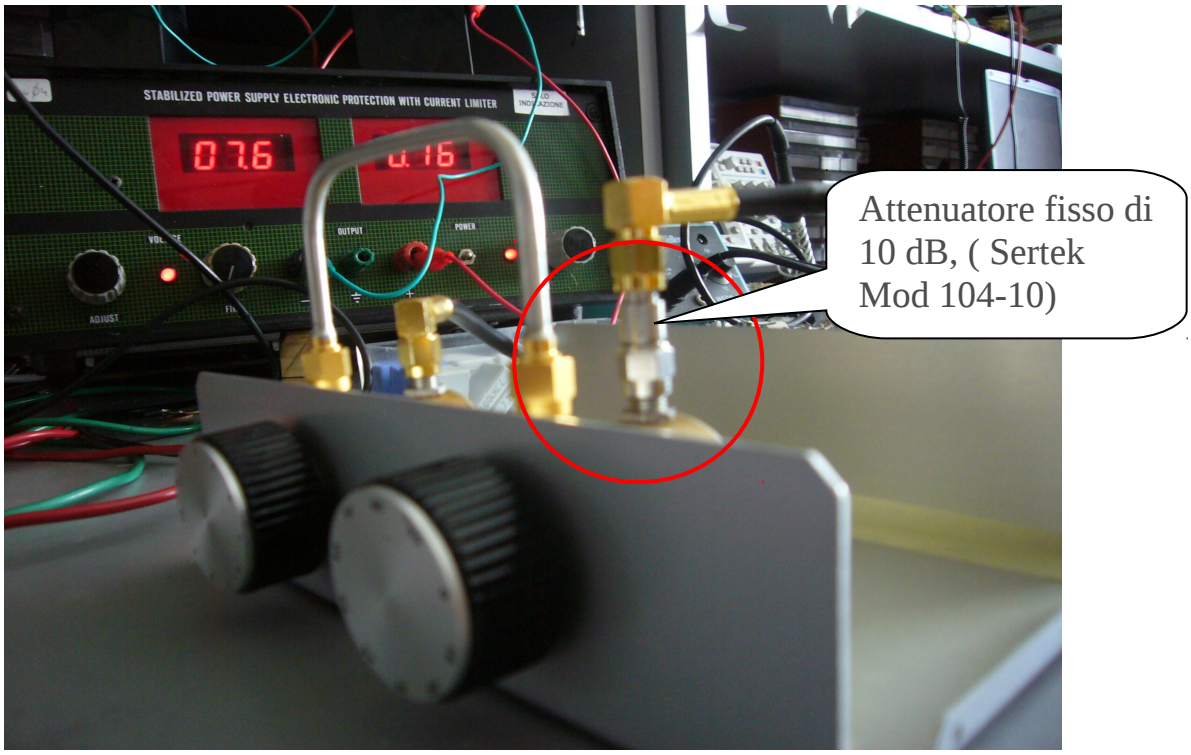

Figura 32: particolare dell'attenuatore fisso montato sull'attenuatore variabile

Per la parte in trasmissione dell'RXTX è stato configurato che per ogni contatto venissero spediti 10 pacchetti da 11 byte l'uno, che chiameremo sessione di pacchetto. La potenza d'emissione, lavorando ad una frequenza di 433 MHz, è stata impostata su 0 dBm, con una bit rate di 250 kbps. Come si può notare da figura 33, il nostro ricetrasmettitore è collegato ad una scheda su cui è montato il µC;il sistema costruito è dotato di *led*, uno di colore rosso ed uno di colore verde. Una volta generato l'impulso che permette al trasmettitore di inviare i pacchetti entrambi i *led* si accenderanno, contemporaneamente, se avviene una trasmissione corretta, dovrebbero accendersi anche i *led* del ricevitore. Di seguito il *led* rosso del trasmettitore inizierà a lampeggiare per 5 volte, ogni cambio di stato (*on**off*) del *led* comporterà un pacchetto inviato. Se il *led* rosso del ricevitore si accende e si spegne ogni volta allo stesso istante del *led* in trasmissione allora c'è sincronismo e i pacchetti vengono ricevuti regolarmente altrimenti ci sarà la perdita di alcuni pacchetti.
L'obbiettivo di questa misura è esattamente questo, cioè valutare fino a quale attenuazione la comunicazione non viene corrotta. Sono state fatte dieci misurazioni nei pressi delle attenuazioni massime sopportabili dal CC1100, prima trasmettendo dal sistema "1" verso il "2", in seguito dal "2" al "1":

	Dall' "1" al "2"		Dal "2" al "1"	
Attenuazione	Esito +	Esito -	Esito +	Esito -
90 dB	10	0	10	0
92 dB	9	1	2	8
93 dB	8	2	\	\
94 dB	2	8	\	\

Tabella 9: esito misura 1

Prendendo in considerazione i valori di tabella 9 si vede che con una attenuazione fino a 93 dB la comunicazione è accettabile, anche se il 20% di perdite non è un valore molto buono, mentre per attenuazioni maggiori il numero di sessioni di pacchetto perse diventa esagerato. Confrontando le specifiche del *data sheets* ([5], pag 10) notiamo che i nostri integrati le soddisfano pienamente avendo una sensibilità in ricezione di ≈ 93 dB con 1% di errore di pacchetto, sia che si utilizzi come ricevitore il sistema "1" sia si adoperi il sistema "2". Un'anomalia che è stata riscontrata consiste nel fatto che, quando un pacchetto non viene trasmesso correttamente, tutte le trasmissioni successive falliscono, non riuscendo a trasmettere più in maniera corretta; alla luce di questo fatto sarebbe opportuno predisporre che il µC dopo un certo periodo di tempo resettasse l'integrato, andando così

a ripristinare le condizioni iniziali. Infatti per le misure dall' "1" al "2" con 90 dB d'attenuazione, l'integrato è stato resettato; l'ultima misura, ad una diversa attenuazione, aveva dato errore nella trasmissione degli ultimi pacchetti, questo fatto ha comportato che tutta la misurazione successiva fosse falsata, appunto quella a 90 dB d'attenuazione; dopo l'azione di *reset* la misurazione è andata a buon fine senza che venisse perso alcun pacchetto.

3.8.2 *Misura 2*

La misura numero 2 consiste nel valutare, grazie ad un analizzatore di modulazione, strumento in basso in figura 33, e ad un analizzatore di spettro, strumento in alto in figura 33, l'indice di modulazione dell'integrato [1], la deviazione di frequenza del segnale e lo spettro del segnale.
Nell'analisi di modulazione in frequenza è abitudine considerare il rapporto tra la deviazione di frequenza Δf e la banda di trasmissione , appunto l'indice di modulazione, definito dal parametro β:

$$\beta = \frac{\Delta f}{B}$$

Incrementando l'ampiezza del segnale modulante viene prodotto un aumento della banda occupata dal segnale FM, il quale corrisponde all'aumento della deviazione in frequenza Δf e di conseguenza anche di β.

Figura 33: oscilloscopio e analizzatore di modulazione

In questo caso viene inviato agli strumenti la sequenza di preambolo, una serie di 0 ed 1 alternati, figura 34.

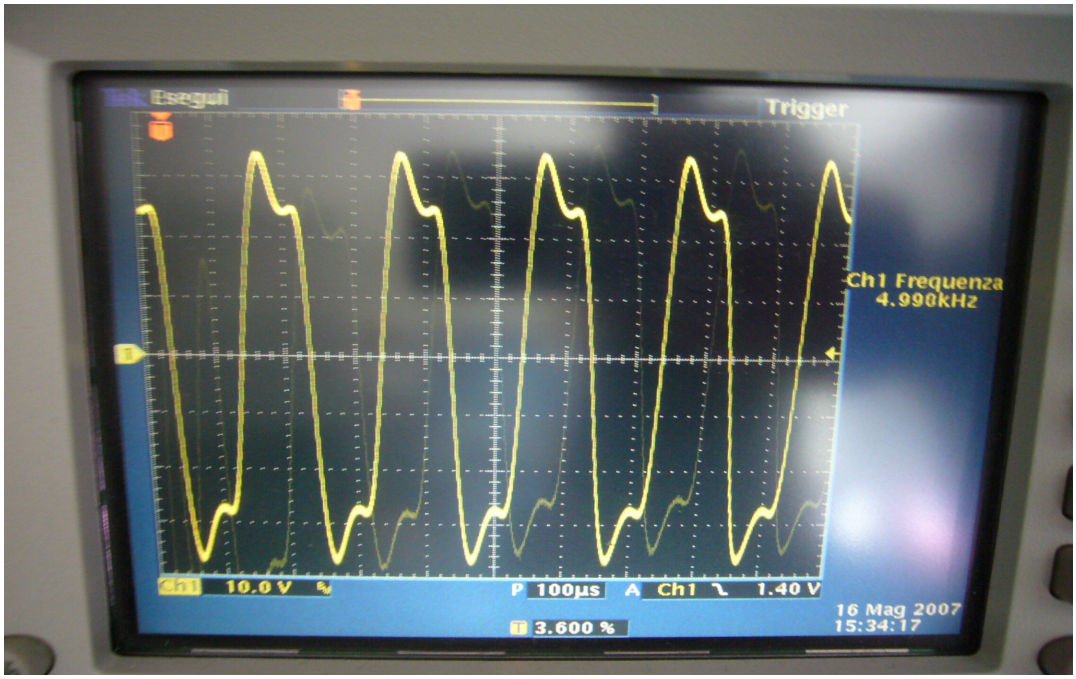

Figura 34: sequenza di preambolo

L'analizzatore di modulazione figura 35, collegato al ricetrasmettitore tramite cavo coassiale, riceve il preambolo, ci fornisce nei display digitali l'indice di modulazione, e la deviazione di frequenza del segnale trasmesso, oltre la frequenza del segnale portante.

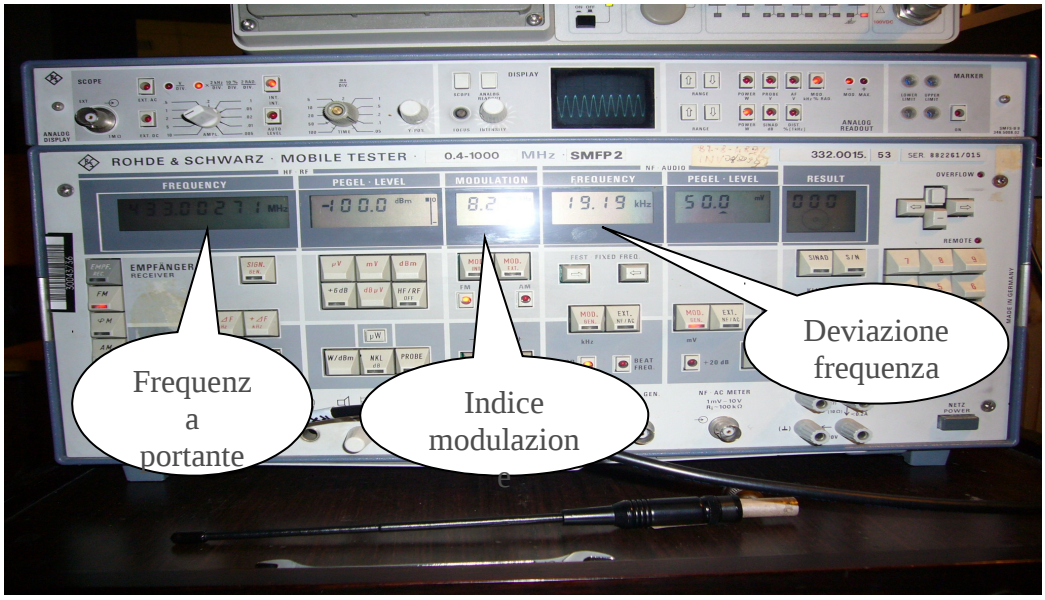

Figura 35: analizzatore di modulazione con valori per 34.4 kbps

La prima misura è stata impostata con valori non prettamente inerenti al sistema Atmocube; essa ha il compito di testare in linea generale l'integrato, verificando che i valori descritti nel *data sheet* rispecchino la realtà. Il ricetrasmettitore, per la prima fase della misurazione, lavora ad una frequenza di 433 MHz, con una *bitrate* di 34,4 kbps. L'analizzatore di modulazione ha registrato un indice di modulazione 8,22 kHz, con una deviazione di frequenza di 19, 19 kHz.
Successivamente è stato collegato il cavo coassiale all'analizzatore si spettro,che ha permesso di vedere nel *display* lo spettro del segnale, figura 36.

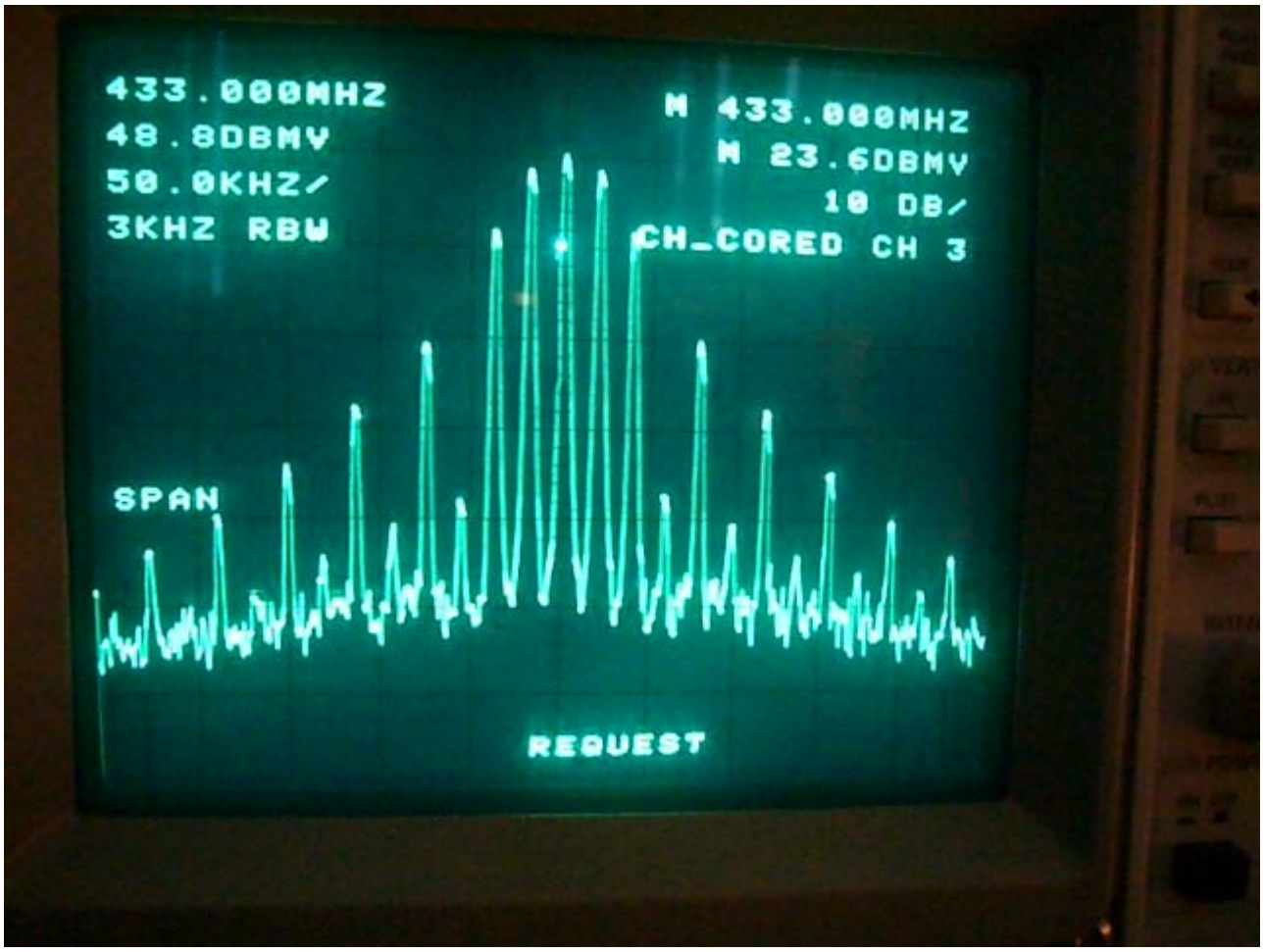

Figura 36: spettro segnale di preambolo a 433MHz*, bit rate* 34.4 kbps, scala 50 kHz

In seconda istanza si è modificata la *bit rate* del ricetrasmettitore, passando a 10 kbps, molto vicino al valore da noi ricercato (ndr. 9,6 kbps). In questo caso l'analizzatore di modulazione non è riuscito a fornirci un indice di modulazione, in compenso ha fornito la deviazione in frequenza del segnale modulato, che è di 4,996 kHz, e la forma d'onda del segnale tramite il *display* del piccolo oscilloscopio in dotazione all'analizzatore di modulazione, figura 37.

Figura 37: analizzatore di modulazione con valori per 10 kbps

Successivamente il cavo coassiale connesso al ricetrasmettitore è stato collegato all'analizzatore di spettro, il quale ha fornito la forma dello spettro del segnale che il CC1100 produce.

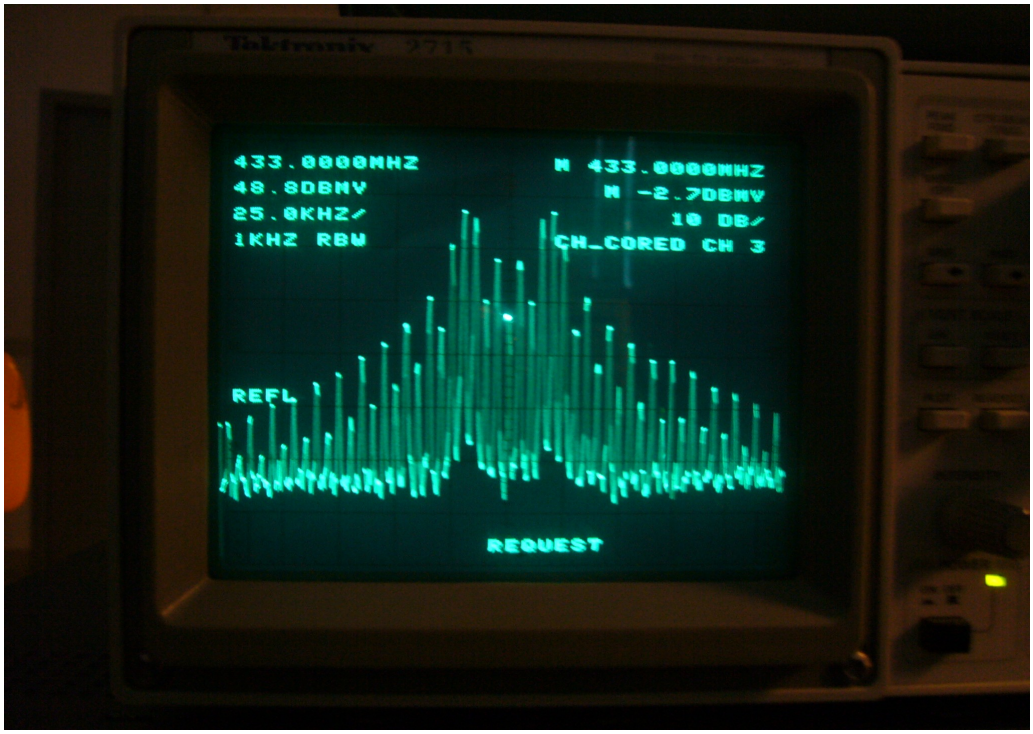

Figura 38: spettro segnale di preambolo a 433MHz, *bit rate* 10 kbps , scala 25 kHz

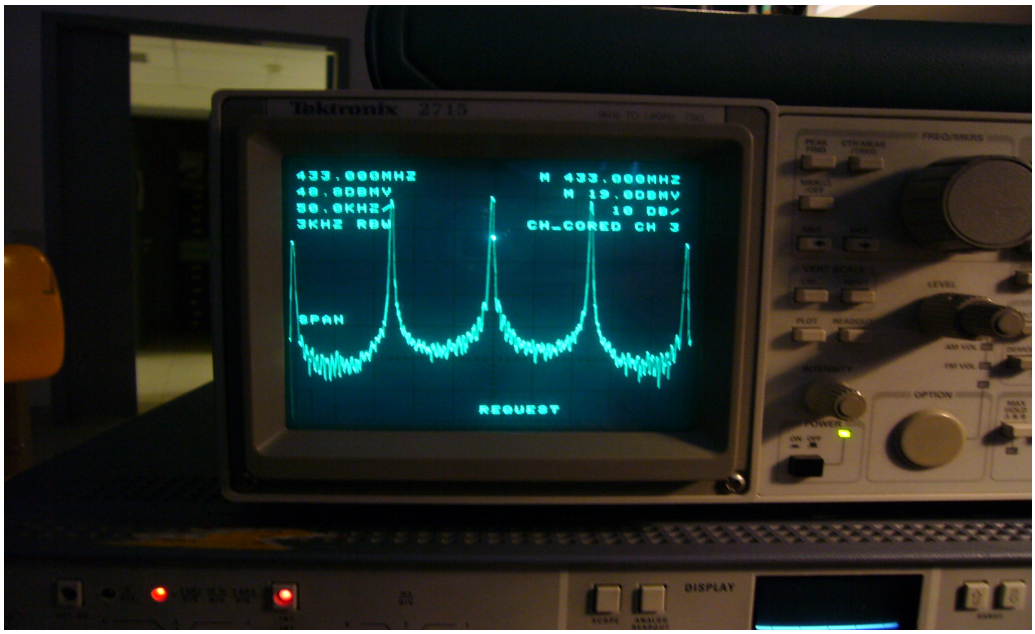

Figura 39: spettro segnale di preambolo a 433MHz, *bit rate* 10 kbps , scala 50 kHz

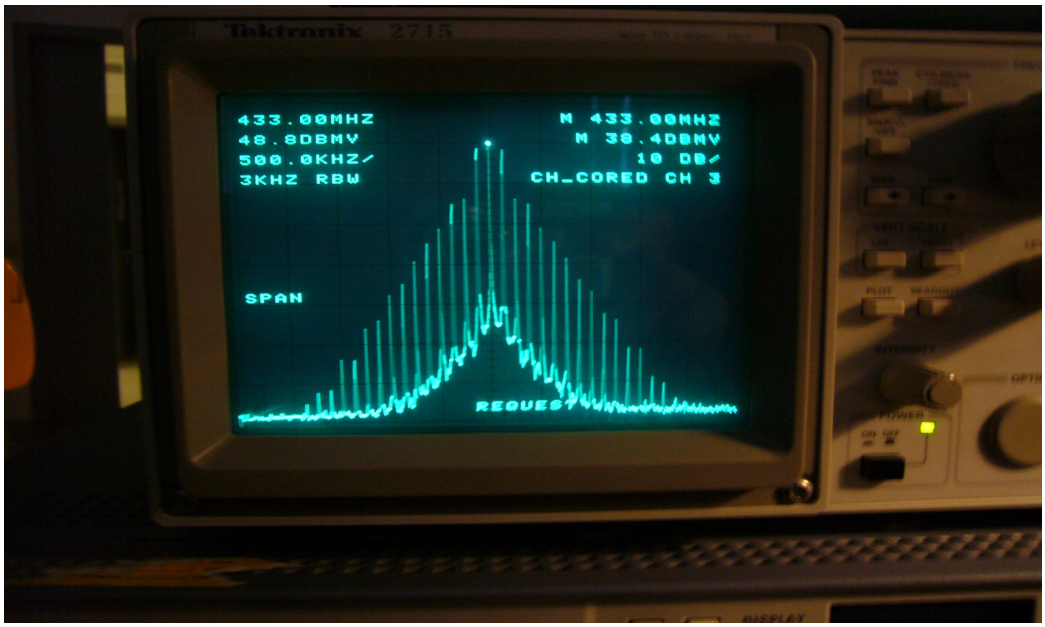

Figura 40: spettro segnale di preambolo a 433MHz, *bit rate* 10 kbps , scala 500 kHz

		Frequenza di lavoro 433 MHz	
Bit Rate		38,4 kbps	10 kbps
Indice di modulazione		8,2 kHZ	Non rilevato
Deviazione frequenza		19,2 kHz	4,996 kHz

Tabella 10: esito misura 2

3.8.3 *Misura 3*

La terza sessione di misure sfrutta il sistema creato per la prima serie di misure, modificando alcuni parametri. La potenza d'emissione, lavorando ad una frequenza di 433 MHz, è stata impostata nuovamente a 0 dBm, mentre la bit rate è stata modificata, configurandola a 10 kbps, molto vicina a quella realmente utilizzata di 9,6 kbps; è da evidenziare che più è alta la bit rate più la banda si allarga; pertanto, avendo ridotto la bit rate di test da 250 kbps a 10 kbps, anche la banda del segnale subirà una riduzione.
In questa misurazione di è tenuto conto di un *range* d'attenuazioni maggiore, visto che queste saranno le impostazioni del ricetrasmettitore sul satellite, si è voluto testare a fondo l'integrato.
Questa volta si è trasmesso solamente dall'integrato 1 verso il 2 e non viceversa.

	Dall' "1" al "2"	
Attenuazione	Esito +	Esito -
90 dB	10	0
91 dB	10	0
92 dB	10	0
93 dB	10	0
94 dB	10	0
95 dB	10	0
96 dB	10	0
97 dB	10	0
98 dB	9	1
99 dB	7	3
100 dB	0	10

Tabella 11: esito misura 3

Prendendo in considerazione i valori di tabella 11 si vede che con una attenuazione fino a 97 dB la comunicazione è presumibilmente perfetta, non perdendo nemmeno una sessione di pacchetto nella trasmissione; con una attenuazione di 98 dB si ha la perdita di una sessione su 10, già un 10% di perdita sul totale. A 99 dB la trasmissione inizia ad essere deficitaria, non riuscendo a trasmettere 3 sessioni dei 10 totali, mentre a 100 dB la comunicazione non avviene, visto che il CC1100 non riesce a trasmettere correttamente nessuna sessione di pacchetto. Confrontando le specifiche del data sheets ([5], pag. 10) notiamo che i nostri integrati le soddisfano pienamente avendo una sensibilità in ricezione di ≈ 93 dB con 1% di errore di pacchetto, sia che si utilizzi come ricevitore il sistema "1" sia si adoperi il sistema "2".

3.8.4 *Conclusioni sulle misure*

Mentre la seconda sessione di misure ha avuto uno scopo didattico e di controllo della bontà dell'integrato, la prima e in particolar modo la terza sessione, sono utili per l'effettivo utilizzo dell'integrato.

Si è scelto d'adottare come potenza emessa dal trasmettitore 0 dBm, vuoi perché si ha un risparmio di energia rispetto la potenza massima erogabile di 10 dBm (circa 13,4 mA di differenza), vuoi perché comunque si fa affidamento nel successivo stadio di potenza per un incremento della potenza del segnale trasmesso, vuoi perché si temeva di bruciare il *chip* e per non rischiare, data la stretta vicinanza tra gli integrati, che ci fosse comunicazione diretta tra il trasmettitore e il ricevitore.

Per la misura effettuata a 250 kbps, controllando nel *data sheet* ([5] pagina 10), la sensibilità attesa è quella da noi riscontrata; questa misura ci fa capire che l'integrato soddisfa le potenzialità determinate dalla casa produttrice ma al fine dell'effettivo uso che dovremo fare i dati non verranno usati come riscontro data la *bit rate* troppo grande utilizzata.

La terza ed ultima misura è, invece, molto importante ed utile. Il dato ottenuto, lavorando a 433 MHz con *bit rate* 10 kbps e potenza 0 dBm, è una sensibilità in ricezione di – 98 dBm (valore confermato dal manuale fornito con l'integrato ma non presente nel *data sheet*).

Considerando il *link budget*, ho calcolato il valore della sensibilità del ricevitore, che è data da :

$$Sensitivity_{link_budget} = EIRP + L_s + G_r$$

con EIRP *Effective Isotropic Radiate Power*, L_S attenuazione di spazio libero e G_r guadagno effettivo dell'antenna in ricezione.

Per il calcolo mi sono aiutato col foglio elettronico, elaborato da Cuttin, scaricabile al sito www.units.it\carrato\didatt . L'EIRP da me considerato è lo stesso supposto per la stazione di Basovizza dato che, anche nel caso si utilizzi un CC1100, la potenza emessa in trasmissione sarà la stessa, progettando quindi un trasmettitore secondo i dati già determinati nel *link budget.*

$$EIRP = P + G_T + L_t$$

con P potenza trasmessa , in dBW, G_T guadagno effettivo dell'antenna trasmittente e L_t attenuazione della linea collegante il trasmettitore all'antenna; da specificare che , relativamente a quanto detto prima, si considera il guadagno dell'antenna della stazione di Basovizza, avendo pre-

messo che qualsiasi sia il trasmettitore i dati da utilizzare si vuole siano quelli determinati nel *link budget*. Per utilizzare i valori corretti si rimanda alla tabella 2, relativa all'*uplink.* Quindi otterremo:

$$Sensitivity_{link_budget} = EIRP + L_s + G_r = -17{,}82 - 148{,}2 + 1{,}56 = -164{,}46\ dBW = -134{,}46\ dBm$$

Si vede quindi che:

$$Sensitivity_{link_budget} - \ Sensitivity_{test} = -36{,}46\ dBm \cong 4{,}425\ W$$

La differenza tra il segnale in ingresso al ricevitore, dopo il passaggio per l'antenna ricevente, e la sensibilità del ricevitore è di 36,46 dBm, pari a 4,425 W.
Si suppone che questa differenza possa essere colmata o amplificando ulteriormente il segnale della stazione di terra, se si utilizza un CC1100, aggiungendo un PA, oppure aggiungendo un LNA in ingresso al ricetrasmettitore installato nel satellite assieme al PA del trasmettitore; è da tenere in considerazione che più si amplifica il segnale più si amplifica il rumore.
Altra possibilità per aumentare notevolmente la sensibilità in ricezione è diminuire la *bit rate* a 1,2 kbps, questo però potrebbe portare dei problemi in quanto ad una bassa *bit rate* il satellite non riuscirebbe a spedire tutte le informazioni in quanto il tempo trascorso all'inclinazione utile per la trasmissione non sarebbe sufficiente; altro fattore contro l'abbassamento della *bit rate* è l'inevitabile restringimento della banda del segnale.

4. RF Studio

L'RF Studio software è un programma "*giftware*" fornito dalla Chipcon scaricabile al sito www.-chipcon.com . La sua utilità sta nel poter configurare alcuni integrati della Chipcon collegando il proprio pc alla *evaluation board* su cui risiede la componente. E' possibile connettersi sia via porta USB sia con una porta parallela, entrambe le soluzioni sono supportate dal sistema operativo Windows XP. E'da porre attenzione al fatto che se l'integrato lavora ad una tensione differente da quella del pc c'è il rischio venga bruciato per la troppa tensione; come possibile soluzione si consiglia l'utilizzo di un traslatore di livello.

Una volta installato e lanciato il programma ci troveremo di fronte ad una finestra che ci permetterà di scegliere la componente da configurare. Nel caso del CC1100 è possibile considerare la *Calculation Windows* cioè una finestra in cui è possibile settare le caratteristiche da noi volute e calcolare i valori dei registri per quella configurazione; essa è uno strumento di valutazione e non comunica con l'integrato vero e proprio. Nel caso si connetta l'EB al computer comparirà nella finestra d'apertura del programma la dicitura "*New Device* – CCXX", selezionandolo e ciccando due volte viene attivata la componente.

Una volta selezionata la *Calculation Windows* utile, si presenterà una finestra come in figura 41.

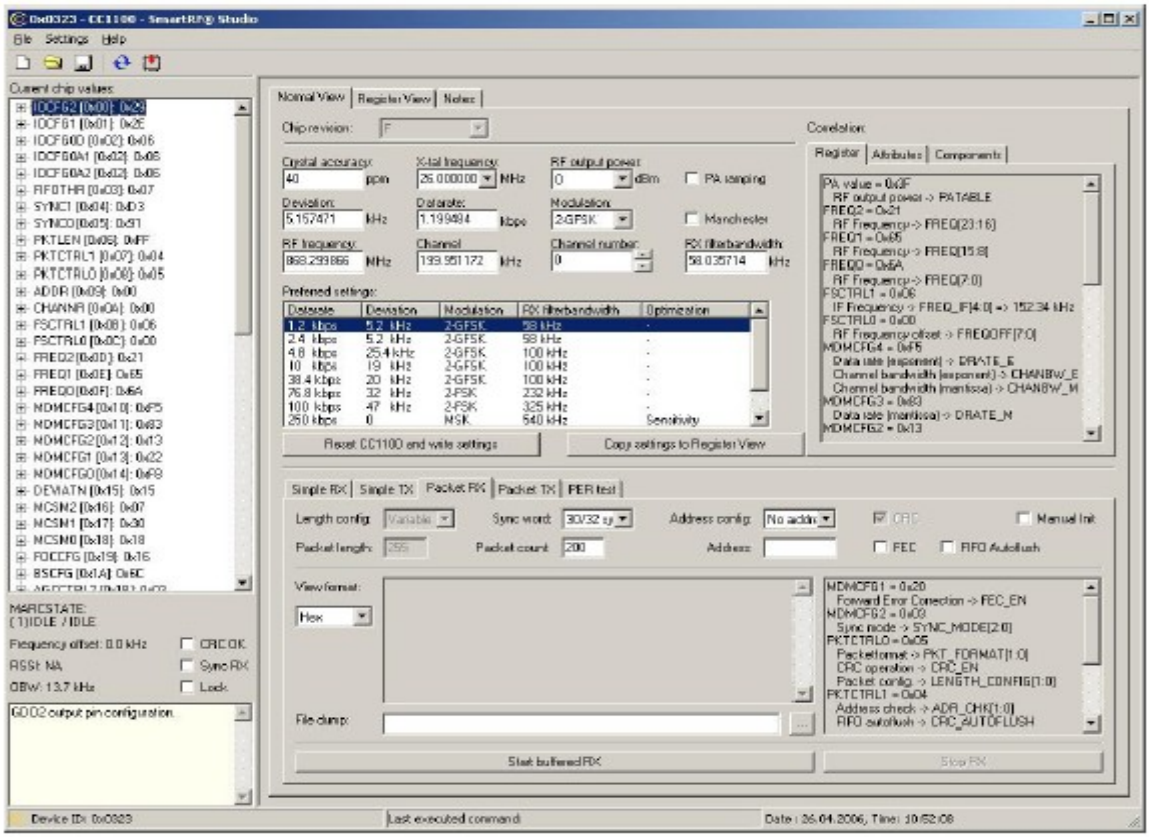

Figura 41: ***screen-shot*** **RF studio,** ***normal view***

Nella colonna a sinistra si trova lo "*Register Status*", da il valore dei registri dell'integrato connesso al pc,nel caso di "*Calculation Windows*" i valori saranno tutti 0x00; l'altra parte della finestra è dedicata alla impostazione dei parametri di sistema e alla lista dei valori dei registri in base ai parametri di sistema impostati. In figura 42 un dettaglio della parte in cui configurare i parametri di sistema.

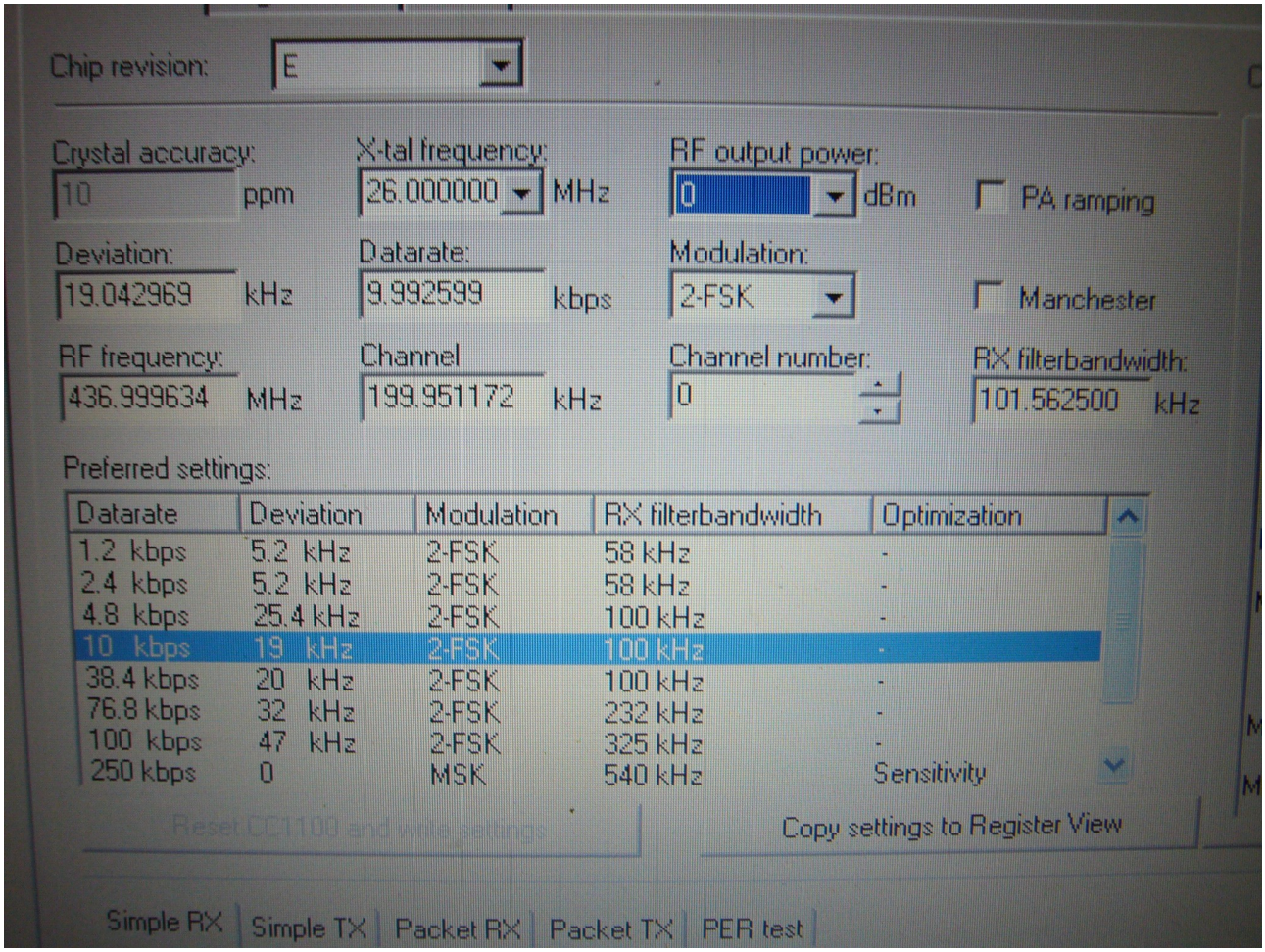

Figura 42: dettaglio impostazioni RF Studio

Quanto descritto sopra corrisponde alla modalità "*Normal View*", figura 41,ma è possibile selezionare anche la modalità "*Register View*", figura 43, che permette la configurazione manuale dei registri del CC1100, impostando manualmente i valori da attribuire a singoli *bit* dei registri.

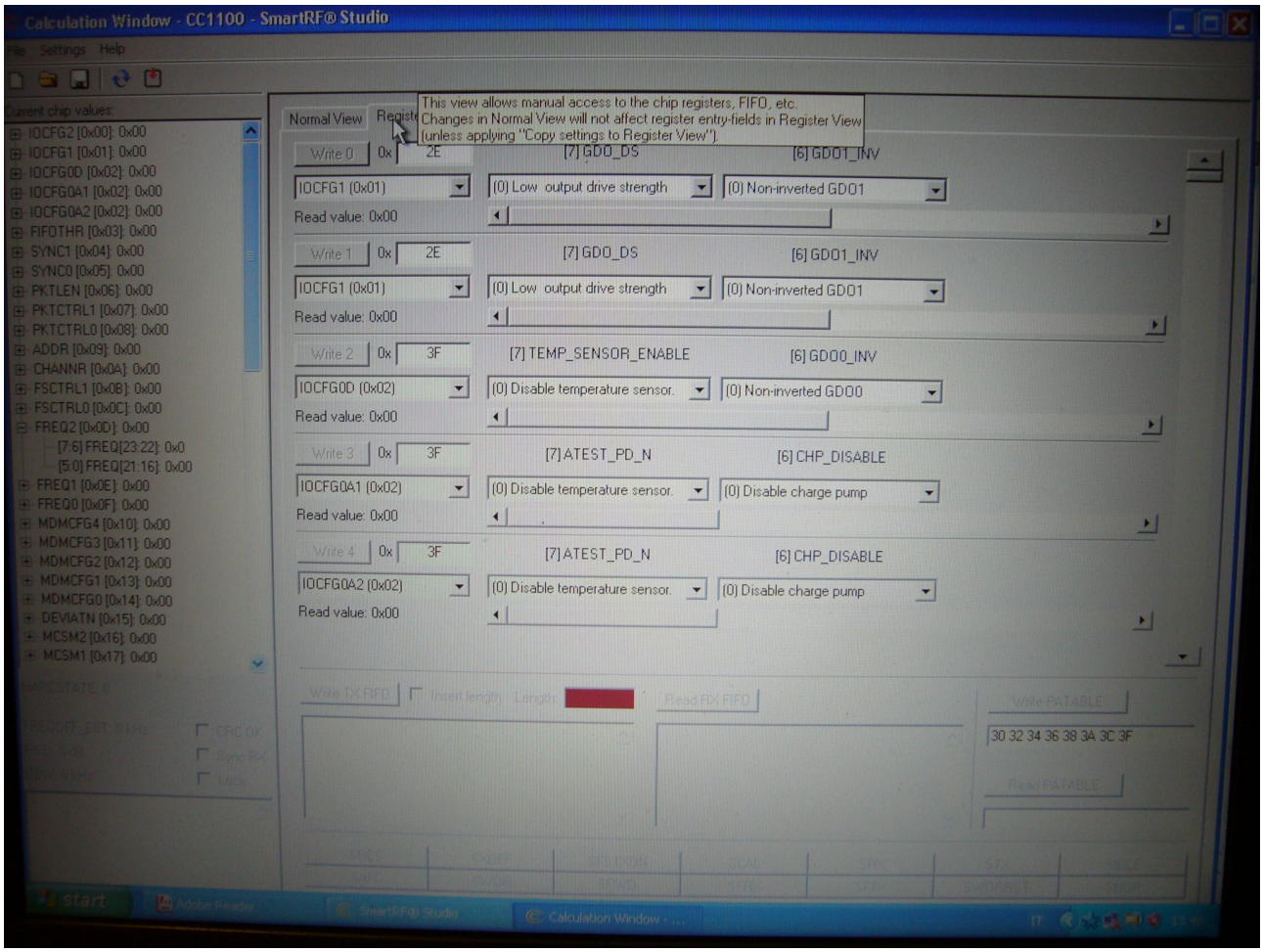

Figura 43: ***screen-shot*** **RF Studio,** ***register view***

Di seguito riporto i valori dei registri più significativi tratti dal programma RF Studio nel rispetto delle specifiche elencate:

- Precisione cristallo 10 ppm
- Frequenza del cristallo 26 MHz
- Potenza in uscita alle RF di 0 dBm
- Deviazione di frequenza 20 kHz
- *Data rate* 10 kbit/s
- Modulazione 2-FSK
- Frequenza RF 437 MHz
- Canale a frequenza 200 kHz
- Banda del filtro ad RF 101,56 kHz

Valori dei registri:

PA value = 0x3F

RF output power -> PATABLE

FREQ2 = 0x10

RF Frequency -> FREQ[23:16]

FREQ1 = 0xCE

RF Frequency -> FREQ[15:8]

FREQ0 = 0xC4

RF Frequency -> FREQ[7:0]

FSCTRL1 = 0x06

IF Frequency -> FREQ_IF[4:0] => 152.34 kHz

FSCTRL0 = 0x00

RF Frequency offset -> FREQOFF[7:0]

MDMCFG4 = 0xC8

Data rate (exponent) -> DRATE_E

Channel bandwidth (exponent) -> CHANBW_E

Channel bandwidth (mantissa) -> CHANBW_M

MDMCFG3 = 0x93

Data rate (mantissa) -> DRATE_M

MDMCFG2 = 0x00

Modulation -> MOD_FORMAT[2:0]

Manchester enable -> MANCHESTER_EN

MDMCFG1 = 0x22

Channel spacing (exponent) -> CHANSPC_E

MDMCFG0 = 0xF8

Channel spacing (mantissa) -> CHANSPC_M

CHANNR = 0x00

Channel number -> CHAN

DEVIATN = 0x34

Frequency dev. (exponent) -> DEVIATN_E

Frequency dev. (mantissa) -> DEVIATN_M

FREND0 = 0x10

RF output power -> PA_POWER

FREND1 = 0x56

MCSM0 = 0x18

FOCCFG = 0x16

BSCFG = 0x6C

AGCCTRL2 = 0x43

AGCCTRL1 = 0x40

AGCCTRL0 = 0x91

FSCAL3 = 0xA9

FSCAL2 = 0x2A

FSCAL1 = 0x00

FSCAL0 = 0x11

FSTEST = 0x59

TEST2 = 0x81

TEST1 = 0x35

TEST0 = 0x0B

5. Conclusioni

5.1 *Obiettivi raggiunti*

L'obiettivo di questo lavoro è stato quello di realizzare un possibile ricetrasmettitore utilizzabile nel satellite Atmocube. L'iter progettuale è stato strutturato secondo un ordine preciso.
Nella prima parte dello studio si è cercato di individuare le specifiche fondamentali del sistema che il nostro ricetrasmettitore avrebbe dovuto rispettare. Condizioni necessarie sono la frequenza di lavoro, la data rate, la potenza del segnale e i bassi consumi, per citare i più importanti. Appurato questo, è stata fatta una ricerca delle componenti che potessero fare al caso nostro.
La seconda parte del nostro progetto è stata dedicata alla valutazione delle componenti scelte. Avendo necessità di buona trasmissione a bassa potenza con basso tasso d'errore, si è posta l'attenzione sulle possibili opzioni che ogni integrato può offrire. Vagliati e confrontati i chip, si è scelto quello che potesse garantire prestazioni migliori.
La terza parte è consistita nello studio approfondito della componente; l'obiettivo principale è stato considerare quali configurazioni potessero essere ideali ai nostri scopi e come utilizzarle. Essendo una comunicazione che necessita lo scambio di 10 pacchetti per contatto è obbligatorio cercare di garantire la migliore comunicazione possibile, cercando di sfruttare le possibilità che l'integrato fornisce.
La quarta parte è dedicata a test sull'integrato basati sullo studio fatto nelle fasi precedenti. Si cercato di simulare , coi mezzi reperibili, la trasmissione e la ricezione nell'atmosfera e le caratteristiche della modulazione del segnale RF utilizzato. L'esperienza in laboratorio è stata importante per chi, come me, non aveva avuto modo fino adesso di testare "con mano" un esperienza sul campo, interagendo con il sistema progettato in maniera diretta.
Essendo uno studio di progettazione non è necessariamente detto che il CC1100 venga utilizzato, soprattutto alla luce del fatto che ci sono diversi fattori nel sistema Atmocube che possono variare in base alle esigenze; questo è un po' il problema di un progetto a cui lavorano più persone in periodi differenti. Pertanto questo elaborato aspira ad essere il manuale per il ricetrasmettitore del satellite Atmocube, ma ,nel caso ci siano variazioni nel sistema, sarà sicuramente utile per dare un indirizzo di progettazione per questa particolare componente del satellite.

5.2 *Per il futuro*

Per quanto riguarda i punti ancora in sospeso:

- In principio era stato scelto l'Osservatorio Astronomico di Basovizza come stazione di terra; questa, però, aveva dei vincoli per quanto riguarda la modulazione che avrebbero causato qualche problema nella ricerca dell'integrato. Nel caso venga utilizzato il CC1100 è necessario allestire una stazione di ricezione a terra con un altro CC1100 (già in possesso di Elcon).
- E' necessario dimensionare più precisamente il circuito di connessione al PA, dato che era stato progettato per un altro chip, anche se con caratteristiche simili, e valutare la reale necessità di progettare la connessione al LNA, argomento in fase di studio.
- In merito a quanto detto nel paragrafo 3.8.4 è necessario colmare quel *gap* tra sensibilità riscontrata nelle misure e sensibilità necessaria calcolata nel *link budget*, ponderando quindi se progettare , come accennato nel punto precedente, un LNA in ingresso al ricetrasmettitore del satellite, oppure una combinazione delle due componenti appena citate, un PA al trasmettitore ed un LNA sul satellite; da valutare l'aumento del rumore, problema costante per quanto riguarda l'amplificazione dei segnali. Il *gap* potrebbe essere anche colmato diminuendo la *bit rate*, ma il tempo utile per la trasmissione dei dati potrebbe non essere sufficiente, visto che diminuendo la *bit rate* aumenta il tempo di trasmissione.
- Si dovrà valutare il ricetrasmettitore collegato a tutte le altre parti del sistema, per valutare l'effettiva compatibilità e la bontà della comunicazione una volta assemblato il sistema.
- Sarebbe interessante vagliare di quanto cambiano le prestazioni a seconda che vengano selezionate le opzioni di miglioramento della comunicazione (Fec e Inteleaving, Whitening) o meno.
- E' da testare in laboratorio la potenza in uscita e la sensibilità dell'integrato, collegandosi ad uno dei punti precedentemente detti riguardo l'utilità o meno della parte LNA in ingresso al ricetrasmettitore.
- Nel caso di introduzione dell'LNA valutare la potenza del rumore e la sua influenza sulla trasmissione

6. Appendice

Si riporta qui di seguito alcune procedure utili alla configurazione dell'integrato utilizzando il linguaggio informatico C++.
Nella sequenza sottostante vengono definiti i registri dell'integrato CC1100.

```
#define CC_GDO0          IO.PDR7.BIT.B6
#define CC_GDO2          IO.PDR6.BIT.B2

extern unsigned char Packet[64];
extern unsigned char Packet_Length;
extern unsigned char Packet_Status[2];
extern unsigned char CC_Status;

void TestPA(void);

void writeRFSettings(void);
void RFSendPacket(unsigned char *, unsigned char);
void RFSendPacketWithPreamble( unsigned char *, unsigned char );
unsigned char RFReceivePacket(void);

void CC_WriteReg(unsigned char, unsigned char);
void CC_WriteBurstReg(unsigned char, unsigned char*, unsigned char);
unsigned char CC_ReadReg(unsigned char);
void CC_ReadBurstReg(unsigned char, unsigned char *, unsigned char);
unsigned char CC_ReadStatus(unsigned char);
void CC_Strobe(unsigned char);
void CC_PowerupReset(void);
void CC_Reset( void );
void CC_PowerDown( void );
void CC_Enable( void );
void CC_Disable( void );
void CC_Check(void);
```

```
// Configuration Registers
#define TI_CCxxx0_IOCFG2        0x00        // GDO2 output pin configuration
#define TI_CCxxx0_IOCFG1        0x01        // GDO1 output pin configuration
#define TI_CCxxx0_IOCFG0        0x02        // GDO0 output pin configuration
#define TI_CCxxx0_FIFOTHR       0x03        // RX FIFO and TX FIFO thresholds
#define TI_CCxxx0_SYNC1         0x04        // Sync word, high byte
#define TI_CCxxx0_SYNC0         0x05        // Sync word, low byte
#define TI_CCxxx0_PKTLEN        0x06        // Packet length
#define TI_CCxxx0_PKTCTRL1      0x07        // Packet automation control
#define TI_CCxxx0_PKTCTRL0      0x08        // Packet automation control
#define TI_CCxxx0_ADDR          0x09        // Device address
#define TI_CCxxx0_CHANNR        0x0A        // Channel number
#define TI_CCxxx0_FSCTRL1       0x0B        // Frequency synthesizer control
#define TI_CCxxx0_FSCTRL0       0x0C        // Frequency synthesizer control
#define TI_CCxxx0_FREQ2         0x0D        // Frequency control word, high byte
#define TI_CCxxx0_FREQ1         0x0E        // Frequency control word, middle byte
#define TI_CCxxx0_FREQ0         0x0F        // Frequency control word, low byte
#define TI_CCxxx0_MDMCFG4       0x10        // Modem configuration
#define TI_CCxxx0_MDMCFG3       0x11        // Modem configuration
#define TI_CCxxx0_MDMCFG2       0x12        // Modem configuration
#define TI_CCxxx0_MDMCFG1       0x13        // Modem configuration
#define TI_CCxxx0_MDMCFG0       0x14        // Modem configuration
#define TI_CCxxx0_DEVIATN       0x15        // Modem deviation setting
#define TI_CCxxx0_MCSM2         0x16        // Main Radio Cntrl State Machine config
#define TI_CCxxx0_MCSM1         0x17        // Main Radio Cntrl State Machine config
#define TI_CCxxx0_MCSM0         0x18        // Main Radio Cntrl State Machine config
#define TI_CCxxx0_FOCCFG        0x19        // Frequency Offset Compensation config
#define TI_CCxxx0_BSCFG         0x1A        // Bit Synchronization configuration
#define TI_CCxxx0_AGCCTRL2      0x1B        // AGC control
#define TI_CCxxx0_AGCCTRL1      0x1C        // AGC control
#define TI_CCxxx0_AGCCTRL0      0x1D        // AGC control
#define TI_CCxxx0_WOREVT1       0x1E        // High byte Event 0 timeout
#define TI_CCxxx0_WOREVT0       0x1F        // Low byte Event 0 timeout
```

```
#define TI_CCxxx0_WORCTRL      0x20        // Wake On Radio control
#define TI_CCxxx0_FREND1       0x21        // Front end RX configuration
#define TI_CCxxx0_FREND0       0x22        // Front end TX configuration
#define TI_CCxxx0_FSCAL3       0x23        // Frequency synthesizer calibration
#define TI_CCxxx0_FSCAL2       0x24        // Frequency synthesizer calibration
#define TI_CCxxx0_FSCAL1       0x25        // Frequency synthesizer calibration
#define TI_CCxxx0_FSCAL0       0x26        // Frequency synthesizer calibration
#define TI_CCxxx0_RCCTRL1      0x27        // RC oscillator configuration
#define TI_CCxxx0_RCCTRL0      0x28        // RC oscillator configuration
#define TI_CCxxx0_FSTEST       0x29        // Frequency synthesizer cal control
#define TI_CCxxx0_PTEST        0x2A        // Production test
#define TI_CCxxx0_AGCTEST      0x2B        // AGC test
#define TI_CCxxx0_TEST2        0x2C        // Various test settings
#define TI_CCxxx0_TEST1        0x2D        // Various test settings
#define TI_CCxxx0_TEST0        0x2E        // Various test settings

// Strobe commands
#define TI_CCxxx0_SRES         0x30        // Reset chip.
#define TI_CCxxx0_SFSTXON      0x31        // Enable/calibrate freq synthesizer
#define TI_CCxxx0_SXOFF        0x32        // Turn off crystal oscillator.
#define TI_CCxxx0_SCAL         0x33        // Calibrate freq synthesizer & disable
#define TI_CCxxx0_SRX          0x34        // Enable RX.
#define TI_CCxxx0_STX          0x35        // Enable TX.
#define TI_CCxxx0_SIDLE        0x36        // Exit RX / TX
#define TI_CCxxx0_SAFC         0x37        // AFC adjustment of freq synthesizer
#define TI_CCxxx0_SWOR         0x38        // Start automatic RX polling sequence
#define TI_CCxxx0_SPWD         0x39        // Enter pwr down mode when CSn goes hi
#define TI_CCxxx0_SFRX         0x3A        // Flush the RX FIFO buffer.
#define TI_CCxxx0_SFTX         0x3B        // Flush the TX FIFO buffer.
#define TI_CCxxx0_SWORRST      0x3C        // Reset real time clock.
#define TI_CCxxx0_SNOP         0x3D        // No operation.

// Status registers
#define TI_CCxxx0_PARTNUM      0x30        // Part number
```

```
#define TI_CCxxx0_VERSION      0x31       // Current version number
#define TI_CCxxx0_FREQEST      0x32       // Frequency offset estimate
#define TI_CCxxx0_LQI          0x33       // Demodulator estimate for link quality
#define TI_CCxxx0_RSSI         0x34       // Received signal strength indication
#define TI_CCxxx0_MARCSTATE    0x35       // Control state machine state
#define TI_CCxxx0_WORTIME1     0x36       // High byte of WOR timer
#define TI_CCxxx0_WORTIME0     0x37       // Low byte of WOR timer
#define TI_CCxxx0_PKTSTATUS    0x38       // Current GDOx status and packet status
#define TI_CCxxx0_VCO_VC_DAC   0x39       // Current setting from PLL cal module
#define TI_CCxxx0_TXBYTES      0x3A       // Underflow and # of bytes in TXFIFO
#define TI_CCxxx0_RXBYTES      0x3B       // Overflow and # of bytes in RXFIFO
#define TI_CCxxx0_NUM_RXBYTES  0x7F       // Mask "# of bytes" field in _RXBYTES

// Other memory locations
#define TI_CCxxx0_PATABLE      0x3E
#define TI_CCxxx0_TXFIFO       0x3F
#define TI_CCxxx0_RXFIFO       0x3F

// Masks for appended status bytes
#define TI_CCxxx0_LQI_RX       0x01       // Position of LQI byte
#define TI_CCxxx0_CRC_OK       0x80       // Mask "CRC_OK" bit within LQI byte

// Definitions to support burst/single access:
#define TI_CCxxx0_WRITE_BURST  0x40
#define TI_CCxxx0_READ_SINGLE  0x80
#define TI_CCxxx0_READ_BURST   0xC0
```

La sequenza di procedure è introdotta da delle righe in cui si specificano i valori del sistema scelti.

```
#include "Compile.h"

unsigned char Packet[64];
unsigned char Packet_Length;
unsigned char Packet_Status[2];
```

```
unsigned char CC_Status;
unsigned char configTable[47];

// Product = CC1100
// Crystal accuracy = 40 ppm
// X-tal frequency = 26 MHz
// RF output power = 0 dBm
// RX filterbandwidth = 540.000000 kHz
// Deviation = 0.000000
// Return state:  Return to RX state upon leaving either TX or RX
// Datarate = 250.000000 kbps
// Modulation = (7) MSK
// Manchester enable = (0) Manchester disabled
// RF Frequency = 433.000000 MHz
// Channel spacing = 199.951172 kHz
// Channel number = 0
// Optimization = Sensitivity
// Sync mode = (3) 30/32 sync word bits detected
// Format of RX/TX data = (0) Normal mode, use FIFOs for RX and TX
// CRC operation = (1) CRC calculation in TX and CRC check in RX enabled
// Forward Error Correction = (0) FEC disabled
// Length configuration = (1) Variable length packets, packet length configured by the first received byte after sync word.
// Packetlength = 255
// Preamble count = (2)  4 bytes
// Append status = 1
// Address check = (0) No address check
// FIFO autoflush = 0
// Device address = 0
// GDO0 signal selection = ( 6) Asserts when sync word has been sent / received, and de-asserts at the end of the packet
// GDO2 signal selection = (11) Serial Clock
void writeRFSettings( void )
{
```

```
/*
        // Write register settings
        CC_WriteReg( TI_CCxxx0_IOCFG2,          0x0B ); // GDO2 output pin config.
        CC_WriteReg( TI_CCxxx0_IOCFG0,          0x06 ); // GDO0 output pin config.
        CC_WriteReg( TI_CCxxx0_PKTLEN,          0xFF ); // Packet length.
        CC_WriteReg( TI_CCxxx0_PKTCTRL1,    0x05 ); // Packet automation control.
        CC_WriteReg( TI_CCxxx0_PKTCTRL0,    0x05 ); // Packet automation control.
        CC_WriteReg( TI_CCxxx0_ADDR,        0x01 ); // Device address.
        CC_WriteReg( TI_CCxxx0_CHANNR,          0x00 ); // Channel number.
        CC_WriteReg( TI_CCxxx0_FSCTRL1,     0x0B ); // Freq synthesizer control.
        CC_WriteReg( TI_CCxxx0_FSCTRL0,     0x00 ); // Freq synthesizer control.
        CC_WriteReg( TI_CCxxx0_FREQ2,           0x10 ); // Freq control word, high byte
        CC_WriteReg( TI_CCxxx0_FREQ1,           0xA7 ); // Freq control word, mid byte.
        CC_WriteReg( TI_CCxxx0_FREQ0,           0x62 ); // Freq control word, low byte.
        CC_WriteReg( TI_CCxxx0_MDMCFG4,     0x2D ); // Modem configuration.
        CC_WriteReg( TI_CCxxx0_MDMCFG3,     0x3B ); // Modem configuration.
        CC_WriteReg( TI_CCxxx0_MDMCFG2,     0x73 ); // Modem configuration.
        CC_WriteReg( TI_CCxxx0_MDMCFG1,     0x22 ); // Modem configuration.
        CC_WriteReg( TI_CCxxx0_MDMCFG0,     0xF8 ); // Modem configuration.
        CC_WriteReg( TI_CCxxx0_DEVIATN,     0x00 ); // Modem dev (when FSK mod en)
        CC_WriteReg( TI_CCxxx0_MCSM1,           0x3F ); // MainRadio Cntrl State Ma-
chine
        CC_WriteReg( TI_CCxxx0_MCSM0,           0x18 ); // MainRadio Cntrl State Ma-
chine
        CC_WriteReg( TI_CCxxx0_FOCCFG,          0x1D ); // Freq Offset Compens. Config
        CC_WriteReg( TI_CCxxx0_BSCFG,           0x1C ); // Bit synchronization config.
        CC_WriteReg( TI_CCxxx0_AGCCTRL2,    0xC7 ); // AGC control.
        CC_WriteReg( TI_CCxxx0_AGCCTRL1,    0x00 ); // AGC control.
        CC_WriteReg( TI_CCxxx0_AGCCTRL0,    0xB2 ); // AGC control.
        CC_WriteReg( TI_CCxxx0_FREND1,          0xB6 ); // Front end RX configuration.
        CC_WriteReg( TI_CCxxx0_FREND0,          0x10 ); // Front end RX configuration.
        CC_WriteReg( TI_CCxxx0_FSCAL3,          0xEA ); // Frequency synthesizer cal.
        CC_WriteReg( TI_CCxxx0_FSCAL2,          0x0A ); // Frequency synthesizer cal.
        CC_WriteReg( TI_CCxxx0_FSCAL1,          0x00 ); // Frequency synthesizer cal.
```

```
        CC_WriteReg( TI_CCxxx0_FSCAL0,          0x11 ); // Frequency synthesizer cal.
*/
        CC_WriteReg( TI_CCxxx0_IOCFG2,configTable[TI_CCxxx0_IOCFG2] ); // GDO2 output
pin config.
        CC_WriteReg( TI_CCxxx0_IOCFG0,configTable[TI_CCxxx0_IOCFG0] ); // GDO0 output
pin config.
//      CC_WriteBurstReg( TI_CCxxx0_IOCFG2, (unsigned char *)&configTable[0], 3 );
        CC_WriteBurstReg(TI_CCxxx0_PKTLEN,(unsignechar*)&configTable[TI_CCxxx0_PK-
TLEN], 34 );
        CC_WriteReg( TI_CCxxx0_FSTEST,configTable[TI_CCxxx0_FSTEST]);//Frequency syn-
thesizer cal.
        CC_WriteBurstReg(TI_CCxxx0_TEST2,
(unsignedchar*)&configTable[TI_CCxxx0_TEST2], 3 );
}

unsigned char configTable[47] = {
        0x0B,  // IOCFG2      GDO2 output pin config.
        0x2E,  // IOCFG1      DEFAULT
        0x06,  // IOCFG0      GDO0 output pin config.
        0x07,  // FIFOTHR     DEFAULT
        0xD3,  // SYNC1       DEFAULT
        0x91,  // SYNC2       DEFAULT
        0xFF,  // PKTLEN      Packet length.
        0x05,  // PKTCTRL1 Packet automation control.
        0x05,  // PKTCTRL0 Packet automation control.
        0x01,  // ADDR                Device address.
        0x00,  // CHANNR      Channel number.
        0x0B,  // FSCTRL1     Freq synthesizer control.
        0x00,  // FSCTRL0     Freq synthesizer control.
        0x10,  // FREQ2       Freq control word, high byte
        0xA7,  // FREQ1       Freq control word, mid byte.
        0x62,  // FREQ0       Freq control word, low byte.
        0x2D,  // MDMCFG4 Modem configuration.
        0x3B,  // MDMCFG3 Modem configuration.
```

```
        0x73,   // MDMCFG2 Modem configuration.
        0x22,   // MDMCFG1 Modem configuration.
        0xF8,   // MDMCFG0 Modem configuration.
        0x00,   // DEVIATN   Modem dev (when FSK mod en)
        0x07,   // MCSM2     DEFAULT
        0x3F,   // MCSM1     MainRadio Cntrl State Machine
        0x18,   // MCSM0     MainRadio Cntrl State Machine
        0x1D,   // FOCCFG    Freq Offset Compens. Config
        0x1C,   // BSCFG     Bit synchronization config.
        0xC7,   // AGCCTRL2        AGC control.
        0x00,   // AGCCTRL1        AGC control.
        0xB2,   // AGCCTRL0        AGC control.
        0x87,   // WOREVT1 DEFAULT
        0x6B,   // WOREVT0 DEFAULT
        0xF8,   // WORCTRL DEFAULT
        0xB6,   // FREND1    Front end RX configuration.
        0x10,   // FREND0    Front end RX configuration.
        0xEA,   // FSCAL3    Frequency synthesizer cal.
        0x0A,   // FSCAL2    Frequency synthesizer cal.
        0x00,   // FSCAL1    Frequency synthesizer cal.
        0x11,   // FSCAL0    Frequency synthesizer cal.
        0x00,   // RCCTRL1   NO WRITE
        0x00,   // RCCTRL0   NO WRITE
        0x59,   // FSTEST    Frequency synthesizer cal.
        0x00,   // PTEST
        0x00,   // AGCTEST   NO WRITE
        0x88,   // TEST2     Various test settings.
        0x31,   // TEST1     Various test settings.
        0x0B    // TEST0     Various test settings.
};

// PATABLE (0 dBm output power)
#define paTableLen   1
extern unsigned char paTable[paTableLen] = { 0x60 };
```

//extern char paTable[] = { 0xC0 }; // +10 dBm

7. Bibliografia

[1] Mischa Schwartz. *Information Transmission, Modulation and Noise*. International Student Edition. McGRAW - HILL KOGAKUSHA,LTD, 1970.

[2] A cura di Mario Lenti. *Enciclopedia dell'ingegneria, volume 5*. Isedi (Istituto Editori Internazionale). Arnoldo Mondadori Editore, 1972.

[3] Sergio Benedetto, Ezio Biglieri, Valentino Castellani. *Teoria della trasmissione numerica*. Gruppo editoriale Jackson s.p.a., Milano, 1990.

[4] Masoud Salehi John G. Proakis. *Communication System Engineering*. Prentice Hall, Upper Sadle River, New Jersey, International second editino, 2002.

[5]Chipcon. CC1100 single chip low cost low power RF transceiver. http:// www.chipcon.com/files/CC110_Data_Sheet_1_1.pdf . Datasheet (Rev 1.1)

[6]Analog Device. ADF7020 high performance, ISM band, FSK/ASK Transceiver IC. Datasheet.

[7] Marco De Din. *Progetto preliminare di un sistema di gestione energetica per il satellite Atmocube*. Corso di laurea in Ingegneria Elettronica, A.A. 2004\2005.

[8] Alessandro Cuttin. *Progetto di sistema del satellite Atmocube architettura di comunicazione e architettura di sistema.* Corso di laurea in Ingegneria delle Telecomunicazioni, A.A 2005/2006.

[9] Walter Caharija. *Sistema di alimentazione del satellite Atmocube.* Corso di laurea in Ingegneria Elettronica, A.A. 2005/2006.

[10] Mauro Popesso. *Studio per la progettazione dell'amplificatore RF di potenza per il trasmettitore del satellite Atmocube mediante l'utilizzo del software di simulazione AWR.*. Corso di laurea in Ingegneria delle Telecomunicazioni, A.A. 2005/2006.

[11] www.chipcon.it

Ringraziamenti

In primo luogo devo ringraziare il prof. Fragiacomo che si è dimostrato in ogni occasione disponibile, comprensivo e generoso, non lesinando mai di dedicarmi il suo tempo prezioso.
Un ringraziamento all'Enteos che mi ha ospitato in questo anno di studio e progettazione.
Un sentito ringraziamento anche alla Elcon, in particolare nelle persone di Luciano Generali, Diego e Martina, che hanno concesso materiale, tempo e "dritte" preziose.
Ringrazio la mia famiglia che ha aspettato con ansia e fiducia la mia laurea.
Ringrazio Chicco che mi ha dato una grossa mano per impaginare al meglio questo elaborato.
Per ultimi, ma non ultimi, ringrazio i miei amici, che quando ero preso dallo sconforto mi hanno esortato a non mollare.
Ringrazio me stesso per essermi tolto questa soddisfazione.