

UNIVERSITÀ DEGLI STUDI DI TRIESTE
FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI ELETTRONICA, ELETTROTECNICA E INFORMATICA

Elaborato finale in

ELETTRONICA PER LE TELECOMUNICAZIONI

STUDIO PER LA PROGETTAZIONE
DEL SISTEMA DI CONTROLLO
A MICROPROCESSORE DEI
SOTTO SISTEMI
DEL SATELLITE ATMOCUBE

Laureando:
Emilio MONTAGNANA

Relatore:
Chiar.mo Prof. Mario FRAGIACOMO

ANNO ACCADEMICO 2005/2006

Indice

| | |
|--------------------------|-----------|
| Introduzione..... | ix |
|--------------------------|-----------|

Capitolo 1

| | |
|--|----------|
| Telemetria..... | 1 |
| 1.1 Generalità..... | 1 |
| 1.2 Telemetria di ATMOCUBE..... | 2 |
| 1.3 Telemetria base (TMB)..... | 4 |
| 1.4 Telemetria estesa (TME)..... | 7 |
| 1.5 Telemetria base addizionata (TMB Addizionata)..... | 14 |
| 1.6 Telemetria estesa addizionata (TME Addizionata)..... | 18 |
| 1.7 Considerazioni..... | 21 |

Capitolo 2

| | |
|--|-----------|
| Telecomando..... | 23 |
| 2.1 Telecomando di accesso..... | 23 |
| 2.2 Telecomando di configurazione..... | 24 |
| 2.3 Telecomando di richiesta..... | 26 |

Capitolo 3

| | |
|--|-----------|
| Struttura interna di ATMOCUBE (livello hardware)..... | 27 |
| 3.1 I μ C P18F452 e P18F252..... | 28 |
| 3.1.1 Moduli interni del μ C master utilizzati..... | 31 |
| 3.1.1.1 Modulo ADC (Analog Digital Converter)..... | 31 |
| 3.1.1.2 Modulo PWM (Pulse With Modulation)..... | 32 |
| 3.1.1.3 Modulo SPI (Serial Peripheral Interface)..... | 34 |
| 3.1.1.4 Timer 0, Timer 1 e Timer 2..... | 36 |
| 3.1.1.5 Modulo USART..... | 38 |
| 3.1.2 Scelta ed uso dei pin del μ C master | 39 |
| 3.2 Sezione di alimentazione..... | 43 |
| 3.3 Sezione di housekeeping..... | 46 |
| 3.4 Assetto..... | 53 |
| 3.5 Sezione di Trasmissione/Ricezione..... | 53 |
| 3.6 Sezione di data handling..... | 57 |

Capitolo 4

| | |
|---|-----------|
| Struttura interna di ATMOCUBE (livello software)..... | 65 |
| 4.1 Acquisizione e tracciatura dei dati di telemetria | 66 |
| 4.1.1 Procedura di acquisizione dall'ADC..... | 66 |
| 4.1.2 Tracciatura della memoria residua e dei dati strumentali..... | 71 |
| 4.1.3 Potenza media nelle trasmissioni e ricezioni..... | 72 |
| 4.1.4 Numero di accessi non autorizzati..... | 73 |
| 4.2 Interazione del μ C master..... | 73 |
| 4.2.1 Interazione μ C master \rightarrow Memoria di massa..... | 73 |
| 4.2.1.1 Scrittura dei dati..... | 74 |
| 4.2.1.2 Prelievo dei dati..... | 76 |
| 4.2.1.3 Cancellazione dei dati..... | 78 |
| 4.2.2 Interazione μ C master \rightarrow CC1100 (configurazione)..... | 79 |
| 4.2.3 Interazione μ C master \rightarrow CC1100 (trasmissione e ricezione dei dati).... | 81 |
| 4.2.4 Interazione μ C master \rightarrow DAC AD5300..... | 81 |
| 4.2.5 Interazione μ C master \rightarrow μ C slave..... | 83 |
| 4.3 Interazione del μ C slave..... | 84 |
| 4.3.1 Interazione μ C slave \rightarrow μ C master (invio dati)..... | 84 |
| 4.3.2 Interazione μ C slave \rightarrow GPS..... | 87 |
| 4.3.3 Interazione μ C slave \rightarrow Dosimetro..... | 88 |
| 4.3.4 Interazione μ C slave \rightarrow Magnetometro..... | 89 |

Capitolo 5

| | |
|---|-----------|
| Stima ed organizzazione della memoria di Telemetria..... | 93 |
| 5.1 Metodi di acquisizione e memorizzazione dei dati di telemetria..... | 94 |
| 5.1.1 Tecnica di memorizzazione "standard" | 94 |
| 5.1.2 Tecnica di memorizzazione differenziale..... | 95 |
| 5.2 Calcolo della memoria occupata per la singola orbita..... | 96 |
| 5.2.1 Tensioni e correnti..... | 96 |
| 5.2.2 Temperature..... | 100 |
| 5.2.3 Memoria residua..... | 101 |
| 5.2.4 Memoria totale dei dati strumentali..... | 101 |
| 5.2.5 Potenza media nelle trasmissioni e ricezioni..... | 102 |
| 5.2.6 Numero di accessi non autorizzati..... | 102 |
| 5.3 Calcolo dell'occupazione di memoria della telemetria..... | 102 |
| 5.3.1 Calcolo dell'occupazione di memoria della telemetria base..... | 103 |
| 5.3.2 Calcolo dell'occupazione di memoria della telemetria estesa..... | 106 |
| 5.3.2.1 Calcolo dell'occupazione di memoria per la singola orbita..... | 106 |
| 5.3.3 Calcolo dell'occupazione di memoria della TME..... | 107 |

| | | |
|---------|--|-----|
| 5.3.4 | Calcolo dell'occupazione di memoria delle telemetrie aggiuntive..... | 108 |
| 5.3.5 | Calcolo dell'occupazione di memoria della TME a cadenza ridotta..... | 110 |
| 5.4 | Organizzazione della memoria..... | 111 |
| 5.4.1 | Organizzazione della memoria di TMB..... | 112 |
| 5.4.1.1 | Organizzazione della memoria del primo pacchetto..... | 112 |
| 5.4.1.2 | Organizzazione della memoria dei pacchetti successivi al primo... | 113 |
| 5.4.2 | Organizzazione della memoria di TME..... | 114 |
| 5.4.2.1 | Organizzazione della memoria per la singola orbita..... | 115 |
| 5.4.2.2 | Organizzazione della memoria di TME..... | 118 |
| 5.5 | Occupazione di memoria dei dati strumentali..... | 118 |
| 5.6 | Considerazioni..... | 120 |

Capitolo 6

| | |
|-----------------------------------|-----|
| Applicazione del telecomando..... | 121 |
|-----------------------------------|-----|

Capitolo 7

| | |
|------------------|-----|
| Conclusioni..... | 129 |
|------------------|-----|

| | |
|-------------------|-----|
| Bibliografia..... | 133 |
|-------------------|-----|

| | |
|---------------------|-----|
| Ringraziamenti..... | 135 |
|---------------------|-----|

Indice delle figure

| | |
|--|----|
| Figura 1.1: Successione dei blocchi di dati di TME..... | 9 |
| Figura 1.2: Possibile successione dei blocchi di dati della settima orbita..... | 10 |
| Figura 1.3: Successione dei tipi di telemetria lungo l'orbita del satellite..... | 14 |
| Figura 1.4: Conclusione telemetria estesa addizionata..... | 21 |
| Figura 3.1: Possibili funzionalità delle piedinature dei μ C utilizzati..... | 29 |
| Figura 3.2: Struttura interna del μ C P18F452..... | 30 |
| Figura 3.3: Convertitore AD..... | 31 |
| Figura 3.4: Modulo PWM..... | 32 |
| Figura 3.5: Schema a blocchi del modulo SPI..... | 35 |
| Figura 3.6: Schema a blocchi del timer 0..... | 36 |
| Figura 3.7: Schema a blocchi timer 1..... | 37 |
| Figura 3.8: Schema a blocchi del timer 2..... | 37 |
| Figura 3.9: Porta USART in modalità trasmissione asincrona..... | 39 |
| Figura 3.10: Porta USART in modalità ricezione asincrona..... | 39 |
| Figura 3.11: Aspetto del μ C master | 40 |
| Figura 3.12: Assegnazione dei pin del μ C master | 41 |
| Figura 3.13: Collegamento dell'integrato MAX471..... | 44 |
| Figura 3.14: Inserimento del MAX471 per la misura della corrente della batteria..... | 45 |
| Figura 3.15: Connessioni per la telemetria di housekeeping..... | 47 |
| Figura 3.16: Sensore di temperatura LM19..... | 48 |
| Figura 3.17: PTC B59901..... | 49 |
| Figura 3.18: Quarzi da 32.768 Hz..... | 50 |
| Figura 3.19: Tuning fork (interno del quarzo di Figura 3.18)..... | 50 |
| Figura 3.20: Quarzo C-MAC XTAL003004..... | 50 |
| Figura 3.21: Contenuto del quarzo di Figura 50..... | 50 |
| Figura 3.22: Sezione di TX/RX..... | 54 |
| Figura 3.23: Procedure per abilitare la modalità TX..... | 56 |
| Figura 3.24: Procedure per abilitare la modalità RX..... | 56 |
| Figura 3.25: Sezione di data handling..... | 58 |
| Figura 3.26: Collegamento tra μ C slave e magnetometro..... | 60 |
| Figura 3.27: Alternativa del collegamento tra μ C slave e magnetometro..... | 61 |
| Figura 4.1: Registro ADCON0..... | 67 |
| Figura 4.2: Registro ADCON1..... | 68 |
| Figura 4.3: Giustificazione dei dati a sinistra e a destra..... | 69 |
| Figura 4.4: Procedura di acquisizione del canale..... | 70 |
| Figura 4.5: Procedura di acquisizione e memorizzazione | 71 |
| Figura 4.6: Memorizzazione di un singolo byte..... | 74 |
| Figura 4.7: Procedura di memorizzazione di N byte consecutivi..... | 75 |
| Figura 4.8: Segnali di memorizzazione dei un singolo byte..... | 76 |

| | |
|--|-----|
| Figura 4.9: Segnali di memorizzazione di due o più byte (max 256)..... | 76 |
| Figura 4.10: Prelievo di dati dalla memoria..... | 77 |
| Figura 4.11: Segnali di prelievo dati dalla memoria..... | 77 |
| Figura 4.12: Istruzione di page erase..... | 78 |
| Figura 4.13: Istruzione di sector erase..... | 78 |
| Figura 4.14: Istruzione di chip erase..... | 79 |
| Figura 4.15: Procedura di invio delle configurazioni di default per la modalità TX..... | 80 |
| Figura 4.16: Procedura di invio delle configurazioni di default per la modalità RX | 80 |
| Figura 4.17: Pacchetto DAC AD5300..... | 82 |
| Figura 4.18: Sequenza valida/non valida nella trasmissione al DAC AD5300..... | 82 |
| Figura 4.19: Procedura di colloqui tra μ C master e μ C slave..... | 83 |
| Figura 4.20: Richiesta ed invio di dati sul bus destinati al μ C master | 86 |
| Figura 4.21: Procedura di colloquio tra μ C slave e GPS..... | 87 |
| Figura 4.22: Procedura di colloquio tra μ C slave e Dosimetro..... | 88 |
| Figura 4.23: Struttura interna DG9415..... | 89 |
| Figura 4.24: Schema interno DAC ADS8343..... | 89 |
| Figura 4.25: Byte di controllo dell'ADS8343..... | 89 |
| Figura 4.26: Modalità internal clock..... | 90 |
| Figura 4.27: Modalità external clock: conversione a 24 clock..... | 91 |
| Figura 4.28: Modalità external clock: conversione a 32 clock..... | 91 |
| Figura 4.29: Procedura di acquisizione dei segnali dal convertitore AD ADS8343..... | 92 |
| Figura 5.1: Giustificazione dei dati a destra e a sinistra..... | 94 |
| Figura 5.2: Organizzazione della memoria del primo pacchetto di TMB..... | 113 |
| Figura 5.3: Organizzazione della memoria dei pacchetti di TMB successivi al primo. | 114 |
| Figura 5.4: Blocchi di dati di tipo A0, A1, B e C..... | 116 |
| Figura 5.5: Successione dei blocchi di dati A0, A1, B e C durante la singola orbita... | 117 |
| Figura 5.6: Organizzazione della memoria di TME..... | 118 |
| Figura 6.1: ICD2..... | 121 |
| Figura 6.2: Screenshot dell'MPLAB IDE..... | 122 |
| Figura 6.3: Demoboard della Microchip..... | 122 |
| Figura 6.4: Collegamento Demoboard..... | 127 |
| Figura 6.5: Screenshot del software TeraTerm..... | 127 |
| Figura 6.6: Screenshot del software Serial Port Monitor..... | 128 |

Indice delle tabelle

| | |
|--|-----|
| Tabella 1.1: Tempi di attesa, accesso e periodo orbitale..... | 2 |
| Tabella 1.2: Configurazione di default registro TMB_CONFIG..... | 5 |
| Tabella 1.3: Cadenze e numero di campioni di TME..... | 8 |
| Tabella 1.4: Configurazione di default del registro TME_CONFIG..... | 12 |
| Tabella 1.5: Cadenza alternativa per i dati di TME..... | 12 |
| Tabella 1.6: Configurazione di default per TIPO_TMB..... | 17 |
| Tabella 1.7: Configurazione di default per TIPO_TME..... | 20 |
| Tabella 3.1: Assegnazione dei compiti di ciascun pin del μ C..... | 42 |
| Tabella 4.1: Configurazione dei bit | 89 |
| Tabella 5.1: Calcolo dell'occupazione di memoria delle tensioni e delle correnti,..... | 98 |
| Tabella 5.2: Calcolo dell'occupazione di memoria delle tensioni e delle correnti,..... | 99 |
| Tabella 5.3: Calcolo dell'occupazione di memoria della temperatura, utilizzando | 100 |
| Tabella 5.4: Calcolo dell'occupazione di memoria della temperatura utilizzando, | 101 |
| Tabella 5.5: Occupazione di memoria del primo pacchetto di TMB..... | 104 |
| Tabella 5.6: Occupazione di memoria dei pacchetti di TMB successivi al primo..... | 105 |
| Tabella 5.7: Occupazione di memoria delle telemetrie aggiuntive..... | 109 |
| Tabella 5.8: Quantità di memoria prodotta dagli strumenti..... | 119 |

Introduzione

Nella fase di progettazione del satellite ATMOCUBE, intrapresa dal nostro dipartimento e da quello di Fisica, si è arrivati al punto tale da avere sottomano una serie di sotto sistemi dei quali si conoscono abbastanza bene, grazie a precedenti lavori di tesi, le caratteristiche e i comportamenti.

I blocchi funzionali di cui è stato portato (e si sta portando) avanti uno studio sono:

- lo **STADIO DI ALIMENTAZIONE** (da realizzare su circuito stampato)
- lo **STADIO DI TRASMISSIONE E RICEZIONE** (in fase di studio)
- lo **STADIO DI POTENZA**
- **L'ANTENNA**
- gli **STRUMENTI** (GPS, Magnetometro e Dosimetro - acquistati)

A questa lista di blocchi funzionali mancano all'appello: l'**AMPLIFICATORE A BASSO RUMORE** (LNA - *Low Noise Amplifier*), il **FILTRO DI TIPO PASSA BANDA** e il **SISTEMA PER L'ASSETTO** del satellite. Tali oggetti potrebbero essere oggetto di tesi per i prossimi mesi.

Di fronte a tale scenario ci si rende conto della necessità di far interagire tra loro i diversi blocchi. Tale compito verrà affidato ad un microcontrollore (μC), il quale non sarà considerato un sotto sistema bensì il nucleo dell'intero satellite; dal momento infatti in cui quest'ultimo verrà espulso dal razzo, che gli consentirà di intraprendere l'orbita attorno alla Terra, il μC prenderà il controllo di tutti i sotto sistemi precedentemente elencati e si occuperà della loro gestione, dell'acquisizione dei segnali dagli strumenti di bordo nonché il loro mantenimento e la loro trasmissione a Terra. Il μC inoltre, dovrà essere in grado di poter interagire anche con la stazione di Terra, la quale invierà segnali di *Telecomando*.

Tale elaborato di tesi vuole quindi descrivere e illustrare come il μC permetta e amministri il colloquio tra i sotto sistemi; saranno inoltre descritti il mantenimento del satellite durante le sue orbite, nonché i dati che dovranno essere inviati alla stazione di Terra.

Capitolo 1

Telemetria

1.1 Generalità

L'era delle missioni spaziali cominciò con il lancio del satellite *Sputnik* avvenuto nel 1957 ad opera dell'U.R.S.S.. Al suo interno era già implementato un sistema di trasmissione di dati verso la stazione di Terra, che caratterizzavano il satellite stesso, si trattava: di dati di *telemetria*. Questi ultimi costituiscono le informazioni provenienti dai diversi sensori e blocchi funzionali del satellite e hanno una notevole importanza nella missione spaziale in quanto, da questi dati, i progettisti possono avere la conferma degli studi effettuati sul satellite, nonché provvedere a migliorare determinati sotto sistemi. Oppure, nella situazione che nessun progettista spera che accada, riescono a determinare il motivo di qualche guasto o avaria. Questi dati servono quindi a monitorare lo stato dell'intero sistema dalla stazione di Terra.

Le grandezze di *telemetria*, ovverosia quelle entità che ci consentono di determinare lo stato del satellite, sono tenute sotto controllo dall'unità di governo del satellite, cioè (nel nostro caso) da un microcontrollore. Il medesimo, nel caso in cui tali grandezze non soddisfino precisi requisiti, ha il compito di agire secondo una determinata procedura, la quale ha l'obiettivo di risolvere il problema e riportare nei limiti richiesti la grandezza con comportamento anomalo.

Alcune grandezze di *telemetria* vengono tenute sotto controllo in modo continuativo, altre invece vengono controllate in modo occasionale.

Dal momento che lo scopo della *telemetria* è l'invio a Terra di tali grandezze e siccome non sempre il satellite è in continuo contatto con la stazione di Terra (come nel nostro caso), occorrerà memorizzare i dati in una memoria di dimensione opportuna (consona alle nostre esigenze). Ovviamente nella scelta della dimensione della memoria occorrerà tenere presente che più è elevato il numero delle grandezze che voglio comunicare a Terra, maggiore sarà lo spazio di memorizzazione richiesto.

1.2 Telemetria di ATMOCUBE

Come è stato osservato (analizzato) da lavori di tesi precedenti, il satellite ATMOCUBE compie un'orbita retrograda: ovvero compie un movimento in direzione opposta a quella normale dei corpi all'interno del sistema in esame. Se la Terra ruota quindi in senso antiorario (senso scelto per convenzione internazionale), il satellite ATMOCUBE si muove attorno alla Terra nel senso opposto, cioè in quello orario, con un'orbita (corrispondente all'angolo tra il piano orbitale e il piano equatoriale terrestre) inclinata di circa $98,2^\circ$. Note quindi le caratteristiche dell'orbita del satellite, conosciamo di conseguenza la sua evoluzione temporale; siamo quindi in grado di determinare (avvalendoci di un software specifico - STK Tool Kit) il *tempo di accesso* (la quantità di tempo in cui il satellite può comunicare con la stazione di Terra, in quanto sono in visibilità) e il *tempo di attesa* (il tempo che è necessario che si consumi affinché il satellite e la stazione di Terra siano di nuovo in visibilità).

Dalle simulazioni fatte si è visto che per un angolo di elevazione dell'antenna di Terra di 20° (caso peggiore che è stato considerato) abbiamo:

| | tempo minimo | tempo medio | tempo massimo |
|------------------|-----------------------|-------------------------------|-----------------------------|
| Tempo di accesso | 12 secondi | 236,68 secondi | 301 secondi |
| Tempo di attesa | 90 minuti (1h 30') | 595,3 minuti (9h 55' 18") | 777 minuti (12h 57' 00") |
| Periodo orbitale | | 5.684 secondi (1h 34' 44") | |

Tabella 1.1: Tempi di attesa, accesso e periodo orbitale

È stato anche indicato il periodo orbitale ovvero il tempo impiegato dal satellite per compiere un'orbita. Tale dato risulterà importante nel momento in cui occorrerà determinare il volume di dati occupati dalle grandezze di *telemetria* per ogni orbita. Già da adesso si può preannunciare che verrà preso, come periodo orbitale, un intervallo di tempo più breve in modo tale da lavorare con multipli di 45 secondi. Il nuovo valore di periodo orbitale sarà di 5.670 secondi.

Risulta inoltre evidente che i dati di *telemetria* raccolti dal microcontrollore contenuto nel satellite durante tutto il *tempo di attesa*, dovranno essere memorizzati in un'opportuna memoria e trasmessi (architettura *store & forward*) una volta che la stazione di Terra e il satellite sono in contatto tra loro: questo accade mediamente dopo 9h 55' 18", tempo necessario al satellite di compiere 7 orbite come riportato sotto.

$$\left\lceil \frac{\text{tempodiattesa}}{\text{tempoimpiegato per un'orbita}} \right\rceil = \left\lceil \frac{595,3 \times 60}{5.684} \right\rceil = \lceil 6,28 \rceil = 7 \text{ orbite}$$

Formula 1.1: Calcolo del numero di orbite di ATMOCUBE

Si è preso l'intero superiore in modo da mantenere un certo margine di sicurezza sui calcoli relativi all'occupazione di memoria.

Per quanto riguarda le grandezze di *telemetria*, si è deciso di considerare:

- l'OCCUPAZIONE DI MEMORIA DEI DATI STRUMENTALI
- le TENSIONI DI ALIMENTAZIONE
 - tensione dei pannelli solari (+ 7V)
 - tensione della batteria (+ 10,8V)
 - tensione + 5V
 - tensione + 3,3V
- il SENSO DELLA CORRENTE DELLA BATTERIA
- le CORRENTI DI ALIMENTAZIONE
 - corrente dei pannelli solari
 - corrente della batteria
- le TEMPERATURE
 - temperatura interna
 - temperatura esterna
 - temperatura del componente di maggior potenza
- la MEMORIA RESIDUA
- la POTENZA MEDIA DEL SEGNALE NELLE PRECEDENTI RICEZIONI
- la POTENZA MEDIA UTILIZZATA NELLE PRECEDENTI TRASMISSIONI
- i TENTATIVI DI ACCESSO NON AUTORIZZATI DURANTE LE ORBITE DEL SATELLITE

Le tensioni e correnti di alimentazione, senso della corrente della batteria, temperature e memoria residua, vengono classificate come *telemetria di housekeeping*, in quanto contribuiscono al corretto funzionamento della strumentazione di bordo del satellite o verosia al suo mantenimento. Le rimanenti grandezze invece (l'occupazione di memoria dei dati strumentali, la potenza media del segnale nelle precedenti ricezioni e trasmissioni e i tentativi di accesso non autorizzati durante le orbite del satellite) non sono essenziali per il funzionamento del satellite. Esse infatti si limitano a fornire informazioni aggiuntive.

D'ora in avanti con il termine *corrente della batteria*, si intenderà anche il dato di telemetria “senso della corrente della batteria”.

Tutti i dati di *telemetria*, di cui vogliamo tenere traccia, vengono memorizzati nella memoria di bordo. Tramite un opportuno *telecomando* inoltre (vedi Capitolo 2), è possibile decidere se questi devono essere cancellati al momento della loro trasmissione verso Terra, oppure una volta ricevuta la conferma della loro ricezione. Inizialmente, come impostazione di *default*, i dati saranno cancellati una volta ottenuta la conferma, da parte della stazione di Terra, dell'avvenuta ricezione (ricezione corretta); inoltre quest'ultima, al momento della prima trasmissione, che avverrà dopo il *tempo di attesa*, otterrà dal satellite un messaggio contenente i settaggi di tutte le impostazioni di *telemetria* del satellite modificabili tramite *telecomando*.

Le grandezze di *telemetria* vengono suddivise in:

- **TELEMETRIA BASE**
- **TELEMETRIA ESTESA**

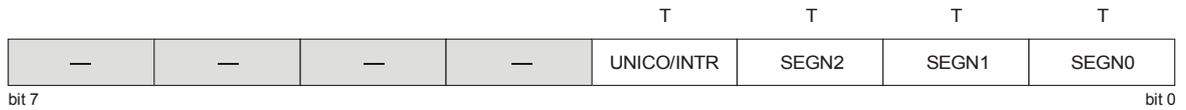
Ambedue le classificazioni fatte comprendono una variante: la *Telemetria Addizionata* (TMx Addizionata). Esse, come spiegato nei paragrafi “*Telemetria base addizionata (TMB Addizionata)*” e “*Telemetria estesa addizionata (TME Addizionata)*”, comprende dei dati di *telemetria* aggiuntivi.

1.3 *Telemetria* base (TMB)

La *telemetria* base, nel nostro caso, comprende le grandezze di *telemetria di housekeeping* e la memoria occupata dai dati strumentali nel momento in cui avviene la prima trasmissione tra satellite e stazione di Terra (cioè al termine del *tempo di attesa*). Fino al termine del *tempo di accesso*, tali dati di *telemetria*, verranno aggiornati e inviati a Terra ogni minuto. Il satellite quindi al primo contatto con la stazione di Terra, prima di inviare i dati strumentali, invierà un pacchetto contenente i dati di *telemetria base* in modo da far apprendere il suo stato alla stazione di Terra; prima di tali dati vi sarà una procedura di instaurazione della comunicazione e l'invio di particolari campi: “*Tipo_TMB*” (il cui significato è spiegato nel §1.5) seguito da “*TMB_CONFIG*”. Questi campi consistono in due registri interni al μ C settabili dalla stazione di Terra tramite *telecomando*. Nel momento in cui occorre quindi inviare la TMB, il μ C non fa altro che prelevare il contenuto di tali registri.

Tramite un opportuno *telecomando* è possibile disabilitare l'invio automatico dei dati di TMB in modo da risparmiare tempo ed energia (seppur in ridotta quantità) per la trasmissione dei dati strumentali, oppure alterare l'intervallo di tempo di segnalazione a Terra. In ogni caso il primo invio di TMB si conserva (non è possibile disabilitarlo tramite *telecomando*). Il *telecomando* in esame, andrà ad agire sul registro interno al μ C denominato TMB_CONFIG, strutturato nel seguente modo:

Campo (registro) 1: **TMB_CONFIG**



T = modificabile tramite *telecomando*

bit 7-4 non utilizzati

bit 3 **UNICO/INTR**: Invio unico o intervallato dei dati di telemetria base
 0 = il satellite invia una sola volta la *telemetria base* nel momento in cui satellite e stazione di Terra sono in contatto per la prima volta dopo il *tempo di attesa*
 1 = il satellite invia periodicamente la *telemetria base* dal momento in cui satellite e stazione di Terra sono in contatto per la prima volta dopo il *tempo di attesa*

bit 2-0 **SEGN2:SEGN0**: Intervallo di segnalazione
 000 = intervallo di segnalazione di 15 secondi
 001 = intervallo di segnalazione di 30 secondi
 010 = intervallo di segnalazione di 45 secondi
 011 = intervallo di segnalazione di 60 secondi
 100 = intervallo di segnalazione di 90 secondi
 101 = intervallo di segnalazione di 120 secondi
 110 = intervallo di segnalazione di 180 secondi
 111 = intervallo di segnalazione di 240 secondi

La configurazione di *default* del campo è la seguente:

| | |
|------------|----------|
| TMB_CONFIG | xxxx1011 |
|------------|----------|

Tabella 1.2: Configurazione di *default* registro TMB_CONFIG

x = bit irrilevante

Supponendo che sia in atto una trasmissione di dati alla stazione di Terra e che contemporaneamente avvenga la richiesta di comunicazione dei dati di TMB, il μC , prima dell'invio di questi ultimi, attenderà il termine del trasferimento in corso.

Il pacchetto di TMB conterrà:

- la **MEMORIA TOTALE DEI DATI STRUMENTALI** immagazzinati dall'ultimo invio di un pacchetto di TMB da parte del satellite fino all'inizio di una nuova comunicazione con la stazione di Terra.
- le **TENSIONI DI ALIMENTAZIONE**
 - tensione dei pannelli solari (+ 7V)
 - tensione della batteria (+ 10,8V)
 - tensione + 5V
 - tensione + 3,3V
- il **SENSO DELLA CORRENTE DELLA BATTERIA**
- le **CORRENTI DI ALIMENTAZIONE**
 - corrente dei pannelli solari
 - corrente della batteria
- le **TEMPERATURE**
 - temperatura interna
 - temperatura esterna
 - temperatura del componente di maggior potenza
- la **MEMORIA RESIDUA** (in quell'istante)

Si ribadisce il fatto che tutti questi dati devono essere misurati rigorosamente nel momento in cui avviene il primo contatto e negli intervalli successivi.

Da notare che ad ogni minuto (intervallo di *default* di invio della *telemetria base*), vengono inviate sempre le grandezze di *telemetria* relative allo spazio di memoria residuo e all'occupazione di memoria da parte dei dati strumentali, in modo da tenere sempre traccia della situazione a bordo del satellite. Durante tale intervallo infatti, parte della memoria sarà liberata in merito alla trasmissione a Terra: questi dati, permettono quindi di conoscere lo stato del satellite nel momento in cui inizia un nuovo *tempo di attesa*.

Si rimanda al §5.3.1 quanto concerne il calcolo del volume di dati prodotti dalla *telemetria base* e quindi della sua dimensione in byte.

1.4 Telemetria estesa (TME)

La *telemetria estesa* si differenzia da quella base per il fatto che i dati vengono acquisiti, a intervalli regolari (modificabili tramite *telecomando* agendo su TME_CONFIG<0>), durante l'avanzamento del satellite lungo la sua orbita attorno alla Terra e memorizzati in un'opportuna memoria secondo un preciso schema. La TME inoltre, a differenza della TMB, viene inviata a Terra solo tramite un opportuno *telecomando*. Senza far uso di quest'ultimo a ogni contatto satellite-stazione di Terra, è possibile usarne un altro che imposta l'invio automatico della TME subito dopo l'invio del primo pacchetto di *telemetria base* (TIPO_TMB<6>). Tale *telecomando* però, una volta inviato al satellite, avrà effetto al successivo contatto satellite-stazione.

Le grandezze che costituiscono la TME sono quelle di *housekeeping* con l'aggiunta della situazione della memoria (ho quindi le stesse grandezze di *telemetria* della TMB).

Abbiamo pertanto:

- la **MEMORIA TOTALE DEI DATI STRUMENTALI** immagazzinati al termine di ogni orbita
- le **TENSIONI DI ALIMENTAZIONE**
 - tensione dei pannelli solari (+ 7V)
 - tensione della batteria (+ 10,8V)
 - tensione + 5V
 - tensione + 3,3V
- il **SENSO DELLA CORRENTE DELLA BATTERIA**
- le **CORRENTI DI ALIMENTAZIONE**
 - corrente dei pannelli solari
 - corrente della batteria
- le **TEMPERATURE**
 - temperatura interna
 - temperatura esterna
 - temperatura del componente di maggior potenza
- la **MEMORIA RESIDUA** al termine di ogni orbita

Come è stato detto poco fa, lungo le orbite del satellite vengono effettuate le misurazioni dei dati di *telemetria* ad opera del microcontrollore a bordo del satellite. Ciascun dato di *telemetria* viene però prelevato con una certa cadenza, come indica la seguente tabella:

| Grandezza | Cadenza | n° misure per orbita | n° misure dopo 7 orbite |
|-------------------------------------|---------------|----------------------|-------------------------|
| Tensioni | 45 secondi | 126 | 882 |
| Correnti | 45 secondi | 126 | 882 |
| Senso della corrente della batteria | 45 secondi | 126 | 882 |
| Temperature | 90 secondi | 63 | 441 |
| Memoria residua | 5.670 secondi | 1 | 7 |
| Memoria dei dati strumentali | 5.670 secondi | 1 | 7 |

Tabella 1.3: Cadenze e numero di campioni di TME

dove il numero di misure per orbita è ottenuto nel seguente modo

$$\text{n° misure per orbita} = \frac{\text{periodo orbitale}}{\text{cadenza della misura}} = \frac{5.670}{\text{cadenza della misura}}$$

Formula 1.2: Calcolo del numero di misure di TME da effettuare in un'orbita

Per quanto riguarda la cadenza dei dati delle memorie, è stata fissata a 5.670 secondi anziché 5.684. Tale valore, come già preannunciato, è stato assunto anche come durata dell'orbita in modo da lavorare con numeri interi (in quest'ambito non hanno senso i numeri decimali: avere per esempio 126,3 campioni non ha senso, o sono 126 o 127). Per lo stesso motivo e per ragioni spiegate nel §1.2, sono state prese sette orbite: in questo modo si opera un'approssimazione per eccesso che si traduce in una sovrastima utile quando si dovrà eseguire il calcolo dello spazio di memoria occupato dai dati di TME.

Il microcontrollore che governa il satellite, una volta entrato in funzione, provvederà quindi ad acquisire i valori di tensione e corrente ogni 45 secondi, quelli di temperatura ogni 90 secondi e quelli relativi alla memoria ogni 5.670 secondi.

La *telemetria* estesa potrà quindi essere rappresentata dalla successione di tre distinti blocchi di dati:

- BLOCCO A** (si presenta all'istante iniziale e ogni 90 secondi): composto dai telemetrici di tensione, corrente e temperatura;
- BLOCCO B** (si presenta dopo 45 secondi dall'inizio dell'orbita e successivamente ogni 90 secondi): composto dai dati di tensione e corrente;
- BLOCCO C** (si presenta ogni 5.670 secondi): composto dai dati di tensione, corrente, memoria residua (al termine dell'orbita) e memoria occupata dai dati strumentali. Non viene considerata la temperatura in modo da mantenere l'alternanza dei blocchi A e B: il blocco C infatti è simile a quello B con l'aggiunta della situazione della memoria. Dopo il blocco C inizia una nuova orbita e quindi si ha di nuovo il blocco A (che contiene la temperatura).

Se rappresentassimo nel tempo la successione di tali blocchi avremmo il seguente andamento:

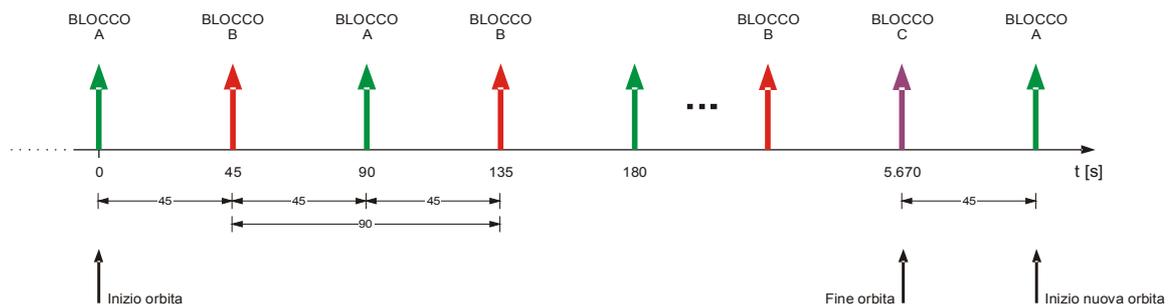


Figura 1.1: Successione dei blocchi di dati di TME

Come si nota dalla figura sopra, al 5.670° secondo dall'inizio dell'orbita compare il blocco di dati di tipo C; dopo quest'ultimo, cioè al 5.671° secondo dall'inizio dell'orbita precedente nonché al secondo 0 dell'orbita successiva, abbiamo di nuovo un blocco di dati di tipo A. Tale situazione si ripeterà per sei orbite (per la settima la situazione è differente). All'interno del μC vi sarà quindi un contatore che, una volta arrivato a 5.670, si azzererà avvisando lo stesso μC che l'orbita è terminata e che ne sta iniziando una nuova. Per nostra convenzione si stabilisce che l'inizio della prima orbita, ovverosia l'inizio del conteggio, avvenga nel momento in cui il satellite non riceve da Terra nessun segnale (il satellite in ogni caso invia segnali ma quando non ottiene il messaggio di avvenuta ricezione reputa la connessione con la stazione di Terra come persa) oppure nel momento in cui la stazione di Terra invia un *telecomando* avvisando il μC di avviare il contatore dei secondi (contatore con fondo scala pari a 5.670).

Il contatore dei secondi viene scandito da un particolare *timer* (Timer 1) presente nel μC . Questo *timer* è collegato esternamente ad un quarzo indipendente da quello del *clock* che scandisce i cicli macchina del μC . La descrizione di tale *timer* viene trattata nel §3.1.1.4.

Si constata inoltre che dalla Formula 1.1 si capisce che tra l'inizio e la fine del *tempo di attesa* il satellite compie circa 6,3 orbite. Pertanto la sequenza dei blocchi di dati (rappresentata in Figura 1.1) sarà uguale per sei orbite mentre per la settima avremo una situazione definibile come anomala, in quanto nel corso di quest'ultima il satellite potrà comunicare con la stazione di Terra, perciò il conteggio dei secondi del periodo dell'orbita terminerà prima di arrivare a 5.670. In questa situazione, cioè nel momento in cui il satellite sente che la stazione di Terra sta comunicando, il μC provvederà ad interrompere il conteggio e a terminare la sequenza dei blocchi di dati (rappresentata in Figura 1.2) con quello di tipo C. Potremmo quindi avere la situazione raffigurata sotto:

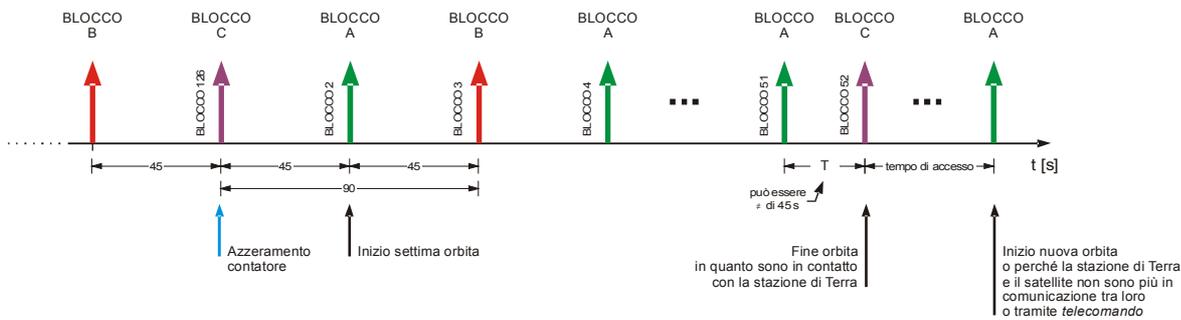


Figura 1.2: Possibile successione dei blocchi di dati della settima orbita

Come si può notare dalla figura sopra, nel momento in cui il μC accerta di essere in contatto con la stazione di Terra, viene inserito in modo immediato (senza quindi attendere l'intervallo di 45 secondi, anche perché il conteggio viene interrotto) un blocco di dati di tipo C. Dopo questo, viene attesa la scadenza del *tempo di accesso* (o il *telecomando* accennato precedentemente) e successivamente, ripartito il conteggio dei secondi dell'orbita, viene inserito il blocco di dati di tipo A.

Ogni volta che il μC provvede all'acquisizione dei dati di TME, questo deve anche preoccuparsi della loro memorizzazione ovvero di trasferirli dalla memoria di lavoro del μC (memoria RAM) alla memoria di massa del satellite.

Prima delle locazioni di memoria contenenti tutti i dati relativi alle sette orbite, il μC dovrà inserire un opportuno campo chiamato “*TIPO_TME*”, seguito da un altro denominato “*TME_CONFIG*”. Anche nel caso della TME, i campi in esame rappresentano dei registri del μC configurabili tramite *telecomando*. Per quanto riguarda il primo registro, esso è descritto nel §1.6. Per quanto concerne invece il secondo, esso mi consente di selezionare le grandezze di *telemetria* che verranno acquisite durante il *tempo di attesa*: in questo modo è possibile ridurre il numero di dati da trasmettere, nonché acquisire solo le grandezze di nostro interesse. Da notare il bit 0 che mi consente di aumentare (del doppio) l'intervallo di tempo tra due acquisizioni consecutive.

Il registro TME_CONFIG è così composto:

Campo (registro) 2: **TME_CONFIG**

| | | | | | | | |
|------------|------|-------|------------|-----------|-----------|-----------|----------|
| T | T | T | T | T | T | T | T |
| V_BATTERIA | V_5V | V_3V3 | I_BATTERIA | T_ESTERNA | T_INTERNA | T_POTENZA | EDIT_CAD |
| bit 7 | | | | | | | bit 0 |

T = modificabile tramite *telecomando*

- bit 7 **V_BATTERIA**: Tensione della batteria
0 = Non viene effettuata la misura durante il *tempo di attesa*
1 = Viene effettuata la misura durante il *tempo di attesa*
- bit 6 **V_5V**: Tensione 5V
0 = Non viene effettuata la misura durante il *tempo di attesa*
1 = Viene effettuata la misura durante il *tempo di attesa*
- bit 5 **V_3V3**: Tensione 3,3V
0 = Non viene effettuata la misura durante il *tempo di attesa*
1 = Viene effettuata la misura durante il *tempo di attesa*
- bit 4 **I_BATTERIA**: Corrente batteria
0 = Non viene effettuata la misura durante il *tempo di attesa*
1 = Viene effettuata la misura durante il *tempo di attesa*
- bit 3 **T_ESTERNA**: Temperatura esterna
0 = Non viene effettuata la misura durante il *tempo di attesa*
1 = Viene effettuata la misura durante il *tempo di attesa*
- bit 2 **T_INTERNA** Temperatura interna
0 = Non viene effettuata la misura durante il *tempo di attesa*
1 = Viene effettuata la misura durante il *tempo di attesa*
- bit 1 **T_POTENZA**: Temperatura del componente di potenza
0 = Non viene effettuata la misura durante il *tempo di attesa*
1 = Viene effettuata la misura durante il *tempo di attesa*
- bit 0 **EDIT_CAD**: Modifica la cadenza delle misure
0 = Le misure di TME vengono eseguite con la cadenza della tabella 1.3
1 = Le misure di TME vengono eseguite con la cadenza della tabella 1.5

la configurazione di *default* è la seguente:

| | |
|------------|----------|
| TME_CONFIG | 11111110 |
|------------|----------|

Tabella 1.4: Configurazione di *default* del registro TME_CONFIG

Per quanto riguarda l'ampliamento dei tempi di acquisizione, agendo quindi su TME_CONFIG<0>, abbiamo:

| Grandezza | Cadenza [s] | n° misure per orbita | n° misure dopo 7 orbite |
|-------------------------------------|-------------|----------------------|-------------------------|
| Tensioni | 90 | 63 | 441 |
| Correnti | 90 | 63 | 441 |
| Senso della corrente della batteria | 90 | 63 | 441 |
| Temperature | 180 | 31 | 219 |
| Memoria residua | 5.670 | 1 | 7 |
| Memoria dei dati strumentali | 5.670 | 1 | 7 |

Tabella 1.5: Cadenza alternativa per i dati di TME

(La terza colonna della tabella sopra è ottenuta applicando la Formula 1.2 e prendendo la parte intera)

La scelta di inserire la possibilità di disabilitare certe acquisizioni, è dettata dal fatto che, ad eccezione del periodo di vita iniziale del satellite, non sempre siamo interessati a conoscere come evolvono nel tempo le grandezze. Come si può notare dal registro TME_CONFIG, la tensione e la corrente dei pannelli non può essere mai disabilitata in quanto tali dati risultano essere di estrema utilità ai fini del funzionamento dell'intero sistema, e la loro conoscenza ci consente di stimare il tempo di vita del satellite. Dalla conoscenza infatti di queste grandezze, siamo in grado di apprendere se i pannelli stanno lavorando (o vengono fatti lavorare) nel modo corretto, oppure se sono stati danneggiati.

Optando per l'ampliamento dell'intervallo tra due acquisizioni consecutive (TME_CONFIG<0> = 1), non si ha più la corretta alternanza delle misure di temperatura (nel caso in cui vengano abilitate): il blocco di dati di tipo C, sarà infatti preceduto da quello di tipo A (ad eccezione della settima orbita in cui non si può effettuare una previsione).

Si stabilisce inoltre che il satellite può immagazzinare solamente due serie di TME consecutive. Nel caso in cui la stazione di Terra decida di non ricevere la TME relativa al *tempo di attesa* appena trascorso, essa è in grado di ottenerla nel seguente accesso al satellite, ma non in quello ulteriormente successivo, siccome la TME meno recente viene automaticamente cancellata al termine del contatto. La memoria riservata alla TME è paragonabile a una FIFO a due celle.

Infine la funzione fondamentale a cui deve provvedere il μ C nell'istante in cui acquisisce i primi dati di *telemetria estesa* di ciascuna orbita, è la richiesta al GPS della data e dell'ora. Il GPS utilizzato, una volta interpellato con il comando appropriato, fornisce svariati tipi di informazione di cui però solo alcune sono utili ai nostri scopi: il μ C ha il compito di discriminare i dati in risposta utili da quelli futili. Il μ C crea quindi un campo contenente i dati temporali del GPS e lo colloca prima di ciascun campione di TME, quindi prima dei blocchi di dati di tipo A, B e C.

Rappresentando quindi graficamente il comportamento del satellite per quanto riguarda la *telemetria* nell'arco di tempo che va dall'inizio del *tempo di accesso* a quello successivo, abbiamo:

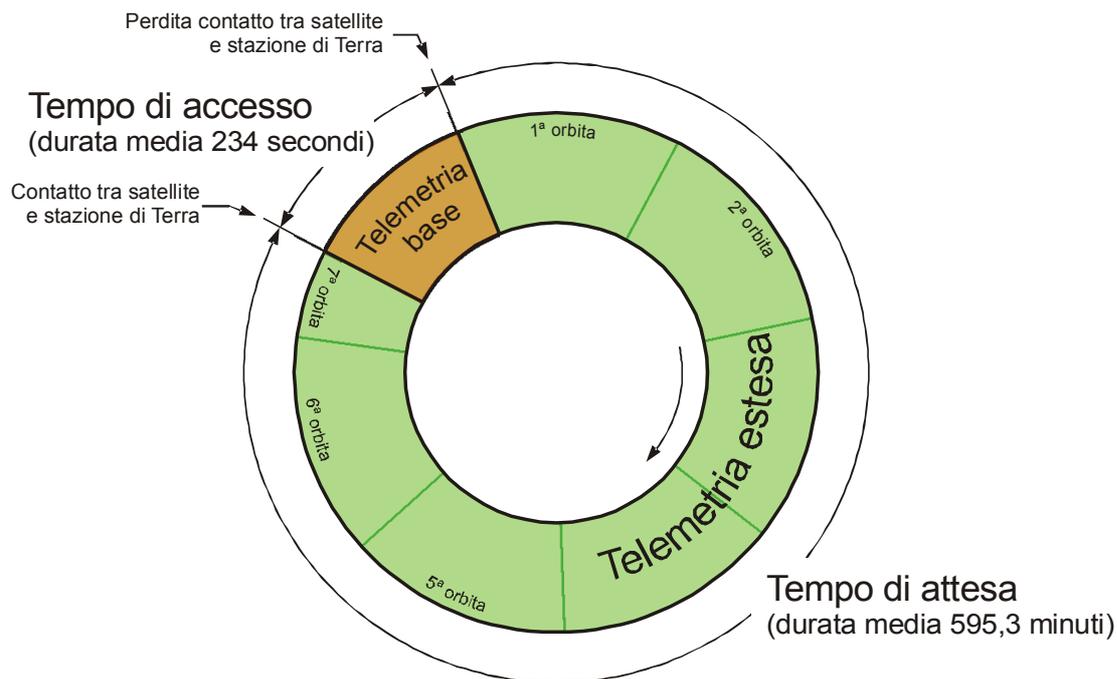


Figura 1.3: Successione dei tipi di telemetria lungo l'orbita del satellite

1.5 Telemetria base addizionata (TMB Addizionata)

Questa tipologia di *telemetria*, costituisce una variante a quella base. Essa infatti si differenzia per il fatto che nel momento in cui avviene il contatto satellite-stazione di Terra, viene inviata la *telemetria base* con l'aggiunta, al termine dei dati di TMB, di ulteriori campi (è possibile scegliere quali, tramite un *telecomando* inviato nell'accesso precedente). Dalla lista delle *grandezze di telemetria*, riportata nel §1.2, si può notare infatti che alcune voci non compaiono mai in TMB e TME.

I campi addizionali sono:

- la POTENZA MEDIA DEL SEGNALE NELLE PRECEDENTI RICEZIONI
- la POTENZA MEDIA UTILIZZATA NELLE PRECEDENTI TRASMISSIONI
- i TENTATIVI DI ACCESSO NON AUTORIZZATI DURANTE LE ORBITE DEL SATELLITE

I primi due campi vengono ottenuti acquisendo, durante il *tempo di accesso*, la potenza del segnale ricevuto/inviato per ciascun pacchetto e operando poi una media. La terza grandezza invece è un contatore che conterrà il numero di tentativi di accesso al satellite non autorizzati durante il *tempo di attesa* appena terminato (nel primo contatto tra stazione di Terra e satellite occorrerà infatti inviare un codice opportuno in modo tale che il satellite capisca che siamo noi i suoi veri gestori e non qualche altro utente). Tale contatore inoltre verrà azzerato all'inizio del *tempo di attesa* successivo.

Poiché per *default* la *telemetria* è di tipo base, il *telecomando* per attivare la *TMB Addizionata*, per quanto riguarda le misurazioni delle potenze dei segnali, ha effetto solo due accessi dopo. Infatti durante l'accesso in cui invio il *telecomando* si effettua solo un settaggio delle impostazioni, nell'accesso successivo avviene l'acquisizione e nel successivo ancora l'invio. Se invece all'interno del *telecomando* per attivare la *TMB Addizionata* è anche settato l'invio del *numero di accessi non autorizzati*, lo stesso *telecomando* avrà effetto nell'accesso successivo ovvero i dati relativi a tale grandezza saranno inviati a Terra. L'acquisizione del *numero degli accessi non autorizzati* avviene infatti durante il *tempo di attesa* e non durante il *tempo di accesso*.

Come detto nel §1.3, prima dell'invio dei dati di *telemetria* saranno posti due campi ("*TIPO_TMB*" e "*TMB_CONFIG*") che mi forniranno informazioni sullo stato dei settaggi e sui dati che saranno inviati subito dopo. Tali campi costituiscono due registri all'interno del μC i quali sono modificabili tramite *telecomando*.

La struttura del registro TIPO_TMB è la seguente:

Campo (registro) 3: **TIPO_TMB**

| | | | | | | | |
|---------|----------|------------|------------|------------|-----------|-----------|-----------|
| T | T | T | T | T | | | |
| TMA_LOC | TME_AUTO | SETT_TMBA2 | SETT_TMBA1 | SETT_TMBA0 | TMBA_DAT2 | TMBA_DAT1 | TMBA_DAT0 |
| bit 7 | | | | | | | bit 0 |

T = modificabile tramite *telecomando*

- bit 7** **TMA_LOC:** Localizzazione della telemetria addizionale
 0 = Telemetria addizionale localizzata nella telemetria base
 1 = Telemetria addizionale localizzata nella telemetria estesa
- bit 6** **TME_AUTO:** Automazione della telemetria estesa
 0 = La telemetria estesa viene inviata su richiesta tramite telecomando
 1 = La telemetria estesa viene inviata automaticamente al termine della telemetria base
- bit 5** **SETT_TMBA2:** Settaggi telemetria base addizionata (potenza media in ricezione)
se TMA_LOC = 0
 0 = il μ C non effettua la misura della potenza media in ricezione nel successivo *tempo di accesso*
 1 = il μ C effettua la misura della potenza media in ricezione nel successivo *tempo di accesso*
se TMA_LOC = 1
 configurazione dei bit influente
- bit 4** **SETT_TMBA1:** Settaggi telemetria base addizionata (potenza media in trasmissione)
se TMA_LOC = 0
 0 = il μ C non effettua la misura della potenza media in trasmissione nel successivo *tempo di accesso*
 1 = il μ C effettua la misura della potenza media in trasmissione nel successivo *tempo di accesso*
se TMA_LOC = 1
 configurazione dei bit influente
- bit 3** **SETT_TMBA0:** Settaggi telemetria base addizionata (numero accessi non autorizzati)
se TMA_LOC = 0
 0 = il μ C non effettua la misura del numero di accessi non autorizzati nel successivo *tempo di attesa*
 1 = il μ C effettua la misura del numero di accessi non autorizzati nel successivo *tempo di attesa*
se TMA_LOC = 1
 configurazione dei bit influente
- bit 2** **TMBA_DAT2:** Dati telemetria base addizionata (potenza media in ricezione)
 0 = nella telemetria base che segue non sono contenute le misure della potenza media in ricezione del precedente *tempo di accesso*
 1 = nella telemetria base che segue sono contenute le misure della potenza media in ricezione del precedente *tempo di accesso*

- bit 1 **TMBA_DAT1**: Dati telemetria base addizionata (potenza media in trasmissione)
 0 = nella telemetria base che segue non sono contenute le misure della potenza media in trasmissione del precedente *tempo di accesso*
 1 = nella telemetria base che segue sono contenute le misure della potenza media in trasmissione del precedente *tempo di accesso*
- bit 0 **TMBA_DAT0**: Dati telemetria base addizionata (numero di accessi non autorizzati)
 0 = nella telemetria base che segue non sono contenute le misure del numero di accessi non autorizzati del precedente *tempo di attesa*
 1 = nella telemetria base che segue sono contenute le misure del numero di accessi non autorizzati del precedente *tempo di attesa*

Da notare che il bit 7 (TMA_LOC) serve ad informare la stazione di Terra se il μC è settato per collocare i dati addizionali nella TMB o nella TME. Nel caso in cui sia posto a 0, i campi SET_TMBA2:SET_TMBA0 non sono presi in considerazione in quanto la *telemetria* addizionale è di tipo esteso; nel registro TIPO_TME tali rispettivi bit potranno invece essere settati. L'utilità dell'invio del bit 7, deriva anche dal fatto che, ad ogni contatto, la TMB viene incondizionatamente inviata mentre la TME, al cui interno è presente il registro TIPO_TME che mi informa della presenza dei dati addizionali, è inviata su richiesta (dipende dal settaggio di TIPO_TME<6>).

La configurazione di *default* del registro TIPO_TMB, che corrisponde alla *telemetria base*, è la seguente:

| | |
|----------|----------|
| TIPO_TMB | 10000000 |
|----------|----------|

Tabella 1.6: Configurazione di *default* per *TIPO_TMB*

x = bit irrilevante

Nel momento in cui vengono ricevuti i *telecomandi* atti ad alterare le impostazioni di *default*, il μC provvederà ad modificare le sue procedure interne in modo da soddisfare le richieste della stazione di Terra. Se per esempio vogliamo conoscere il *numero di accessi non autorizzati* durante il *tempo di attesa*, il μC , una volta terminato il *tempo di accesso*, avvierà una sua procedura interna che consentirà di tenere traccia di tali accessi; allo stesso modo, se volessimo conoscere la potenza media ricevuta nel prossimo *tempo di accesso*, dopo l'invio dell'opportuno *telecomando*, il μC agirà in modo tale da effettuare le misure della potenza ricevuta e provvedere al calcolo del suo valore medio.

Per disattivare la TMB Addizionata occorre inviare un *telecomando* che reimposti gli opportuni bit. Il medesimo ha effetto immediato: ovvero non occorre aspettare il *tempo di accesso* successivo affinché produca i propri effetti.

In ogni caso i dati di TMB Addizionata vengono inviati solo una volta per *tempo di accesso*, anche perché non avrebbe senso inviare dati sempre uguali (il calcolo delle potenze necessita di tutti i valori fino al termine del *tempo di accesso* e il conteggio degli accessi non autorizzati viene fatto durante il *tempo di attesa*).

1.6 Telemetria estesa addizionata (TME Addizionata)

Tale *telemetria* costituisce una variante della *telemetria estesa* (descritta nel §1.4). Anche in questo caso, come in quello della TMB Addizionata, vengono aggiunti ulteriori dati di *telemetria* al termine di quelli di TME. Tali dati sono:

- la POTENZA MEDIA DEL SEGNALE NELLE PRECEDENTI RICEZIONI
- la POTENZA MEDIA UTILIZZATA NELLE PRECEDENTI TRASMISSIONI
- i TENTATIVI DI ACCESSO NON AUTORIZZATI DURANTE LE ORBITE DEL SATELLITE

Affinché tali grandezze vengano inserite, è necessario l'invio di un opportuno *telecomando* che setti un opportuno registro del μC . Infatti per *default* la TME Addizionata è disabilitata. Tale *telecomando* inoltre può stabilire quali grandezze aggiuntive inserire e quali no. Tutte e tre le grandezze vengono inserite al termine della TME.

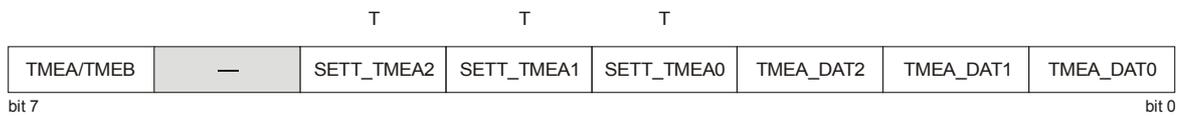
Si è visto che agendo sul bit TIPO_TMB<7>, si impone al μC di memorizzare i dati addizionali o al termine della TMB o al termine della TME. Nel caso quindi in cui il bit 7 del campo è posto a 1, al primo contatto tra satellite e stazione di Terra, il μC provvederà a terminare la *telemetria estesa* con l'immagazzinamento dei dati addizionali calcolati (se nell'accesso precedente è stato inviato il *telecomando* per memorizzare le potenze di trasmissione/ricezione, occorrerà attendere un nuovo accesso al satellite affinché questo abbia effetto, in quanto occorre ancora acquisire e mediare i dati).

Come nella TMB Addizionata, le prime due grandezze di *telemetria* vengono calcolati tramite le acquisizioni (della potenza del segnale ricevuto/inviato per ciascun pacchetto) effettuate nel *tempo di accesso*, e operando una media. La terza grandezza invece è un contatore che conterrà il numero di tentativi di accesso al satellite non autorizzati durante il *tempo di attesa* appena terminato. Tale contatore inoltre verrà azzerato all'inizio del *tempo di attesa* successivo.

Siccome proprio per *default* abbiamo la TME, il *telecomando* per attivare la *TME Addizionata* ha effetto solo due accessi dopo, per quanto riguarda le misurazioni delle potenze dei segnali, ed effetto immediato per quanto riguarda il numero di accessi non autorizzati. Durante il primo *tempo di accesso* infatti, in cui opero l'invio del *telecomando*, ho solo un settaggio delle impostazioni, nel *tempo di accesso* successivo ottengo l'acquisizione dei livelli di potenza del segnale ricevuto/trasmesso e nel successivo ancora l'invio.

La struttura di tale registro è la seguente:

Campo (registro) 4: **TIPO_TME**



T = modificabile tramite *telecomando*

- bit 7 **TMEA/TMEB:** Serie di telemetria estesa
 0 = Telemetria estesa A
 1 = Telemetria estesa B

- bit 6 non utilizzato

- bit 5 **SETT_TMEA2:** Settaggi telemetria estesa addizionata (potenza media in ricezione)
 0 = il μ C non effettua la misura della potenza media in ricezione nel successivo *tempo di accesso*
 1 = il μ C effettua la misura della potenza media in ricezione nel successivo *tempo di accesso*

- bit 4 **SETT_TMEA1:** Settaggi telemetria estesa addizionata (potenza media in trasmissione)
 0 = il μ C non effettua la misura della potenza media in trasmissione nel successivo *tempo di accesso*
 1 = il μ C effettua la misura della potenza media in trasmissione nel successivo *tempo di accesso*

- bit 3 **SETT_TMEA0:** Settaggi telemetria estesa addizionata (numero accessi non autorizzati)
 0 = il μ C non effettua la misura del numero di accessi non autorizzati nel successivo *tempo di attesa*
 1 = il μ C effettua la misura del numero di accessi non autorizzati nel successivo *tempo di attesa*

- bit 2 **TMEA_DAT2:** Dati telemetria estesa addizionata (potenza media in ricezione)
 0 = nella telemetria che segue non sono contenute le misure della potenza media in ricezione del precedente *tempo di accesso*
 1 = nella telemetria che segue sono contenute le misure della potenza media in ricezione del precedente *tempo di accesso*

- bit 1 **TMEA_DAT1:** Dati telemetria estesa addizionata (potenza media in trasmissione)
 0 = nella telemetria che segue non sono contenute le misure della potenza media in trasmissione del precedente *tempo di accesso*
 1 = nella telemetria che segue sono contenute le misure della potenza media in trasmissione del precedente *tempo di accesso*

| | |
|-------|---|
| bit 0 | <p>TMEA_DAT0: Dati telemetria estesa addizionata (numero di accessi non autorizzati)</p> <p>0 = nella telemetria che segue non sono contenute le misure del numero di accessi non autorizzati del precedente <i>tempo di attesa</i></p> <p>1 = nella telemetria che segue sono contenute le misure del numero di accessi non autorizzati del precedente <i>tempo di attesa</i></p> |
|-------|---|

Da notare il bit 7: esso serve per contraddistinguere la serie di *telemetria estesa*. È stato detto infatti che a bordo del satellite vengono conservate, nel caso in cui non vengano scaricate alla stazione di Terra (tramite *telecomando*), al massimo due telemetrie estese consecutive. Se quindi, ad ogni contatto satellite-stazione, decido di ricevere la TME rispettiva al *tempo di attesa* appena terminato, dovrei ricevere il campo TIPO_TME con il bit 7 alternativamente 1 e 0.

La configurazione di *default* del registro TIPO_TME, che corrisponde alla *telemetria estesa*, è la seguente:

| | |
|----------|----------|
| TIPO_TME | xx000xxx |
|----------|----------|

Tabella 1.7: Configurazione di *default* per TIPO_TME

x = bit irrilevante

Nel momento in cui viene ricevuto il *telecomando* atto ad alterare le impostazioni di *default*, il μC provvederà ad alterare le sue procedure interne in modo da soddisfare le richieste della stazione di Terra.

Per disattivare la TME Addizionata occorre inviare un *telecomando* che reimposti gli opportuni bit nella posizione di *default*. Tale *telecomando* ha effetto immediato.

Come detto nel §1.4, prima dell'immagazzinamento dei dati di *telemetria estesa* (ogni inizio del *tempo di attesa*) sarà posto un registro (TIPO_TME) che mi fornirà sia lo stato dei settaggi sia il tipo di dati che saranno contenuti in fondo. Se quindi la TME Addizionata è attiva ed è in corso d'opera la settima orbita, il blocco di dati di tipo C sarà seguito dai dati addizionali.

Rappresentando graficamente la situazione abbiamo:

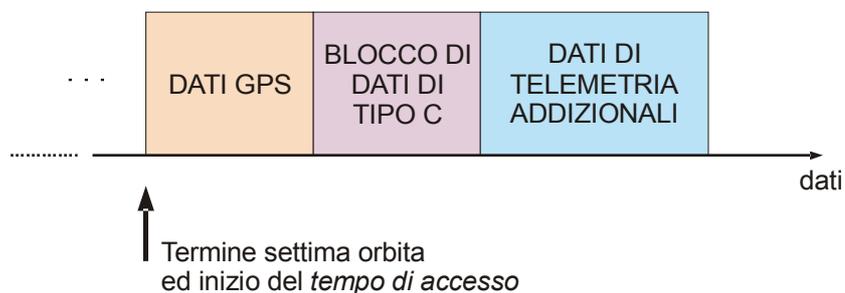


Figura 1.4: Conclusione telemetria estesa addizionata

1.7 Considerazioni

Si sottolinea il fatto che, nel momento in cui il satellite è in comunicazione con la stazione di Terra, non avviene alcun processo di *telemetria estesa*, ma solo di *telemetria base*. Quest'ultima, inoltre, nel momento in cui viene inviata, contiene sempre tutti i dati di *telemetria*, comprensivi o meno di quelli di TMB Addizionata a seconda dai settaggi effettuati all'accesso precedente.

Per quanto riguarda il contatore per tenere traccia del numero di accessi non autorizzati, si può dire fin da ora che la sua occupazione di memoria è di 1 byte, dal momento che si presume che non avvengano molti accessi irregolari.

Capitolo 2

Telecomando

Ai fini di garantire un controllo sul satellite in orbita, è stato introdotto il *telecomando*. Esso consiste nell'invio dalla stazione di Terra di particolari messaggi atti ad alterare le impostazioni del satellite, nonché a richiedere l'invio delle sue impostazioni e dei dati acquisiti lungo la sua orbita.

Verranno ora descritte le tre tipologie di *telecomando* utilizzate per ATMOCUBE senza entrare nel dettaglio sulla struttura del pacchetto che lo contiene.

2.1 *Telecomando* di accesso

Affinché avvenga la ricezione del *telecomando* da parte del satellite, è necessaria la visibilità di quest'ultimo con la stazione di Terra: ciò avviene al termine del *tempo di attesa* e per tutta la durata del *tempo di accesso*. Come visto dalla Tabella 1.1, tali tempi non sono costanti, bensì variano da un valore massimo di 301 secondi ad un minimo di 12 secondi. Per il calcolo di tali tempi è stato utilizzato un software specifico il quale ha richiesto una certa potenza di calcolo, e pertanto un certo consumo di energia. Siccome a bordo del satellite l'energia è un bene prezioso, non è possibile implementare l'algoritmo del calcolo previsionale del *tempo di accesso*. Per tale motivo, il satellite invierà verso Terra un particolare messaggio (*beacon*) che lo identificherà in modo univoco dagli altri satelliti. Immediatamente dopo esso provvederà ad attivare il ricevitore e a “mettersi in ascolto” dei segnali provenienti dalla stazione di Terra. Quest'ultima, conoscendo l'evoluzione dei *tempi di accesso*, saprà quando il satellite entrerà in visibilità e quindi quando trasmettere un particolare segnale (*telecomando di accesso*) facente parte della procedura di instaurazione della connessione. Parte del pacchetto inviato conterrà le credenziali per autenticarsi

ovverosia un codice, conosciuto solo dalla stazione di Terra (che è l'effettiva padrona del satellite) e dal satellite. Tale codice consente di accedere alle funzionalità del satellite. Quest'ultimo, una volta verificata la correttezza del codice inviato (il codice è presente all'interno del campo dati del pacchetto inviato secondo lo standard AX.25) provvederà ad inviare la conferma di avvenuta autenticazione e procederà all'invio della TMB, non prima però di aver inviato lo stato di tutte le sue impostazioni interne (ad eccezione di quelle relative alla TMB, vedi §1.3).

2.2 Telecomando di configurazione

Una volta stabilita la connessione tra satellite e stazione di Terra, sarà possibile trasmettere i segnali contenenti i *telecomandi* atti ad alterare le impostazioni di bordo.

Ciascun *telecomando*, una volta inviato, andrà ad alterare lo stato logico di determinati registri interni al μC che gestisce l'intero sistema e quindi ad alterare le sue procedure. Al momento dell'esecuzione di queste ultime infatti il μC dovrà testare il contenuto di ciascun registro che la coinvolge e agire di conseguenza. Ogni *telecomando* potrà sia abilitare certe funzione che disabilitarle.

Nel capitolo relativo alla *telemetria* ne sono stati accennati alcuni: si procede ora a riassumerli e a descriverli, senza esporre la loro struttura e disposizione all'interno del pacchetto inviato.

➤ **Telecomando per il settaggio del registro TMB_CONFIG**

Consente di alterare il contenuto del registro TMB_CONFIG descritto nel §1.3. L'invio di questo *telecomando* provocherà la sostituzione della configurazione presente nel registro del μC .

Tale *telecomando* consente in particolare di abilitare/disabilitare l'invio automatico dei dati di TMB durante il *tempo di accesso*; con il medesimo è possibile inoltre regolare il periodo di segnalazione da un minimo di 15 secondi ad un massimo di 240 secondi.

➤ **Telecomando per il settaggio del registro TIPO_TMB**

Consente di alterare il contenuto del registro TIPO_TMB descritto nel §1.5. L'invio di questo *telecomando* provocherà la sostituzione della configurazione presente nel registro del μC .

Tale *telecomando* consente in particolare di stabilire dove posizionare la *telemetria* addizionale (se al termine della TMB o della TME), nonché decidere quali grandezze di tale *telemetria* (base) abilitare.

Il settaggio di un particolare bit del registro in esame (TMA_AUTO), consente di abilitare/disabilitare l'invio automatico delle grandezze di TME immediatamente dopo l'invio di quelle di TMB. Tale procedura ha la “precedenza” su quella di invio automatico di TMB.

➤ **Telecomando per il settaggio del registro TME_CONFIG**

Consente di alterare il contenuto del registro TME_CONFIG descritto nel §1.4. L'invio di questo *telecomando* provocherà la sostituzione della configurazione presente nel registro del μ C.

Il *telecomando* in esame consente di selezionare le grandezze di *telemetria* e la cadenza con cui queste ultime dovranno essere misurate.

➤ **Telecomando per il settaggio del registro TIPO_TME**

Consente di alterare il contenuto del registro TIPO_TME descritto nel §1.6. L'invio di questo *telecomando* provocherà la sostituzione della configurazione presente nel registro del μ C.

Il *telecomando* in questione consente di stabilire quali grandezze di *telemetria* estesa addizionale abilitare.

➤ **Telecomando cancellazione automatica dei dati di TME**

Consente di abilitare o no la cancellazione automatica dei dati di TME al momento della loro trasmissione alla stazione di Terra. Se tale *telecomando* è orientato alla disabilitazione della cancellazione automatica dei dati, significa che questi ultimi potranno essere cancellati solamente nel caso in cui il satellite riceva dalla stazione di Terra la conferma di avvenuta ricezione. Siccome, per i limiti imposti del protocollo utilizzato per il collegamento, non sarà possibile inviare tutti i dati di TME in un'unica trasmissione (max 256 byte), la cancellazione avverrà solamente nel momento in cui si sarà ottenuta la conferma dell'avvenuta ricezione di tutti i dati. Si tenga presente che ad ogni inizio di *tempo di attesa* la TME meno recente tra le due memorizzate (vedi §1.4) viene automaticamente cancellata.

➤ **Telecomando di inizio del tempo di attesa**

Consente di segnalare al satellite l'inizio del *tempo di attesa*, quindi ad avviare il conteggio dei secondi di ciascuna orbita, nonché cominciare la tracciatura del numero di accessi non autorizzati (nel caso tale grandezza sia stata attivata). Tramite il *telecomando* in questione viene quindi avviata la procedura di acquisizione dei dati di TME.

➤ **Telecomando per le cadenze delle misure dagli strumenti di bordo**

Consente di impostare l'intervallo di tempo tra un'acquisizione dagli strumenti di bordo e la successiva. Inizialmente il valore della cadenza è posto a 237 secondi. All'interno del μC che gestisce tali acquisizioni sarà impostata una soglia minima (non modificabile) al di sotto della quale non si potrà andare in modo da non saturare la memoria del satellite. Il valore di quest'ultima determina il valore della soglia.

Una volta ricevuto il *telecomando* in questione, la nuova cadenza andrà a sostituire la precedente. Siccome all'interno del satellite saranno presenti due μC , uno che gestisce esclusivamente l'acquisizione dei dati strumentali e l'altro che si occupa della gestione del rimanente sistema, nel momento in cui viene ricevuto il nuovo dato di cadenza, esso dovrà essere trasmesso, attraverso il bus interno, al μC incaricato delle acquisizioni.

➤ **Telecomando di ripristino del sistema**

Consente di reinizializzare tutte le configurazioni di *default*. Ciò è possibile grazie alla memorizzazione (in fase di programmazione del μC) di tutte le impostazioni iniziali all'interno della EEPROM del μC . Tale *telecomando*, una volta ricevuto, verrà anche trasmesso al μC che si occupa delle acquisizioni dagli strumenti di bordo.

2.3 Telecomando di richiesta

Con il termine *telecomando di richiesta*, si intende quel particolare *telecomando* atto ad ottenere dal satellite determinate informazioni in merito o al suo stato o ad una misura. Al momento attuale è stato definito solamente un *telecomando* appartenente a questa categoria, esso è descritto di seguito.

➤ **Telecomando di richiesta TMB**

Nel §1.3 è stato detto che l'invio automatico dei dati di TMB, a determinate cadenze, può essere disattivato tramite l'opportuno *telecomando* descritto nel capitolo precedente. Nel caso quindi in cui l'automazione di tale invio è disabilitata, la stazione di Terra può, inviando una richiesta (***telecomando di richiesta***), procurarsi la TMB in quel determinato istante. L'esecuzione del *telecomando* in questione avverrà nel momento in cui sarà terminato l'invio del pacchetto di dati precedente.

Capitolo 3

Struttura interna di ATMOCUBE (livello hardware)

Da differenti lavori di tesi, si è giunti ad uno scenario in cui vi erano diversi sotto sistemi isolati i quali necessitavano di interagire tra loro. Affinché ATMOCUBE potesse – per così dire – prendere forma, è stata necessaria l'introduzione di un microcontrollore in modo da gestire i sottosistemi e introdurre una sorta di intelligenza al satellite. Il microcontrollore in questione consente quindi l'interazione e il controllo di tutto il sistema.

Al termine di continui abbozzi, si è giunti ad uno schema generale che riassume in modo sufficientemente dettagliato la struttura interna del satellite. Tale schema è riportato al termine della presente tesi. Dallo stesso sono identificabili le seguenti sezioni:

- SEZIONE DI ALIMENTAZIONE
- SEZIONE DI *HOUSEKEEPING*
- ASSETTO
- SEZIONE DI TRASMISSIONE E RICEZIONE
- SEZIONE DI *DATA HANDLING*

Dallo schema generale inoltre, si può dapprima constatare che a bordo del satellite saranno presenti due unità di calcolo/controllo denominate μC MASTER e μC SLAVE. I loro compiti saranno spiegati nelle sezioni di *housekeeping* e di *data handling*. In linea di massima il primo tra i due si occupa del controllo del satellite, nonché della trasmissione a Terra dei dati. Il secondo invece, ha esclusivamente il compito di prelevare i dati strumentali.

In questo capitolo, dopo una descrizione del microcontrollore utilizzato e dei suoi moduli interni, e quindi dopo la scelta dei pin per il suo interfacciamento, sarà esposta in modo dettagliato la struttura *hardware* di ciascuna sezione.

3.1 I μ C P18F452 e P18F252

I μ C P18F452 e P18F252, prodotti dalla MICROCHIP, fanno parte della famiglia 18 e sono caratterizzati dall'aver tra le varie caratteristiche un set di istruzioni più esteso rispetto alla famiglia 16, nonché un'organizzazione differente della memoria (non più a banchi orizzontali ma verticali). Ciò che ha spinto ad optare per la famiglia 18 anziché la 16, è stata l'esistenza in commercio di un compilatore di linguaggio C dedicato proprio a tale famiglia; ciò rende la programmazione molto più semplice, in quanto si lavora ad alto livello di programmazione e non a basso livello (caratteristica peculiare dell'ASSEMBLER).

Entrambi i μ C sono caratterizzati dalla lettera F presente all'interno della loro sigla. Essa rappresenta il tipo di memoria di programma presente al loro interno, che in questo caso risulta essere di tipo Flash (è possibile cancellarla e riscriverla per un elevato numero di volte). Tale famiglia, come la 16, lavora a 8 bit (1 byte) pertanto tutti i registri interni saranno di tale dimensione.

I due μ C presi in esame si differenziano, oltre che per il numero di pin, e quindi di porte, anche per il numero di sorgenti di *interrupt*, nonché per le linee dedicate alla conversione analogica-digitale (a 10bit). Entrambi sono dotati al loro interno di una memoria di programma da 32 Kbyte, una memoria RAM da 1.536 byte e una memoria EEPROM da 256 byte.

I μ C P18F452 e P18F252, verranno di seguito nominati rispettivamente come μ C MASTER e μ C SLAVE.

In questa trattazione, ci si concentrerà maggiormente sul μ C MASTER, dal momento che il μ C SLAVE, ad eccezione delle caratteristiche precedentemente elencate, lavora in modo del tutto equivalente; inoltre il P18F252, di cui verranno date alcune linee guida, è oggetto di altrui lavoro di tesi.

Di seguito vengono proposti gli schemi che descrivono le funzionalità che possono essere assegnate alle porte (pin) dei due μ C.

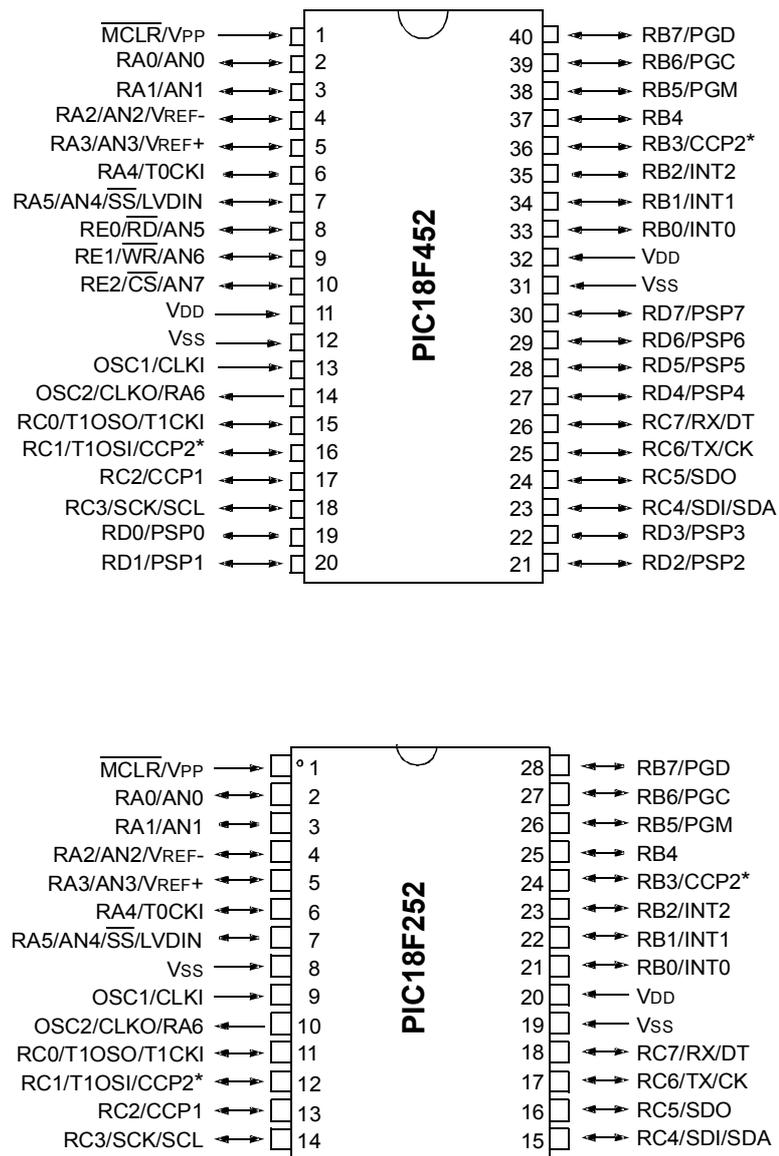


Figura 3.1: Possibili funzionalità delle piedinature dei μC utilizzati

Facendo riferimento al μC P18F452, viene riportata la sua struttura interna.

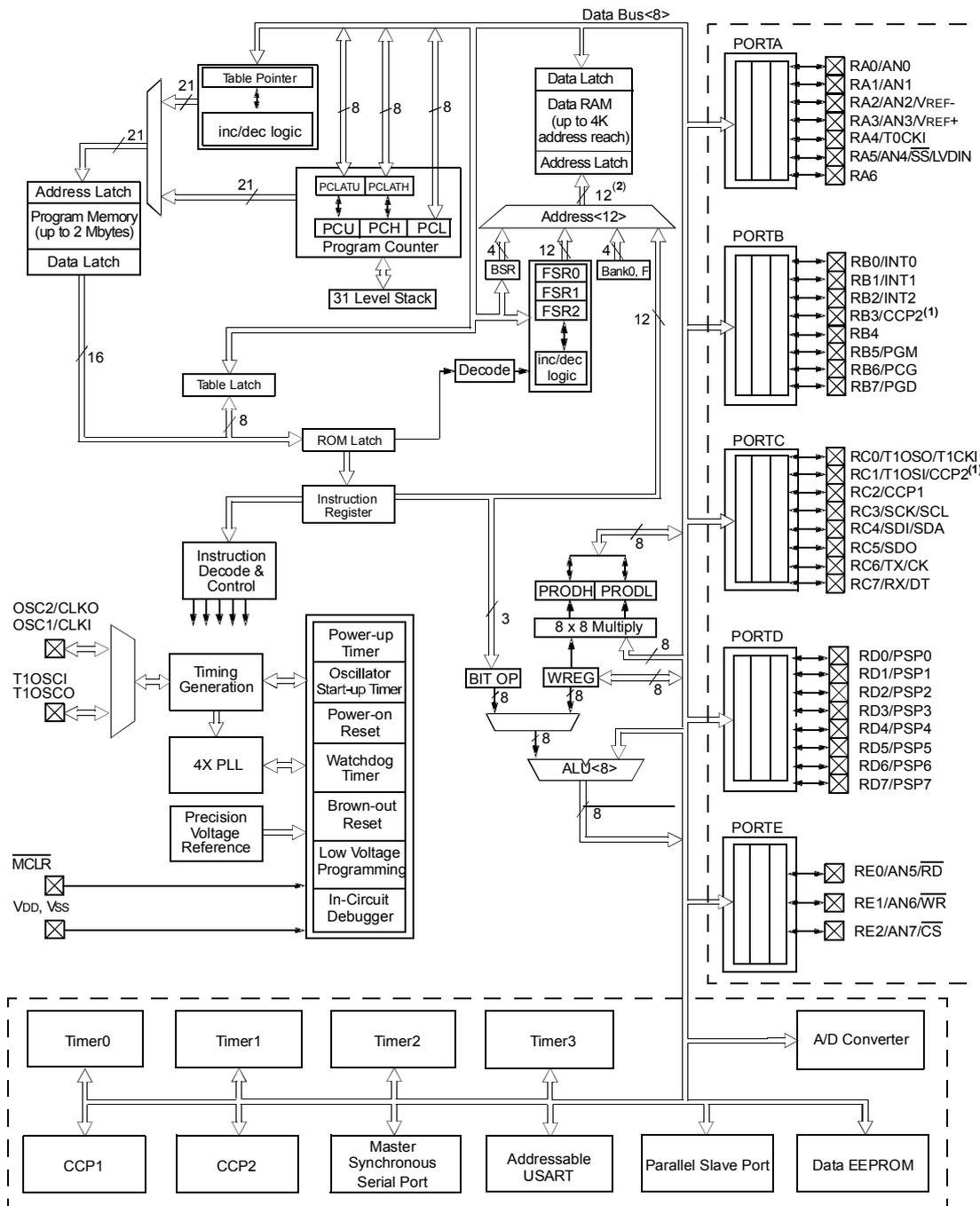


Figura 3.2: Struttura interna del μ C P18F452

3.1.1 Moduli interni del μC MASTER utilizzati

Nell'interfacciamento del μC MASTER con i sottosistemi, nonché per il funzionamento interno, sono stati utilizzati alcuni suoi moduli interni. Tra questi vi è:

- modulo **ADC** (a 10bit)
- modulo **PWM**
- modulo **SPI**
- *timer 0*
- *timer 1*
- *timer 2*
- porta **USART**

Per ciascuno dei suddetti moduli, segue una breve descrizione.

3.1.1.1 Modulo ADC (*Analog Digital Converter*)

Il convertitore AD all'interno del μC ha la seguente struttura:

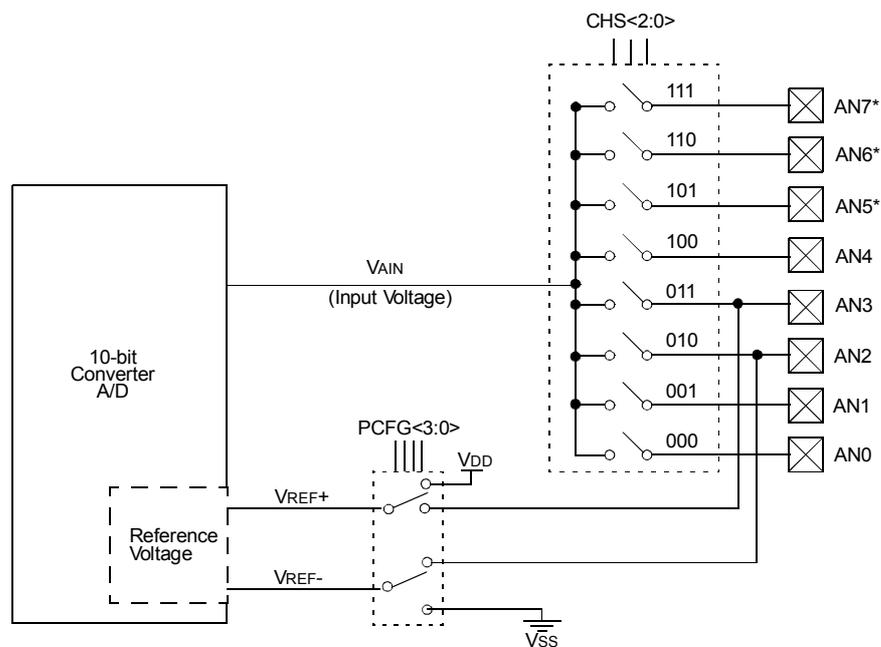


Figura 3.3: Convertitore AD

Il modulo in esame, affinché funzioni, utilizza il *timer 2*. All'uscita RC2 infatti, ho la presenza di un segnale digitale ad onda quadra, con periodo impostabile tramite il registro PR2 e duty cycle tarabile agendo sul registro CCPR1L e sui bit CCP1CON<5:4>. Le formule necessarie al calcolo dei due sono descritte nel *datasheet* e vengono riportate di seguito.

$$\text{Periodo PWM} = [(PR2) + 1] \times T_{osc} \times (\text{valore del prescaler del TMR2})$$

Formula 3.1: Calcolo del periodo della PWM

$$f_{PWM} = \frac{1}{\text{Periodo PWM}}$$

Formula 3.2: Calcolo della frequenza della PWM

$$\text{duty cycle PWM} = (CCPR1L : CCP1CON < 5 : 4 >) \times T_{osc} \times (\text{valore del prescaler del TMR2})$$

Formula 3.3: Calcolo del duty cycle della PWM

dove con il termine *CCPR1L : CCP1CON < 5 : 4 >* si intende il valore ottenuto dall'unione dei due registri.

Viene inoltre fornita la risoluzione del modulo PWM che è funzione sia della frequenza del *clock* del μC , sia della frequenza del modulo PWM precedentemente calcolata.

$$\text{Risoluzione(max) PWM} = \frac{\log\left(\frac{f_{osc}}{f_{PWM}}\right)}{\log 2} \quad \text{bits}$$

Formula 3.4: Calcolo della risoluzione del modulo PWM

3.1.1.3 Modulo SPI (*Serial Peripheral Interface*)

Il bus SPI è un bus sincrono e *full-duplex* ideale in tutte quelle situazioni in cui un dispositivo *master* deve interfacciarsi a diversi dispositivi *slave* e con loro dialogare in modo efficace e prestante. Esso si basa su 4 segnali:

- **SCLK** *Serial Clock*
- **SDI** *Serial Data Input*
- **SDO** *Serial Data Output*
- **CS** *Chip Select*

Di questi il *Chip Select* non è indispensabile in tutte le situazioni (nella nostra si in quanto abbiamo più dispositivi che comunicano sul bus in esame).

Il segnale SCLK è il *clock* seriale che scandisce gli istanti di emissione e di lettura dei bit sulle linee di dato. È un segnale emesso dal MASTER ed è quindi quest'ultimo a richiedere di volta in volta la trasmissione di una parola.

Il segnale SDI è la linea attraverso cui il dispositivo (MASTER o SLAVE) riceve il dato seriale emesso dalla controparte. Corrispondentemente, il dispositivo emette, con la stessa cadenza, il suo output ponendo il dato sulla linea SDO (linea di output di dato).

La linea CS è dedicata all'abilitazione del dispositivo SLAVE da parte del MASTER in modo da informare il primo che la trasmissione sul bus è dedicata a esso. La linea CS, normalmente attiva a livello logico basso, in caso di disabilitazione, lascia il dispositivo SLAVE con uscita in alta impedenza facendolo quindi rimanere completamente isolato.

Il numero di dispositivi SLAVE che si possono connettere al bus è limitato esclusivamente dal numero di possibili linee di *chip select* gestibili dal dispositivo master. La frequenza di *clock* (e di conseguenza la velocità del bus) può raggiungere, con questo standard, livelli anche elevati nell'ordine delle decine di MHz ed anche oltre.

Il diagramma a blocchi del modulo SPI è il seguente:

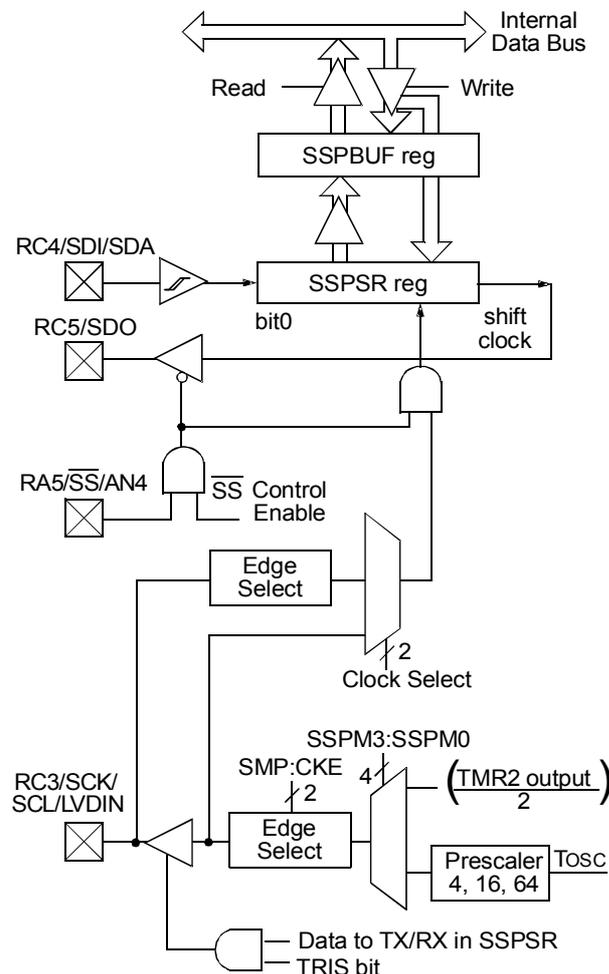


Figura 3.5: Schema a blocchi del modulo SPI

La trasmissione dei dati sul bus SPI si basa sul funzionamento del registro a scorrimento SSPSR. Tale registro è presente sia nel MASTER che nello SLAVE. Ad ogni colpo di *clock*, i bit presenti nel registro vengono fatti scorrere e inviati tramite la linea SDO lasciando la cella liberata ai bit ricevuti tramite la linea SDI. Il registro può avere dimensione arbitraria (ma uguale per dispositivo MASTER e SLAVE) anche se il più delle volte ha la dimensione di 8 bit.

Il registro a scorrimento è un'interfaccia completa, mediante la quale vengono impartiti comandi e trasmessi dati che arrivano in modo seriale, ma che internamente sono prelevati a fine trasmissione, in modo parallelo.

Per quanto riguarda le configurazioni dei registri si rimanda al *datasheet*.

3.1.1.4 Timer 0, Timer 1 e Timer 2

I *timer* del μ C sono dei dispositivi in grado generare, una volta programmati opportunamente, una prestabilita temporizzazione. Questi si basano su dei contatori (da 8 o 16 bit), i quali vengono incrementati ad ogni colpo di *clock*. Nel momento in cui oltrepassano il limite di conteggio (256 oppure 65.536), vale a dire vanno in *overflow*, possono scatenare un *interrupt*. Questo verrà opportunamente gestito dal μ C.

Nel nostro caso il *timer 2* viene utilizzato dal modulo PWM per generare il segnale digitale descritto tramite le Formule 3.6, 3.5, 3.1 e 3.2. Il *timer 1*, con l'ausilio di un quarzo esterno, viene invece impiegato per scandire gli intervalli di tempo della *telemetria* (da parte del μ C MASTER), nonché dell'acquisizione dei dati strumentali (da parte del μ C SLAVE). Il *timer 0* infine viene utilizzato dai μ C per generare delle temporizzazioni interne necessarie al normale utilizzo.

Al loro interno è possibile abilitare un dispositivo di prescaler, il quale opera una divisione di frequenza.

Nelle figure seguenti, vengono proposti gli schemi a blocchi interni.

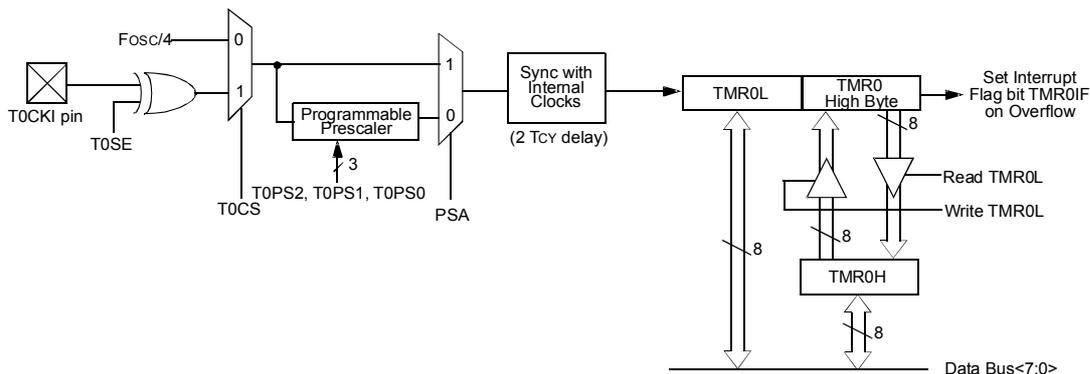


Figura 3.6: Schema a blocchi del timer 0

3.1.1.5 Modulo USART

L'USART (Universal Synchronous Asynchronous Receiver Trasmitter) è uno dei due moduli di I/O seriali del μ C: esso è conosciuto anche come Serial Communications Interface (SCI).

L'USART può essere configurato in modalità *full-duplex* asincrona per poter comunicare con dispositivi periferici, quali terminali CRT e personal computer, oppure in modalità *half-duplex* sincrona per comunicare con dispositivi periferici tipo circuiti integrati A/D o D/A , EEPROM seriali o altri μ C.

Le configurazioni possibili dell'USART sono:

- **Asincrona (*full-duplex*)**
- **Sincrona – Master (*half-duplex*)**
- **Sincrona – Slave (*half-duplex*)**

Il modulo USART ha una compatibilità a 9 bit, cioè si possono ricevere e trasmettere anche 9 bit per frame (c'è la possibilità di gestire il bit di parità).

Per utilizzare il modulo, è necessario operare su determinati registri che mi configurano la porta di uscita e quella di ingresso.

All'interno del modulo è presente un *Baud Generator*, il quale genera il *clock* per lo scorrimento dei dati contenuti nei registri. Il *Baud Generator* è configurabile tramite un opportuno registro; dal *datasheet* è possibile calcolare i suoi possibili valori.

I sottosistemi che implementano la comunicazione seriale (GPS e dosimetro) utilizzano la porta USART in configurazione asincrona; tale modalità usa il formato standard non-return-to-zero (NRZ) che prevede un bit di start e uno di stop, otto o nove bit per dato. Il formato più comune di trasmissione è quello ad otto bit. Per la generazione del baud rate viene impiegato un registro dedicato che deriva la frequenza del baud rate dall'oscillatore del microcontrollore. La trasmissione e la ricezione inizia dal bit meno significativo LSB. La trasmissione e ricezione funzionano in modo indipendente, usano lo stesso formato per i dati e lo stesso baud rate. La parità non è supportata nel hardware, ma può essere implementata via software (memorizzando il nono bit del dato). La modalità asincrona è fermata durante lo SLEEP.

Di seguito vengono riportati i diagrammi a blocchi della sezione di trasmissione e ricezione nella modalità asincrona.

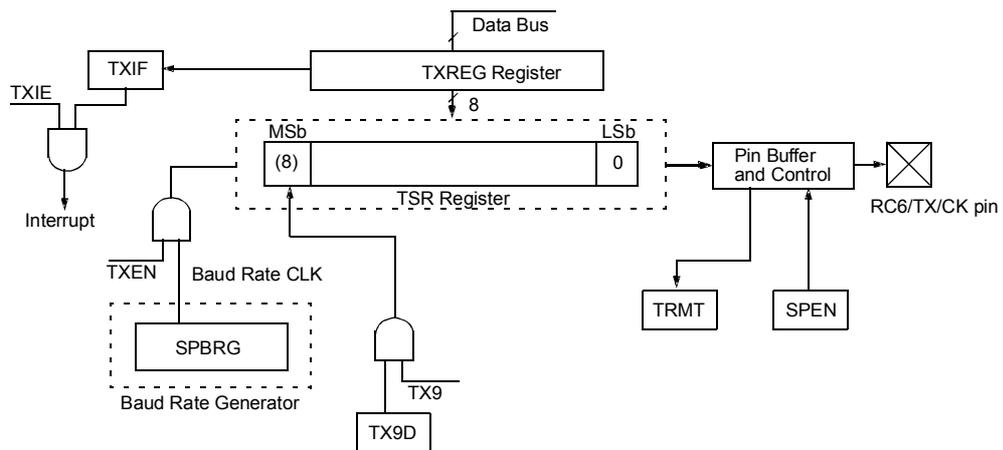


Figura 3.9: Porta USART in modalità trasmissione asincrona

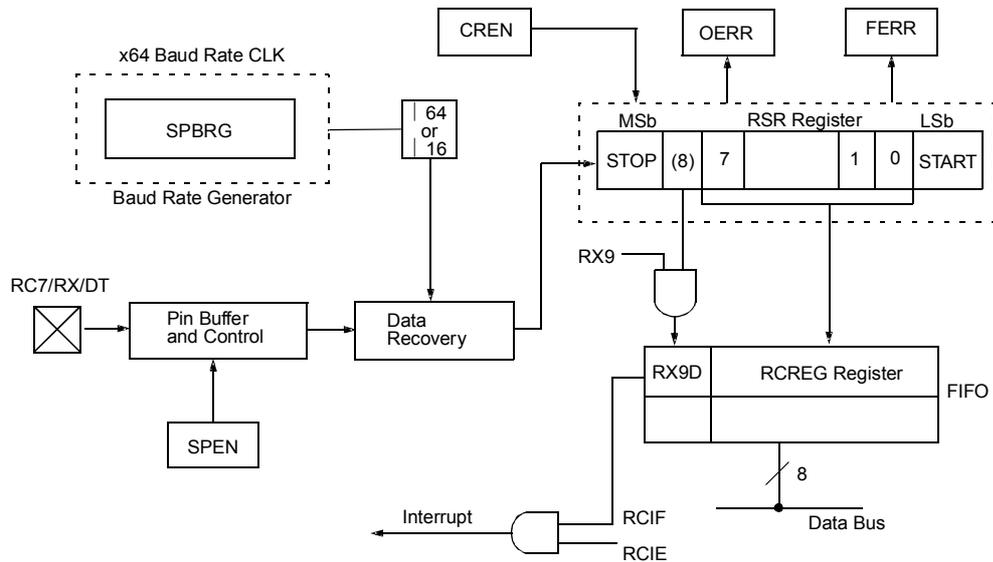


Figura 3.10: Porta USART in modalità ricezione asincrona

3.1.2 Scelta ed uso dei pin del μC MASTER

Il μC MASTER si trova in commercio nelle versioni DIP, PLCC e TQFP. Ognuno di questi *footprint*, si differenzia sostanzialmente per il passo dei pin, dalle dimensioni del componente nonché dalla tecnica di saldatura utilizzata. Il μC preso in esame in questo elaborato, si trova nel formato DIP (*Dual In Package*) e il suo aspetto è visibile nella Figura 3.11.

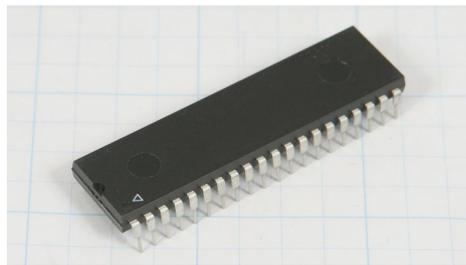
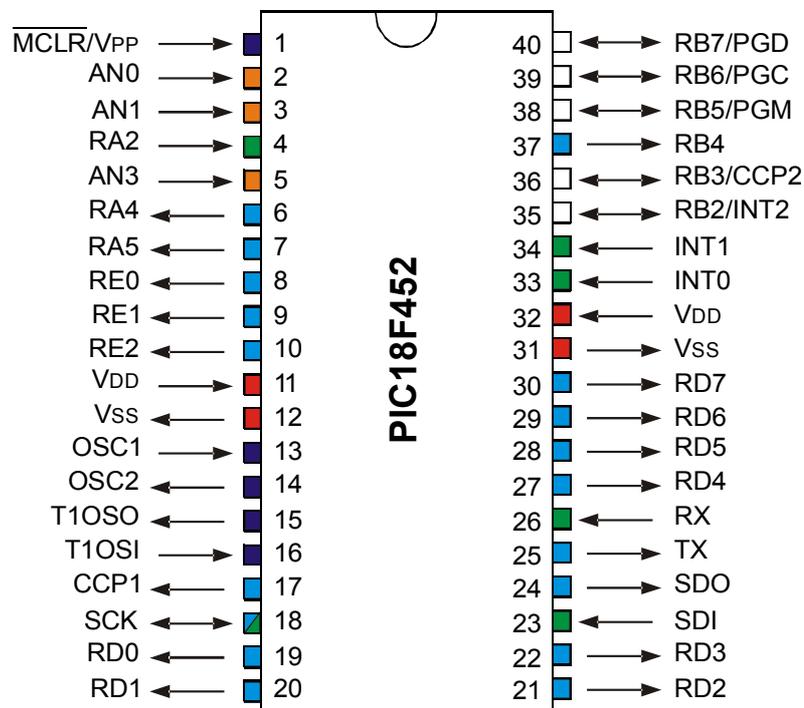


Figura 3.11: Aspetto del μ C MASTER

La scelta del *footprint* è ricaduta sul DIP in quanto, al momento attuale, i nostri obiettivi non sono quelli di ottenere un sistema definitivo, bensì un prototipo; il formato DIP inoltre ha il vantaggio di consentire un facile accesso ai pin, consentendo un agevole *debug* del circuito. Una volta realizzato e verificato che funzioni in tutte le sue parti, si potrà procedere alla miniaturizzazione delle schede elettroniche e quindi convertire il formato DIP in quello TQFP.

Nella figura seguente, viene riportata l'assegnazione dei pin, mentre nella Tabella 3.1 vengono descritti i compiti a cui dovranno sottostare.

DIP



LEGENDA

- PIN SEGNALE ANALOGICO
- PIN SEGNALE DIGITALE IN USCITA DAL μC
- PIN SEGNALE DIGITALE IN INGRESSO AL μC
- PIN PER IL FUNZIONAMENTO DEL μC
- PIN DI ALIMENTAZIONE

Figura 3.12: Assegnazione dei pin del μC MASTER

Dalla figura sopra si può notare che alcuni pin sono al momento inutilizzati. Occorre tenere presente che nello schema generale di fine elaborato, non si fa nessun riferimento alle linee necessarie alla programmazione del μC . Tali linee sono: la $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ (pin 1), quella $\text{PGD}/\text{RB7}$ (pin 40), quella $\text{PGC}/\text{RB6}$ (pin 39) e quella $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ (pin 1). Non sono state inoltre considerate le linee necessarie al sottosistema per l'assetto.

| | | | | | |
|----|----------|--|----|----------|---|
| 1 | MCLR/VPP | Linea per programmare | 21 | RD2 | CS Memoria 1 |
| 2 | AN0 | Linea analogica per pannelli solari | 22 | RD3 | CS Memoria 2 |
| 3 | AN1 | Linea analogica per tensioni, correnti e temperature | 23 | SDI | SPI Serial Data Input |
| 4 | RA2 | Segno della corrente della batteria | 24 | SDO | SPI Serial Data Output |
| 5 | AN3 | | 25 | TX | USART con CC1100 |
| 6 | RA4 | U1 shutdown | 26 | RX | USART con CC1100 |
| 7 | RA5 | U2 shutdown | 27 | RD4 | Comando per switch antenna |
| 8 | RE0 | A0 su ADG708 | 28 | RD5 | Segnale da μ C MASTER a μ C SLAVE per avvertire della comunicazione sul bus SPI |
| 9 | RE1 | A1 su ADG708 | 29 | RD6 | ON/OFF Preamplificatori dell'amplificatore di potenza |
| 10 | RE2 | A2 su ADG708 | 30 | RD7 | ON/OFF Stadio finale di potenza |
| 11 | VDD | +3,3V | 31 | VSS | +3,3V |
| 12 | VSS | GND | 32 | VDD | GND |
| 13 | OSC1 | Quarzo da 20MHz | 33 | INT0 | Interrupt proveniente da μ C SLAVE per chiedere la comunicazione sul bus SPI |
| 14 | OSC2 | Quarzo da 20MHz | 34 | INT1 | Interrupt proveniente dal sottosistema per l'assetto |
| 15 | T1OSO | Quarzo da 32.768 Hz | 35 | RB2/INT2 | |
| 16 | T1OSI | Quarzo da 32.768 Hz | 36 | RB3/CCP2 | |
| 17 | CCP1 | PWM per i pannelli solari | 37 | RB4 | ON/OFF LNA |
| 18 | SCK | SPI Clock | 38 | RB5/PGM | |
| 19 | RD0 | CS per CC1100 | 39 | RB6/PGC | |
| 20 | RD1 | CS (SYNC) DAC AD5300 | 40 | RB7/PGD | |

Tabella 3.1: Assegnazione dei compiti di ciascun pin del μ C

3.2 Sezione di alimentazione

Tale sezione è già stata per buona parte studiata in un precedente lavoro di tesi (Marco De Din) e comprende:

- i **PANNELLI SOLARI** (collegati in serie)
- un circuito di **POWER CHARGE & POWER MANAGMENT** (dedicato ai pannelli)
- un **DC/DC CONVERTER DI TIPO BOOST** (questione dell'MPPT) comandato dal $\mu\text{C MASTER}$ tramite il modulo PWM incorporato in esso
- un altro circuito di **POWER CHARGE & POWER MANAGMENT** (dedicato alla batteria)
- la **BATTERIA**
- una serie di **DC/DC CONVERTER** per ottenere le tensioni di +5V e +3,3V per alimentare gli strumenti e il resto della circuiteria

I pannelli solari forniscono una tensione a vuoto di 2,415 V ed una corrente di cortocircuito di 355 mA. Tramite il DC/DC Converter di tipo *boost*, tali pannelli vengono sfruttati nel punto di lavoro tale per cui forniscono la massima potenza: si ottiene quindi una tensione di 2,3 V e una corrente di 344 mA (problema dell'MPP – Maximum Power Point). Collegando in serie tre pannelli solari si ottiene una tensione di 6,9 V.

Il controllo del DC/DC Converter di tipo *boost* sopra citato, avviene per mezzo della circuiteria interna al $\mu\text{C MASTER}$. Tale circuiteria è gestita da un codice (già stilato) che fa uso della tensione dei pannelli solari: quest'ultima viene opportunamente prelevata dal μC tramite il pin 2, connesso al convertitore A/D a 10 bit presente al suo interno. Tale controllo prende il nome di PWM (*Pulse With Modulation*). Inoltre, tramite il DC/DC Converter di tipo *boost*, si passa da una tensione di circa 7V dei pannelli solari, a una tensione di 10,8 V, la quale andrà a caricare la batteria, ad alimentare il magnetometro e quindi a fornire energia agli altri convertitori necessari a ottenere le tensioni di +5V e +3,3V.

L'integrato che svolge il compito di *Power Charge & Power Managment* è il MAX471. Tale circuito integrato, collegato come riportato in Figura 3.13, mi consente di misurare la corrente che lo attraversa (in questo quella proveniente dai pannelli).

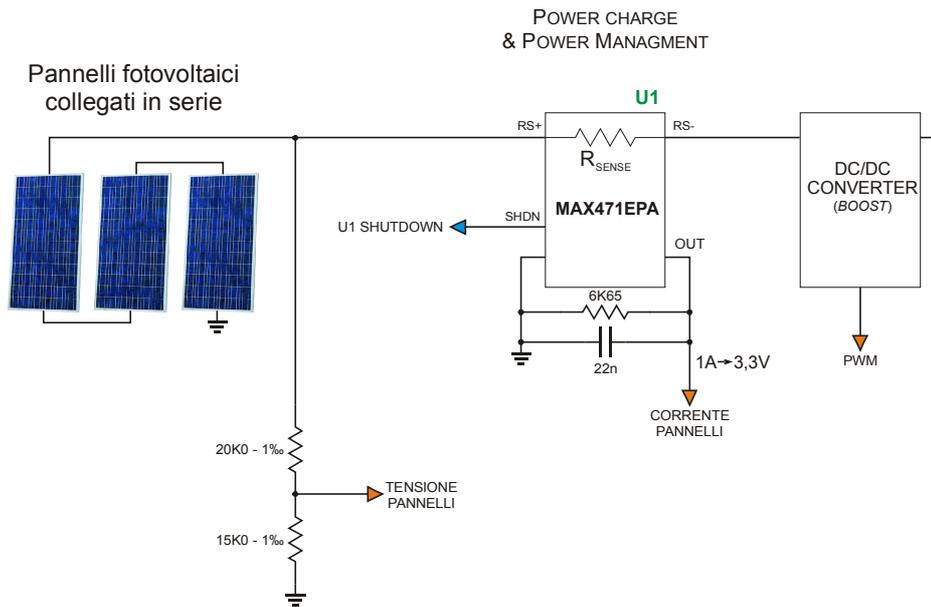


Figura 3.13: Collegamento dell'integrato MAX471

Al suo interno vi è una resistenza di *sense* tipica di $35\text{ m}\Omega$ ed una coppia di amplificatori differenziali i quali, misurando la caduta di tensione ai capi della resistenza, forniscono in uscita una corrente proporzionale a quella che lo attraversa. Siccome l'uscita è in corrente ($500\mu\text{A/A}$), tramite una resistenza di valore opportuno, possiamo lavorare con le tensioni. Il calcolo di tale resistenza avviene tramite la seguente formula:

$$R_{\text{OUT}} = \frac{V_{\text{OUT}}}{(I_{\text{LOAD}} \times 500\mu\text{A/A})} = \frac{3,3\text{V}}{(1\text{A} \times 500\mu\text{A/A})} = 6,6\text{K}\Omega$$

quindi $R_{\text{OUT}} = 6,65\text{K}\Omega$ (valore commerciale che più si avvicina)

Formula 3.5: Calcolo della resistenza di uscita del MAX471

In questo modo abbiamo che ad 1A di corrente che fluisce attraverso l'integrato, corrispondono 3,3V in uscita. Tale resistenza è stata inserita anche all'uscita del MAX471 che si occupa della misura della corrente della batteria. Il calcolo di quest'ultima resistenza è differente dal precedente, poiché può accadere che la batteria debba fornire dei picchi di corrente al magnetometro, questi provocherebbero quindi una tensione, in uscita dal MAX471, che danneggerebbe il μC . Una possibile soluzione è quella di inserire, in parallelo alla resistenza, un diodo zener da 3,3V come protezione.

Per quanto riguarda il valore della resistenza abbiamo:

$$R_{OUT} = \frac{V_{OUT}}{(I_{LOAD} \times 500\mu A/A)} = \frac{3V}{(1A \times 500\mu A/A)} = 6K\Omega$$

quindi $R_{OUT} = 6,04K\Omega$ (valore commerciale che più si avvicina)

Formula 3.6: Calcolo della resistenza di uscita del MAX471 per la batteria

Come si vede dalla Figura 3.14, il circuito integrato in esame è dotato di un ulteriore pin che mi dà l'indicazione del senso della corrente. La batteria infatti dovrà sia caricarsi sia fornire energia; la corrente quindi fluirà in entrambi i sensi. Dal *datasheet* dell'integrato si apprende che, quando tale uscita (a *open collector*) è a livello logico basso, la batteria sta fornendo energia. Tale pin dell'integrato infine viene connesso al μC MASTER e il suo stato logico costituisce un dato di *telemetria*.

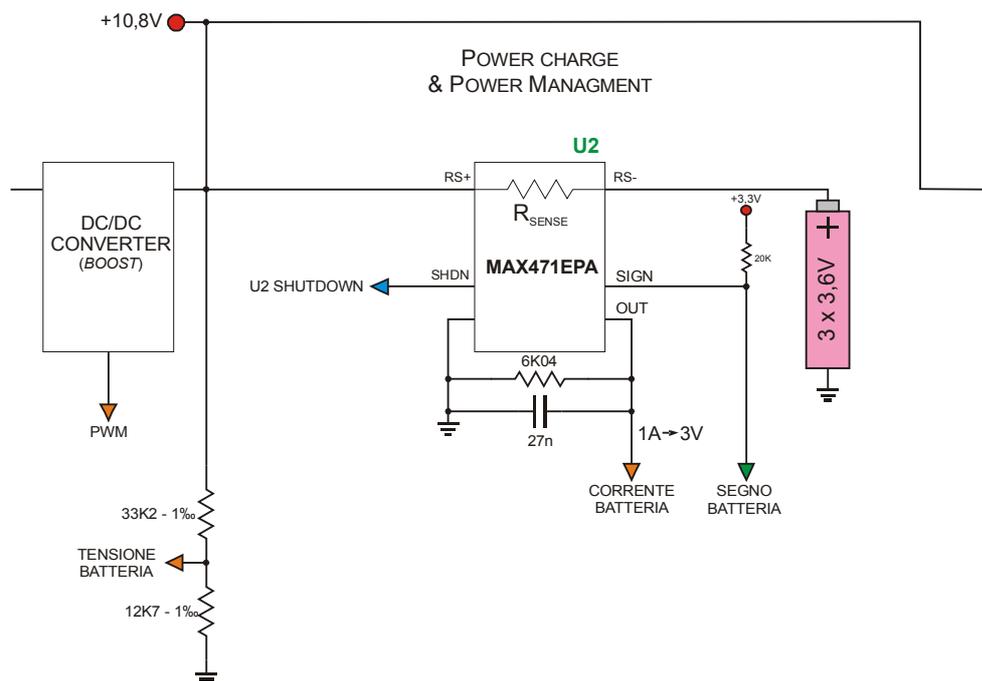


Figura 3.14: Inserimento del MAX471 per la misura della corrente della batteria

Il MAX471 è dotato infine di un particolare piedino (quello di SHUTDOWN) che, collegato a massa, fa entrare l'integrato in modalità *low power* assorbendo una corrente di $18\mu\text{A}$. Per entrambi i MAX471 utilizzati, tale pin è collegato al $\mu\text{C MASTER}$ (U1 SHUTDOWN è collegato a RA4 mentre U2 SHUTDOWN è collegato a RA5). Ponendo quindi lo stato della porta al livello logico alto, il MAX471 si pone in condizione di *power down mode*; viceversa si ha il normale utilizzo.

3.3 Sezione di *housekeeping*

È stato detto che per il mantenimento del satellite (Capitolo "Telemetria"), è necessaria la conoscenza delle tensioni, di alcune correnti e della memoria; la sezione di *housekeeping* ha l'incarico di acquisire questi dati. Per quanto riguarda la situazione della memoria, sarà il *firmware* interno al $\mu\text{C MASTER}$ che se ne occuperà, per le rimanenti grandezze invece, si fa uso di una circuiteria esterna.

Siccome per le acquisizioni viene utilizzato l'ADC contenuto nel μC (descritto in §3.1.1.1), c'è da dire subito che per ottenere delle misure valide occorre avere un *reference* stabile. Il μC in esame utilizza come *reference* interno la tensione di alimentazione, ma è dotato anche di due piedini che consentono di applicare un riferimento esterno al convertitore A/D (non utilizzati però a tale scopo). Si suppone che l'acquisizione delle grandezze avvenga una volta che la tensione dei 3,3 volt si è stabilizzata.

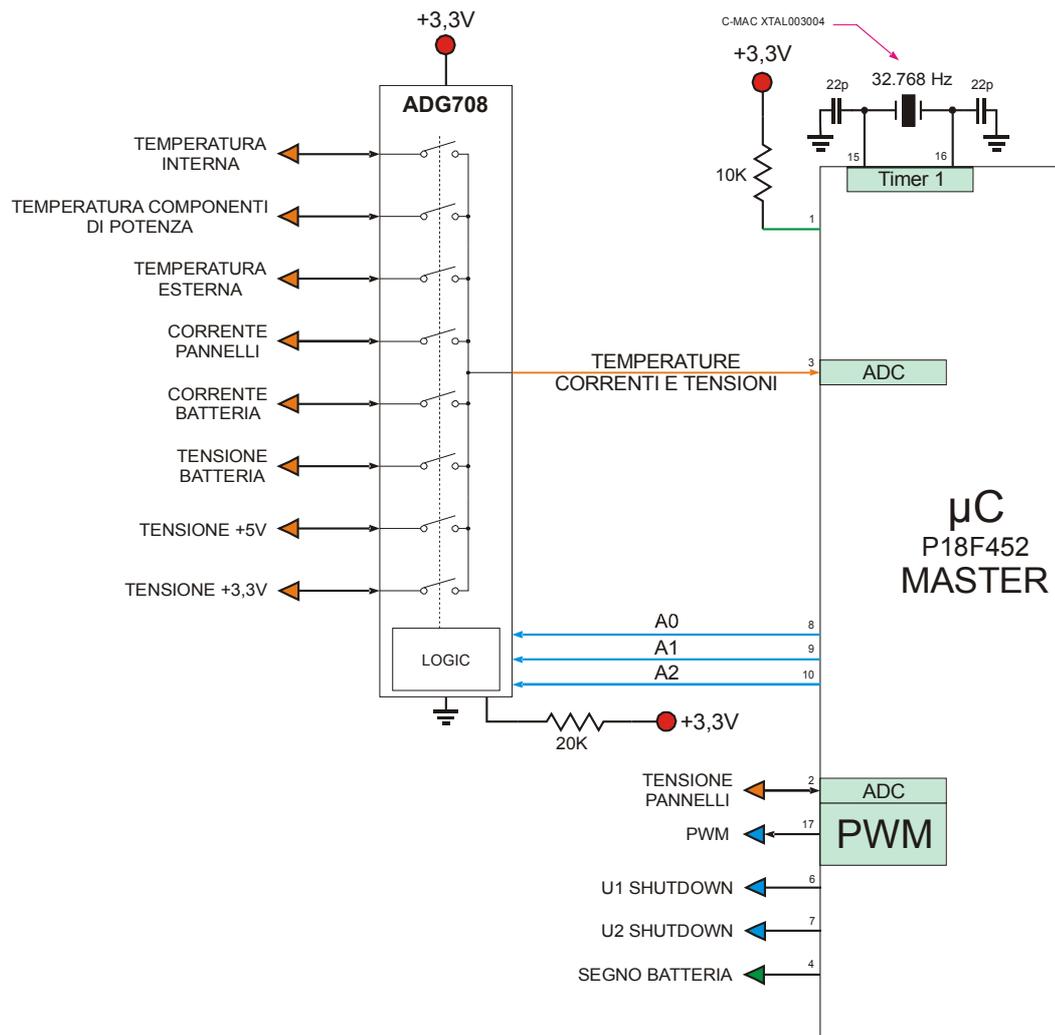


Figura 3.15: Connessioni per la telemetria di housekeeping

Come si può notare dalla Figura 3.15, all'integrato ADG708 convogliano i segnali di tensione (ad eccezione della tensione dei pannelli), di corrente e di temperatura. Tale integrato svolge la funzione di multiplexer, ovvero commuta alternativamente gli ingressi (in questo caso otto) sull'uscita. Quest'ultima viene connessa al pin 3 del μC MASTER il quale procederà (alternativamente) alla loro acquisizione tramite l'ADC interno. Le linee A0, A1, e A2 provenienti dal μC , vanno ad agire sulla logica interna dell'integrato in esame, il quale, a seconda della combinazione, eseguirà la commutazione di un preciso *switch* interno sull'uscita. Nel caso in cui siamo nelle condizioni di TME, al momento opportuno, cioè secondo la Tabella 1.1, il μC MASTER provvederà ad effettuare la selezione e, successivamente, l'acquisizione dei canali di interesse in quel momento (la TME non richiede sempre l'immagazzinamento di tutte le grandezze di *telemetria*).

La tensione dei pannelli solari, a differenza delle altre, non convoglia sull'integrato ADG708, ma sul pin 2 del $\mu\text{C}_{\text{MASTER}}$. Tale scelta, oltre ad essere dettata dal limitato numero di ingressi dell'ADG708, è stata operata in quanto la tensione considerata deve essere acquisita quasi continuamente, in modo tale da ottimizzare l'utilizzo dei pannelli solari. Come infatti è stato detto nel §3.2, particolare procedura del *firmware* del μC usufruisce di tale tensione per comandare il modulo PWM presente al suo interno, che a sua volta va a comandare il DC/DC Converter di tipo *boost*. Se avessimo posto la tensione dei pannelli solari all'ingresso dell'ADG708 (ammesso che vi fosse stato un pin inutilizzato), ad ogni controllo ad opera del *firmware* (che avviene con una cadenza maggiore rispetto alle altre grandezze di *telemetria*) avremmo dovuto selezionare lo *switch* corrispondente: il $\mu\text{C}_{\text{MASTER}}$ avrebbe quindi eseguito istruzioni inutili, causando una perdita di tempo. L'ADC contenuto nel μC inoltre, affinché possa essere utilizzato, deve essere prima configurato; l'unica impostazione del modulo idonea ai nostri scopi, comprende tre linee analogiche (che convogliano sull'ADC): siccome ne sono state utilizzate due, una è inutilizzata (pin 5).

Come sensore di temperatura interna possono essere utilizzati due tipi di componenti discreti: LM19 oppure KTY81. Il primo, opportunamente alimentato, mi fornisce in uscita una tensione proporzionale alla temperatura (come mostrato in Figura 3.16 ed espresso tramite le Formule 3.7 e 3.8) assorbendo $10\mu\text{A}$; il secondo equivale invece ad una termo resistenza (occorrerà inserirlo in un partitore di tensione perché la sua resistenza varia da circa 500Ω a circa $2\text{K}\Omega$ e comporterebbe un assorbimento di corrente eccessivo).

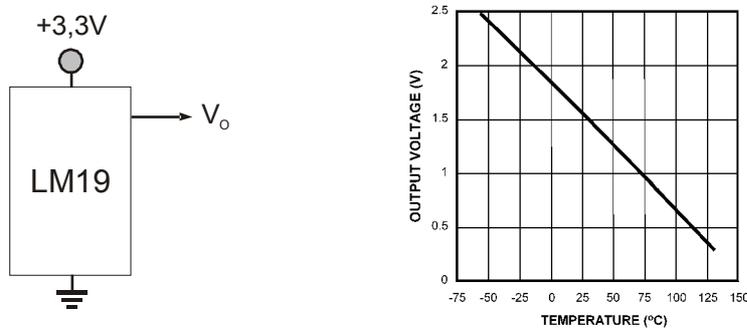


Figura 3.16: Sensore di temperatura LM19

$$V_o = (-3,88 \times 10^{-6} \times T^2) + (-1,15 \times 10^{-2} \times T) + 1,8639 \quad \text{volt}$$

Formula 3.7: Calcolo della tensione di uscita per il sensore LM19

$$T = -1481,96 + \sqrt{2,1962 \times 10^6 + \frac{1,8639 - V_o}{3,88 \times 10^{-6}}} \quad ^\circ\text{C}$$

Formula 3.8: Temperatura in funzione della tensione di uscita per il sensore LM19

Per quanto riguarda il sensore di temperatura esterna e quello del componente di maggior potenza, una possibile scelta è il B59901, caratterizzato dall'aver un'aletta per essere messo a stretto contatto con la superficie di cui si vuole misurare la temperatura (vedi Figura 3.17); si tratta di un termistore di tipo PTC (*Positive Temperature Coefficient*).

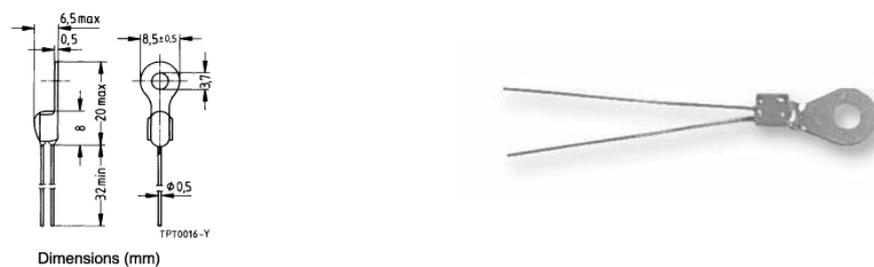


Figura 3.17: PTC B59901

La sezione di *housekeeping* fa inoltre uso del *timer1* presente all'interno del μC in esame. Tale modulo viene impiegato per le cadenze delle acquisizioni dei dati di *telemetria*. Infatti, opportunamente configurato, mi consente di ottenere degli intervalli di tempo di un secondo. Ogni volta che il *timer* in esame va in *overflow*, esso mi scatena un *interrupt* il quale mi incrementa un'opportuna variabile presente nella memoria RAM del μC . Quest'ultimo, una volta che la variabile ha raggiunto uno dei valori indicati nella Tabella 1.1, avvia la rispettiva procedura di acquisizione dei dati di *telemetria*. Affinché si possa utilizzare il *Timer 1* è necessario un ulteriore quarzo connesso tra i pin 15 e 16; l'uso di tale quarzo è del tutto indipendente e ininfluente con il quarzo necessario al funzionamento del resto del μC . A tal proposito, occorre procedere con cautela: generalmente i quarzi da 32.768 Hz, sono del tipo *tuning fork* (Figura 3.18 e 3.19). Tali quarzi però, nel nostro caso, sono molto delicati in quanto potrebbero non riuscire a sopportare le accelerazioni a cui sarà sottoposto il satellite al momento del lancio. Nello schema di Figura 3.15, è stato indicato il quarzo della C-MAC XTAL003004 (visibile nelle Figure 3.20 e 3.21) il quale, secondo il *datasheet*, sembrerebbe non essere del tipo *tuning fork*. Una possibile alternativa è quella di operare una divisione di frequenza a partire dal *clock* che scandisce i cicli macchina del μC .

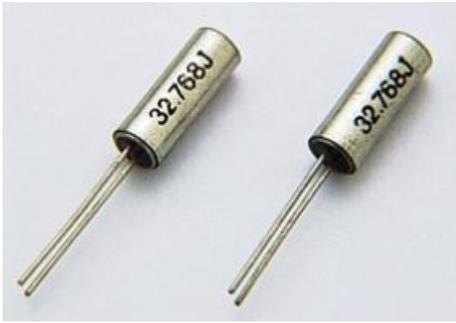


Figura 3.18: Quarzi da 32.768 Hz

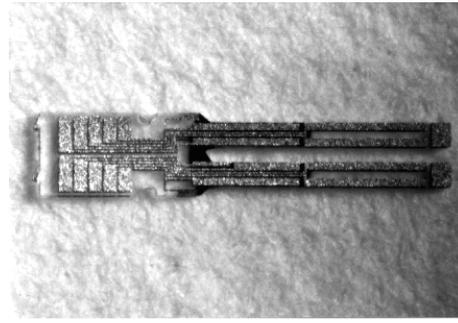


Figura 3.19: Tuning fork (interno del quarzo di Figura 3.18)



Figura 3.20: Quarzo C-MAC XTAL003004

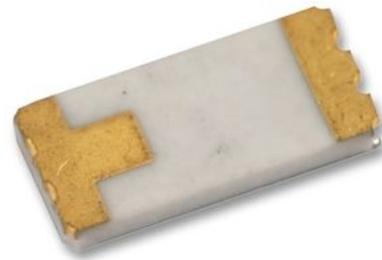


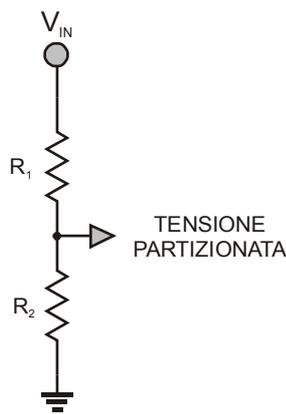
Figura 3.21: Contenuto del quarzo di Figura 50

Ai pin 6 e 7 del μC , sono collegati rispettivamente i segnali digitali U1 SHUTDOWN e U2 SHUTDOWN. Nel momento in cui occorre effettuare le misure di corrente, il μC agisce su questi pin in modo da abilitare gli integrati MAX471 ovvero fargli uscire dalla modalità *power down mode* (modalità di *default*). Una volta effettuata la misura, il μC provvede a reimpostare la modalità di *default*.

Al pin 4, nominato RA2, è collegato infine l'ingresso digitale che mi informa del senso della corrente della batteria (se è in carica o in scarica). Quando occorrerà determinare tale senso, il μC provvederà a testare il livello logico della linea, senza procedere ad acquisizioni.

Poiché l'ADC del μC ha un *reference* di +3,3V, e si vogliono acquisire tensioni di valore più elevato, è necessario operare una partizione della tensione. In particolare vengono progettati i partitori di tensione in modo tale da ottenere, all'ingresso dell'ADC, una tensione di +3V. Tale scelta è stata effettuata in modo che, nel caso si

verifichino delle oscillazioni delle tensioni (entro i $\pm 0,3V$, pari ad una variazione di circa 6,5 bit), l'ADC riesca ad apprezzarle. Se infatti avessimo tarato i partitori in modo tale da avere +3,3V sul convertitore, un'oscillazione positiva della tensione di una delle alimentazioni, non sarebbe stata percepita o, peggio, avrebbe danneggiato la porta di ingresso del μC . Di seguito vengono riportati i valori delle resistenze di partizione, le quali devono essere precise all'1% in modo da avere un preciso rapporto di partizione.



CALCOLO PER LA TENSIONE DEI PANNELLI

$$V_{IN} = + 7V$$

$$V_{PARTIZIONATA} = \frac{R_2}{R_1 + R_2} \times V_{IN}$$

$$\Rightarrow \frac{V_{IN}}{V_{PARTIZIONATA}} = \frac{1}{\frac{R_2}{R_1 + R_2}} = \frac{R_1 + R_2}{R_2}$$

$$\Rightarrow \frac{V_{IN}}{V_{PARTIZIONATA}} = 1 + \frac{R_1}{R_2} \Rightarrow \frac{7}{3} - 1 = \frac{R_1}{R_2}$$

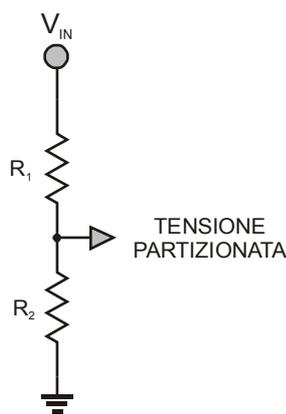
$$\Rightarrow \frac{R_1}{R_2} = 1,33\bar{3}$$

Scegliendo i valori commerciali abbiamo:

$$R_1 = 20K\Omega$$

$$R_2 = 15K\Omega$$

Effettuando il medesimo calcolo per le altre tensioni otteniamo:



CALCOLO PER LA TENSIONE DELLA BATTERIA

$$V_{IN} = + 10,8V$$

$$\Rightarrow \frac{V_{IN}}{V_{PARTIZIONATA}} - 1 = \frac{R_1}{R_2} \Rightarrow \frac{10,8}{3} - 1 = \frac{R_1}{R_2}$$

$$\Rightarrow \frac{R_1}{R_2} = 2,6$$

Scegliendo i valori commerciali abbiamo:

$$R_1 = 33,2K\Omega$$

$$R_2 = 12,7K\Omega$$

CALCOLO PER LA TENSIONE DEI +5V

$$V_{IN} = +5V$$

$$\Rightarrow \frac{V_{IN}}{V_{PARTIZIONATA}} - 1 = \frac{R_1}{R_2} \Rightarrow \frac{5}{3} - 1 = \frac{R_1}{R_2}$$

$$\Rightarrow \frac{R_1}{R_2} = 0,66\bar{6}$$

Scegliendo i valori commerciali abbiamo:

$$R_1 = 8,06K\Omega \quad R_2 = 12,1K\Omega$$

CALCOLO PER LA TENSIONE DEI +3,3V

$$V_{IN} = +3,3V$$

$$\Rightarrow \frac{V_{IN}}{V_{PARTIZIONATA}} - 1 = \frac{R_1}{R_2} \Rightarrow \frac{3,3}{3} - 1 = \frac{R_1}{R_2}$$

$$\Rightarrow \frac{R_1}{R_2} = 0,1$$

Scegliendo i valori commerciali abbiamo:

$$R_1 = 1K\Omega \quad R_2 = 10K\Omega$$

Una volta raccolti i dati di *telemetria*, il μC dovrà memorizzarli. La sezione di *housekeeping* comprende infatti una memoria (“condivisa” con la *sezione di data handling*), in cui ordinare i dati secondo una logica ben precisa (vedi §5.4.1). Tale memoria è visibile nello schema generale riportato al termine dell'elaborato.

Prima però della memorizzazione dei primi dati di *telemetria* di ciascuna orbita, il μC MASTER provvederà ad inviare al μC SLAVE la richiesta di data e ora in quel particolare istante. In questo modo si tiene traccia dell'evoluzione nel tempo delle grandezze di *telemetria*.

Per quanto riguarda la memoria residua e l'occupazione di memoria da parte dei dati strumentali, si rimanda al Capitolo 4 e al paragrafo 5.5.

3.4 Assetto

Al momento attuale la sezione di assetto è del tutto assente. Come riportato nella Figura 3.12, vi sono alcuni pin del μC inutilizzati. Uno di questi è una linea di *interrupt* (INT1, pin 34), riservata per la sezione in esame, nel caso in cui quest'ultima debba comunicare con una certa priorità con il μC (la comunicazione sul pin scatena un *interrupt* nel μC).

3.5 Sezione di Trasmissione/Ricezione

La sezione di TX/RX, come dice il nome, si occupa della trasmissione e della ricezione dei dati. Tale sezione, comunica con il $\mu\text{C}_{\text{MASTER}}$ tramite il bus SPI (vedi §4.2.2 e §4.2.1) nonché tramite i pin 18, 23 e 24 di quest'ultimo rispettivamente SCLK, SDI e SDO). Il bus SPI è un bus condiviso quindi occorre definire delle regole affinché non si verifichino le collisioni. Per questo motivo, è stato deciso che il μC della sezione di *housekeeping* svolgerà il ruolo di MASTER, ovvero colui che decide quale sotto sistema può usufruire del bus e quando gli è consentito farlo. Ciascun sotto sistema inoltre, prima di usufruire del bus, dovrà interpellare il $\mu\text{C}_{\text{MASTER}}$; sarà il μC a sua volta a stabilire se acconsentire o negare il permesso. Come si può notare dallo schema generale riportato al termine dell'elaborato, il bus SPI è condiviso non solo con i sottosistemi della sezione di TX/RX, ma anche con quelli della sezione di *data handling*, nonché con la memoria e un DAC. Al momento attuale si è appreso che l'integrato CC1100, ovvero colui che si occupa sia di trasmettere che di ricevere i dati, può comunicare con il $\mu\text{C}_{\text{MASTER}}$ tramite il bus precedentemente nominato. Le modalità con cui i due sistemi colloquiano, invece, non sono al momento definite, in quanto l'integrato in esame rappresenta un lavoro di tesi attualmente in corso di svolgimento.

In ogni caso, nel momento in cui occorre inviare a Terra i dati, è il $\mu\text{C}_{\text{MASTER}}$ che si occupa della costruzione dei pacchetti secondo il protocollo AX.25, scelto come protocollo di comunicazione a livello *data link* in quanto risulta essere un protocollo di tipo amatoriale.

Ci limitiamo quindi ad abbozzare uno schema di collegamento come riportato nella figura sottostante.

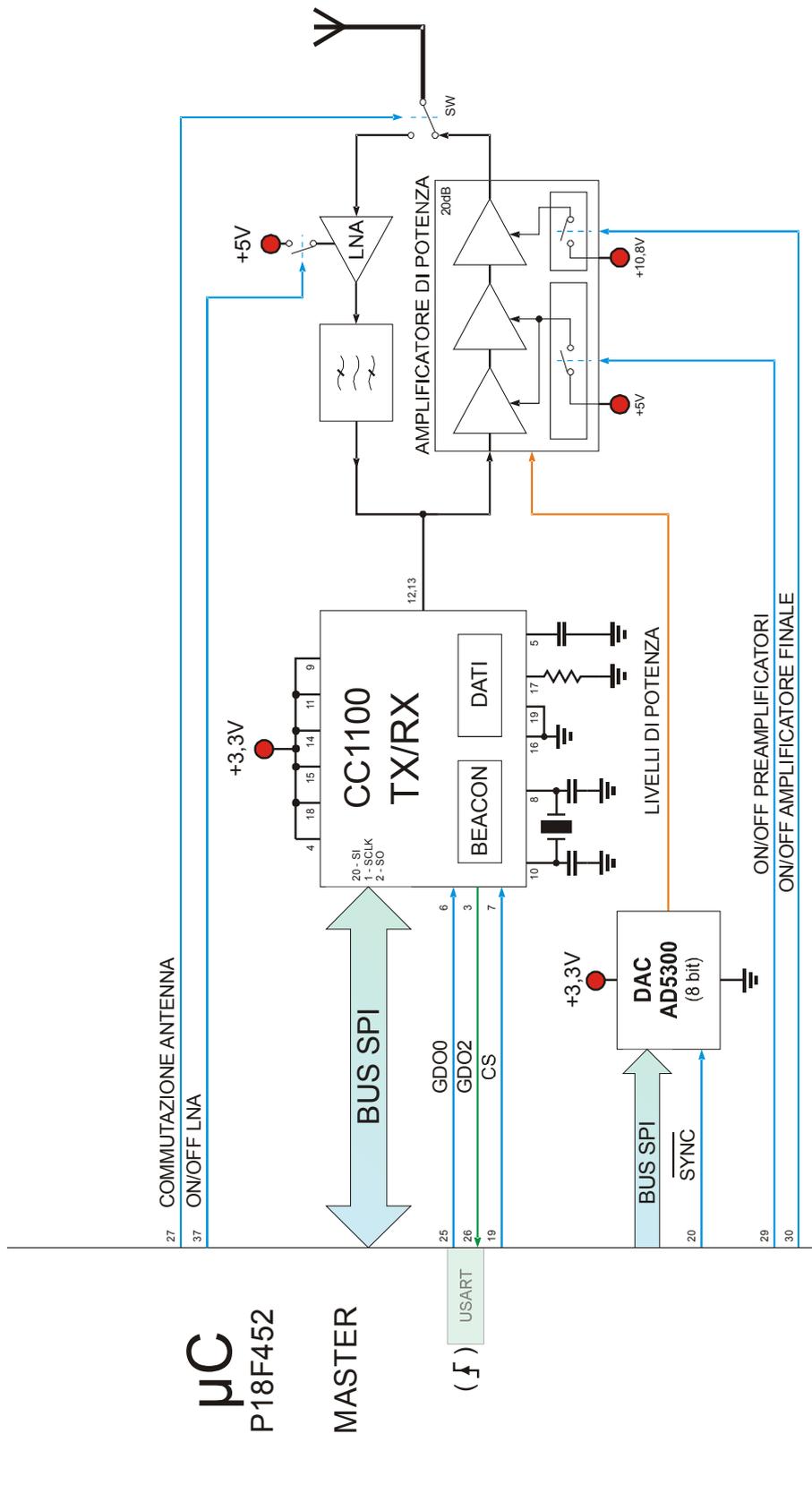


Figura 3.22: Sezione di TX/RX

La sezione di TX/RX è quindi composta dai seguenti elementi:

- l'integrato **CC1100**
- l'**AMPLIFICATORE DI POTENZA**
- il **DAC AD5300**
- lo *switch* **SW** (a diodi PIN)
- l'**ANTENNA**
- l'**LNA** (*Low Noise Amplifier*)
- il **FILTRO PASSA BANDA**

La linea RD4 (pin 27) del μC va a comandare lo *switch* SW. Questo commuta alternativamente l'antenna dal ramo inferiore (costituito dall'amplificatore di potenza) a quello superiore (costituito dall'LNA e dal filtro passa banda) a seconda che siamo rispettivamente in trasmissione o in ricezione. Se siamo in condizione di ricezione, e lo *switch* è commutato sul ramo inferiore, siamo nella situazione di MUTE ovvero si riceve il rumore presente all'ingresso dell'LNA in assenza del segnale.

I circuiti di alimentazione dell'amplificatore di potenza, sono controllati anch'essi dal $\mu\text{C MASTER}$ tramite le porte RD6 e RD7 (pin 29 e 30). Esso, tramite tali linee, provvede ad accendere e spegnere in maniera sequenziale gli stadi che compongono l'amplificatore, vale a dire il preamplificatore e lo stadio di potenza. Tra le due accensioni e spegnimenti sequenziali, è necessario un intervallo di tempo adatto: a tal proposito viene utilizzato il *timer 0* del μC . L'alimentazione in sequenza è necessaria per evitare che uno stadio venga pilotato prima che la sua porta di uscita sia chiusa sulla sua impedenza di carico. Questa procedura, durante la fase di transizione da acceso a spento e viceversa, evita disadattamenti di impedenza e quindi riflessioni indesiderate. Il controllo delle linee di alimentazione dei vari stadi può essere ottenuto tramite alcuni transistori di media – bassa potenza.

Riguardo all'integrato CC1100, è già stato detto che comunica con il μC tramite il bus SPI. L'integrato in esame è inoltre dotato di tre ulteriori pin: GDO2, GDO0 e CS. I primi due sono collegati alla porta USART del μC ; essi possono essere configurati in diversi modi, uno dei quali come linea di segnalazione dell'arrivo di un pacchetto di dati. Sfruttando quest'ultima caratteristica, risulta saggio collegare una delle due linee (bisogna ancora capire come lavora il CC1100) ad una linea di *interrupt* del $\mu\text{C MASTER}$ (INT2). Il CS (*chip select*) è necessario al μC per selezionare l'integrato in esame.

Si è detto che tale integrato viene utilizzato sia come trasmettitore sia come ricevitore. Siccome occorre un certo intervallo di tempo affinché l'integrato CC1100 passi dalla modalità trasmissione a quella ricezione (i settaggi dell'integrato dovranno essere effettuati tramite il bus SPI ad opera del $\mu\text{C}_{\text{MASTER}}$), la commutazione dello *switch* deve avvenire in modo tale che il segnale ricevente non saturi il CC1100 prima che la configurazione di quest'ultimo non si stia conclusa; condizioni analoghe possono essere dettate per il passaggio da ricezione a trasmissione. Occorre quindi seguire determinate procedure affinché non si verifichi nessun problema. Le rappresentazioni sottostanti chiariscono le fasi da rispettare.

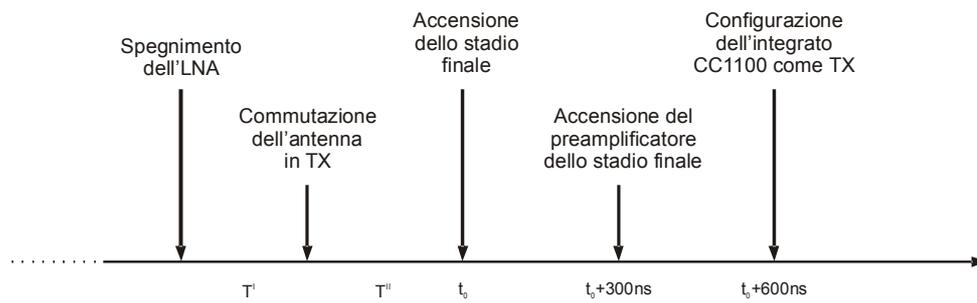


Figura 3.23: Procedure per abilitare la modalità TX

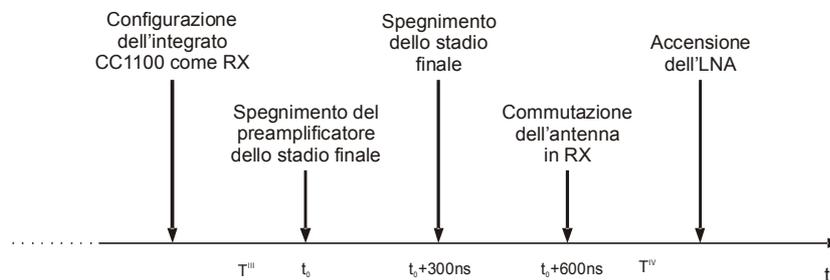


Figura 3.24: Procedure per abilitare la modalità RX

I 300ns riportati nelle figure, rappresentano il tempo necessario ai transistor per stabilizzarsi.

Per quanto riguarda i tempi T^I e T^{III} , essi possono essere di breve durata. I tempi invece T^{II} e T^{IV} , devono essere tali che consentano allo *switch* di stabilizzarsi.

Lo stadio di potenza amplifica di 20 dB il segnale applicato al suo ingresso. Tramite il DAC AD5300 (a 8 bit), è possibile generare fino a 255 livelli di potenza trasmessa, in quanto il segnale generato da tale integrato va a comandare l'amplificatore di potenza agendo sul suo ramo di retroazione. Anche l'AD5300 comunica sul bus SPI, ma la comunicazione avviene solo in un senso ovvero l'integrato è passivo. Esso è dotato di un pin, denominato come \overline{SYNC} , che funge da *chip select* ed è connesso al pin 20 del μ C MASTER. Da notare che, conoscendo il legame che esiste tra il valore del segnale in uscita dell'AD5300 e la potenza in uscita dell'amplificatore di potenza, siamo in grado di calcolare facilmente il dato di *telemetria* POTENZA MEDIA UTILIZZATA NELLE PRECEDENTI TRASMISSIONI.

3.6 Sezione di *data handling*

Il componente che si occupa dell'acquisizione dei dati strumentali e del loro processo di immagazzinamento, è il μ C PIC 18F252, denominato come SLAVE. Esso preleva, a cadenze regolari e modificabili tramite *telecomando*, i dati del GPS, del Dosimetro e del Magnetometro; tramite la memoria inoltre, indicata come MEMO, è possibile l'immagazzinamento dei dati prodotti.

Di seguito è riportato lo schema della sezione in esame.

Il GPS e il dosimetro, poiché possono comunicare con l'esterno in modo seriale, e con livelli logici TTL e CMOS, vengono connessi alla porta USART del $\mu\text{C SLAVE}$, descritta nel §3.1.1.5, tramite l'interposizione nel mezzo dell'integrato DG9415. Quest'ultimo, tramite un opportuno comando (linea SELEZIONE STRUMENTO), svolge la funzione di commutare i segnali di TX e RX del $\mu\text{C SLAVE}$ sullo strumento di interesse. Nel caso in cui si dovrà prelevare i dati del GPS, i deviatori all'interno dell'integrato DG9415 saranno quindi commutati nella posizione tale da permettere la continuità della linea TX del GPS con quella RX del $\mu\text{C SLAVE}$, nonché la continuità della linea RX del GPS con quella TX del $\mu\text{C SLAVE}$. Siccome il dosimetro, per come è fatto al suo interno, può comunicare solo attraverso lo standard RS232, è stato inserito, nel collegamento con il $\mu\text{C SLAVE}$ tra lo strumento e l'integrato DG9415, un traslatore di livello (MAX3233) il quale converte i segnali (0 - 3,3V) che stanno a monte (lato μC) in quelli consoni allo standard di comunicazione con lo strumento ($\pm 12\text{V}$). Il consumo dell'integrato che opera la commutazione delle linee, ammonta a $1\mu\text{A}$ (con una tensione di alimentazione di 3V).

Per quanto riguarda il formato dei pacchetti con cui colloquiano i due strumenti presi in esame fino ad adesso, sarà il $\mu\text{C SLAVE}$ ad occuparsi della loro interpretazione nonché, nel caso in cui il GPS e il dosimetro debbano essere interpellati, della loro costruzione. Inoltre, per ciò che concerne i dati inviati dal GPS, il $\mu\text{C SLAVE}$ dovrà discriminare i dati utili da quelli superflui in modo da limitare l'occupazione di memoria (caso dei dati di ora e data da porre nella TME).

Il collegamento tra $\mu\text{C SLAVE}$ e il magnetometro è reso possibile tramite l'inserimento di uno *switch* (ADG736) e un convertitore AD a 16 bit. Il magnetometro infatti fornisce in uscita tre segnali analogici (X, Y e Z); essi vengono fatti convogliare all'ingresso dell'integrato ADS8343. Quest'ultimo, al suo interno, è dotato di un multiplexer (a quattro ingressi), il quale consente di commutare i segnali provenienti dal magnetometro all'ingresso del convertitore AD. La commutazione dei segnali avviene per mezzo di un messaggio inviato dal $\mu\text{C SLAVE}$ tramite il bus SPI (il formato del messaggio è indicato nel *datasheet* e descritto in §4.3.4).

L'integrato ADG736, comandabile tramite la linea digitale del $\mu\text{C SLAVE}$, è stato inserito in modo da accedere al magnetometro anche se è in atto la comunicazione sul bus da parte del $\mu\text{C MASTER}$ (con il CC1100 oppure con la memoria o il DAC). Nel momento in cui il $\mu\text{C MASTER}$ deve comunicare con il $\mu\text{C SLAVE}$, questi porterà a livello logico alto la linea SEGNALE PER AVVERTIRE DELLA COMUNICAZIONE SUL BUS costringendo il $\mu\text{C SLAVE}$ ad interrompere qualsiasi attività e ad orientare il deviatore verso il $\mu\text{C MASTER}$. Una volta terminato l'invio, il $\mu\text{C SLAVE}$ continuerà l'eventuale procedura precedentemente sospesa. Nel normale funzionamento il deviatore ADG736 è orientato verso il magnetometro.

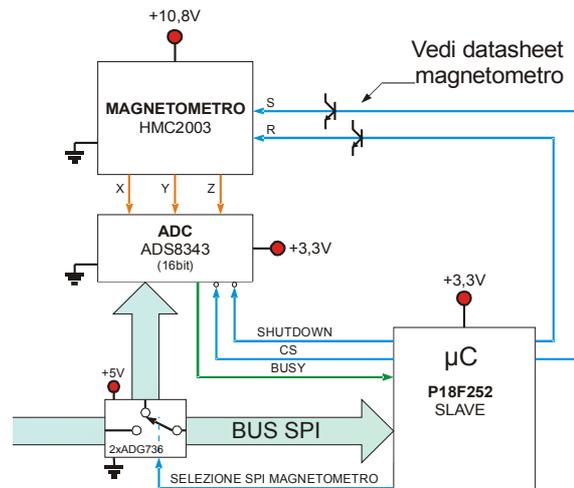


Figura 3.26: Collegamento tra μC SLAVE e magnetometro

Nel caso in cui si decida di non ricorrere all'integrato ADG736 (considerando quindi la Figura 3.8), nell'intervallo di tempo in cui il bus è utilizzato dal μC MASTER non potrei effettuare alcuna misura di campo magnetico; ciò causa un errore nella terna di valori ottenuta dai dati strumentali, in quanto, tra l'istante in cui occorre effettuare la misura e quello in cui si riottiene la disponibilità del bus, il satellite ha compiuto una certa distanza: la misura effettuata non corrisponderebbe alle coordinate ricavate dal GPS. Si potrebbe quindi pensare di rimandare tutte e tre le misure al riottenimento della disponibilità del bus: così facendo non si avrebbe più l'uniformità nel tempo delle misure. Con questo modo di operare dilazionatorio, si giunge alla conclusione che tutte le misure potrebbero essere effettuate dal μC MASTER; tale conclusione non può essere però realizzata in quanto il numero di pin dell'integrato P18F452 (il μC MASTER) sono limitati (non sono ancora state considerate le linee dedicate all'assetto in quanto quest'ultimo non è ancora stato definito).

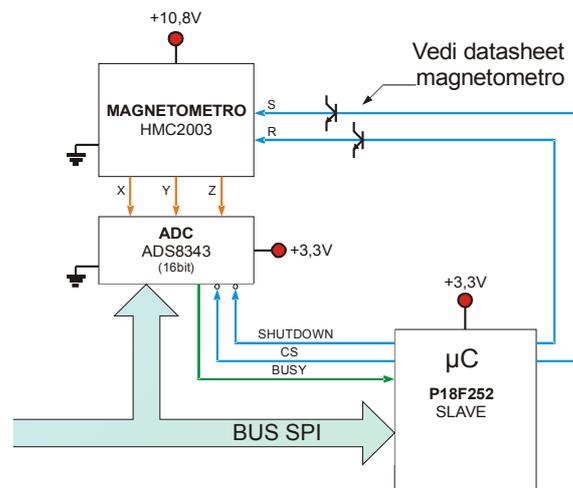


Figura 3.27: Alternativa del collegamento tra $\mu\text{C SLAVE}$ e magnetometro

Siccome abbiamo a che fare con un bus SPI, è presente il comando di *chip select* il quale è attivo a livello logico basso.

Anche l'ADS8343 inoltre è dotato di un pin denominato SHUTDOWN, attivo a livello logico basso; questo mi consente di abilitare la modalità basso consumo, caratterizzata da un assorbimento di corrente pari a $3\mu\text{A}$.

Dal *datasheet* del magnetometro, si apprende inoltre la necessità di utilizzare due ulteriori pin del $\mu\text{C SLAVE}$ per operare il SET e RESET dello strumento. Tali linee sono indicate con le lettere S e R.

Nel *datasheet* dello strumento in esame, è proposto un possibile circuito atto a svolgere i compiti di SET e RESET.

La cadenza con cui il $\mu\text{C SLAVE}$ deve effettuare il prelievo dei dati dagli strumenti di bordo, è fissato ed è pari a 237 secondi. L'intervallo in questione è modificabile tramite l'invio di un opportuno *telecomando*, ma non può scendere al di sotto del precedente valore, in quanto sarebbero prodotti dati che non verrebbero mai inviati a Terra a causa del limitato *tempo di accesso*. Nel momento in cui il $\mu\text{C MASTER}$ riceve l'invio di tale *telecomando* dalla stazione di Terra, questo viene trasmesso al $\mu\text{C SLAVE}$ attraverso il bus SPI. Prima di effettuare il trasferimento, il $\mu\text{C MASTER}$ informa il $\mu\text{C SLAVE}$ della trasmissione sul bus portando a livello logico alto la linea denominata SEGNALE PER AVVERTIRE DELLA COMUNICAZIONE SUL BUS (visibile nella Figura 3.25). Per quanto riguarda l'intervallo di tempo che intercorre tra un'acquisizione e l'altra, si fa uso del *Timer1* del $\mu\text{C SLAVE}$. Per esso valgono le stesse considerazioni fatte in §3.3.

Una volta acquisiti i segnali dagli strumenti di bordo, risulta necessario il trasferimento dalla memoria RAM del $\mu\text{C SLAVE}$ alla memoria di massa (indicata come MEMO). Siccome il $\mu\text{C SLAVE}$ non ha accesso diretto alla memoria, questi deve necessariamente inviare i dati al $\mu\text{C MASTER}$, il quale si prenderà l'incarico di memorizzarli. Affinché la memorizzazione avvenga, il $\mu\text{C SLAVE}$ deve necessariamente avvisare il $\mu\text{C MASTER}$ attraverso la linea INTERRUPT PER CHIEDERE LA COMUNICAZIONE SUL BUS (visibile nella Figura 3.25). Quest'ultimo, nel caso in cui stia utilizzando il bus, gli negherà il permesso, ma provvederà ad avvisare il $\mu\text{C SLAVE}$ della sua disponibilità nel momento in cui non dovrà più usufruirne. Si affida quindi il massimo controllo della memoria al $\mu\text{C MASTER}$. Tale scelta è stata effettuata in quanto può accadere che durante il *tempo di accesso*, che in generale ha una durata breve, occorra effettuare la misura dei dati strumentali. Se ciò accadesse, la memoria, nella quale sono contenuti anche i dati di *telemetria*, verrebbe occupata dal $\mu\text{C SLAVE}$: si avrebbe quindi un perdita di tempo in quanto, in quell'intervallo temporale, non si potrebbe inviare a Terra nessun dato.

Siccome il $\mu\text{C SLAVE}$ deve effettuare in generale una misura ogni 237 secondi (intervallo di *default*), per il rimanente tempo risulterebbe inutilizzato. È possibile quindi mettere il μC in modalità SLEEP: così facendo si ha un risparmio energetico (il *timer1* non viene posto in *standby*). Il risveglio da tale modalità può essere causato o dall'*overflow* del *timer1* oppure dall'*interrupt* della linea SEGNALE PER AVVERTIRE DELLA COMUNICAZIONE SUL BUS.

Durante il trasferimento dei dati a Terra, il bus SPI è completamente sotto il controllo del $\mu\text{C MASTER}$. Può accadere che durante questo intervallo di tempo il $\mu\text{C SLAVE}$ debba effettuare l'acquisizione dei segnali degli strumenti di bordo, ed in particolare dal magnetometro (il quale necessita dell'utilizzo del bus SPI); in tale situazione il $\mu\text{C SLAVE}$ invierà la richiesta di utilizzo del bus al $\mu\text{C MASTER}$, il quale, una volta terminato il suo utilizzo, provvederà a comunicare al $\mu\text{C SLAVE}$ il consenso a prenderne il controllo. Può accadere quindi che durante l'intero *tempo di accesso* non venga effettuata nessuna misura di campo magnetico; tale misura viene rimandata al termine del *tempo* appena menzionato.

Per quanto riguarda la memoria di massa, è stata scelta la 25LC1024, la quale può immagazzinare 1Mbit equivalenti a 131.072 byte. Nel Capitolo 5, relativo alla stima della memoria necessaria, si constaterà se tale memoria sia sufficiente ai nostri scopi. In ogni caso è stato riservato un ulteriore pin del $\mu\text{C MASTER}$ (CS2, pin 22) in modo da poter interfacciare fino a quattro memorie di questo tipo.

Riguardo alla presente sezione, in questo lavoro di tesi sarà trattata solamente la procedura di colloquio tra $\mu\text{C MASTER}$ e $\mu\text{C SLAVE}$ (e viceversa), in quanto la tecnica di prelievo dei dati strumentali, la discriminazione dei dati forniti dagli strumenti e il colloquio con gli stessi, è oggetto di un ulteriore lavoro di tesi portato avanti da un altro studente.

Per il calcolo del volume di dati prodotti dagli strumenti, si rimanda al §5.5.

Capitolo 4

Struttura interna di ATMOCUBE (livello software)

Dopo la panoramica effettuata nel capitolo precedente sulla struttura interna del satellite ATMOCUBE a livello *hardware*, si procede alla descrizione delle operazioni che avvengono all'interno dei componenti “dotati di intelligenza”. In particolare, nei capitoli che seguiranno, si stabiliranno le procedure atte a misurare le grandezze di *telemetria*, nonché consentire l'interazione tra i sottosistemi del satellite.

4.1 Acquisizione e tracciatura dei dati di *telemetria*

Nel §3.3, si è detto che alcune grandezze di *telemetria*, cioè tensioni, correnti e temperature, vengono acquisite tramite l'ADC del $\mu\text{C}_{\text{MASTER}}$. L'ordine di acquisizione delle grandezze non può essere alterato, quindi il $\mu\text{C}_{\text{MASTER}}$ provvederà ad operare la conversione analogico-digitale secondo la seguente scaletta:

- 1) Tensione dei pannelli solari
- 2) Tensione della batteria
- 3) Tensione dei +5V
- 4) Tensione dei +3,3V
- 5) Senso della corrente della batteria
- 6) Corrente dei pannelli solari
- 7) Corrente della batteria
- 8) Temperatura interna
- 9) Temperatura esterna
- 10) Temperatura del componente di maggior potenza

Nel caso in cui una delle acquisizioni di tali grandezze è stata disabilitata (vedi TME_CONFIG), si procede con quella immediatamente successiva.

La tracciatura della memoria residua e di quella dell'occupazione da parte dei dati strumentali, avviene invece ogniqualvolta si debba procedere all'accesso della memoria di massa.

4.1.1 Procedura di acquisizione dall'ADC

Nella fase di *boot* del satellite (ovvero al suo primo avvio), occorre configurare in modo opportuno i registri che interessano il modulo ADC. Nel §3.1.1.1 è già stata affrontata una descrizione generale di tale modulo; si procede ora ad alcune precisazioni in merito ai settaggi dei registri ADCON0 e ADCON1.

Dal *datasheet* del P18F452, si ricavano le configurazioni dei registri in esame; esse sono riportate nelle Figure 4.1 e 4.2.

REGISTRO ADCON0

| | | | | | | | |
|-------|-------|-------|-------|-------|---------|-----|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 |
| ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/DONE | — | ADON |
| | | | | | | | bit 0 |
| | | | | | | | bit 7 |

bit 7-6 **ADCS1:ADCS0**: A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

| ADCON1 <ADCS2> | ADCON0 <ADCS1:ADCS0> | Clock Conversion |
|-------------------|-------------------------|---|
| 0 | 00 | Fosc/2 |
| 0 | 01 | Fosc/8 |
| 0 | 10 | Fosc/32 |
| 0 | 11 | FRC (clock derived from the internal A/D RC oscillator) |
| 1 | 00 | Fosc/4 |
| 1 | 01 | Fosc/16 |
| 1 | 10 | Fosc/64 |
| 1 | 11 | FRC (clock derived from the internal A/D RC oscillator) |

bit 5-3 **CHS2:CHS0**: Analog Channel Select bits

- 000 = channel 0, (AN0)
- 001 = channel 1, (AN1)
- 010 = channel 2, (AN2)
- 011 = channel 3, (AN3)
- 100 = channel 4, (AN4)
- 101 = channel 5, (AN5)
- 110 = channel 6, (AN6)
- 111 = channel 7, (AN7)

Note: The PIC18F2X2 devices do not implement the full 8 A/D channels; the unimplemented selections are reserved. Do not select any unimplemented channel.

bit 2 **GO/DONE**: A/D Conversion Status bit

When ADON = 1:

- 1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)
- 0 = A/D conversion not in progress

bit 1 **Unimplemented**: Read as '0'

bit 0 **ADON**: A/D On bit

- 1 = A/D converter module is powered up
- 0 = A/D converter module is shut-off and consumes no operating current

Figura 4.1: Registro ADCON0

REGISTRO ADCON1

| | | | | | | | |
|-------|-------|-----|-----|-------|-------|-------|-------|
| R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| ADFM | ADCS2 | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | bit 0 | | | |

bit 7 **ADFM:** A/D Result Format Select bit
 1 = Right justified. Six (6) Most Significant bits of ADRESH are read as '0'.
 0 = Left justified. Six (6) Least Significant bits of ADRESL are read as '0'.

bit 6 **ADCS2:** A/D Conversion Clock Select bit (ADCON1 bits in **bold**)

| ADCON1 <ADCS2> | ADCON0 <ADCS1:ADCS0> | Clock Conversion |
|-------------------|-------------------------|---|
| 0 | 00 | Fosc/2 |
| 0 | 01 | Fosc/8 |
| 0 | 10 | Fosc/32 |
| 0 | 11 | FRC (clock derived from the internal A/D RC oscillator) |
| 1 | 00 | Fosc/4 |
| 1 | 01 | Fosc/16 |
| 1 | 10 | Fosc/64 |
| 1 | 11 | FRC (clock derived from the internal A/D RC oscillator) |

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits

| PCFG <3:0> | AN7 | AN6 | AN5 | AN4 | AN3 | AN2 | AN1 | AN0 | VREF+ | VREFR | / C |
|---------------|-----|-----|-----|-----|-------|-------|-----|-----|-------|-------|-------|
| 0000 | A | A | A | A | A | A | A | A | VDD | VSS | 8 / 0 |
| 0001 | A | A | A | A | VREF+ | A | A | A | AN3 | VSS | 7 / 1 |
| 0010 | D | D | D | A | A | A | A | A | VDD | VSS | 5 / 0 |
| 0011 | D | D | D | A | VREF+ | A | A | A | AN3 | VSS | 4 / 1 |
| 0100 | D | D | D | D | A | D | A | A | VDD | VSS | 3 / 0 |
| 0101 | D | D | D | D | VREF+ | D | A | A | AN3 | VSS | 2 / 1 |
| 011x | D | D | D | D | D | D | D | D | — | — | 0 / 0 |
| 1000 | A | A | A | A | VREF+ | VREF- | A | A | AN3 | AN2 | 6 / 2 |
| 1001 | D | D | A | A | A | A | A | A | VDD | VSS | 6 / 0 |
| 1010 | D | D | A | A | VREF+ | A | A | A | AN3 | VSS | 5 / 1 |
| 1011 | D | D | A | A | VREF+ | VREF- | A | A | AN3 | AN2 | 4 / 2 |
| 1100 | D | D | D | A | VREF+ | VREF- | A | A | AN3 | AN2 | 3 / 2 |
| 1101 | D | D | D | D | VREF+ | VREF- | A | A | AN3 | AN2 | 2 / 2 |
| 1110 | D | D | D | D | D | D | D | A | VDD | VSS | 1 / 0 |
| 1111 | D | D | D | D | VREF+ | VREF- | D | A | AN3 | AN2 | 1 / 2 |

A = Analog input D = Digital I/O

C/R = # of analog input channels / # of A/D voltage references

Figura 4.2: Registro ADCON1

Vediamo di analizzare brevemente il registro ADCON0: la sua configurazione iniziale è 00000000. Il modulo viene quindi posto inizialmente in *standby*, impostato per acquisire il segnale sul canale AN0 e settato con un *clock* di conversione pari a $fosc/2$.

Tramite i bit <CHS2:CHS0> si selezionano i canali (vedi Figura 3.3) quindi, nel nostro caso potremmo avere le sole due combinazioni 000 e 001 (vedi Tabella 3.1); affinché avvenga però l'acquisizione delle differenti tensioni, correnti e temperature, occorrerà comandare in modo opportuno le linee A0, A1 e A2 in uscita dal $\mu\text{C MASTER}$ in modo da far convogliare il segnale di interesse all'ingresso dell'ADC (vedi Figura 3.15).

Nel momento in cui viene settato (a “1”) il livello logico del bit ADON, viene avviata l'acquisizione. Quando invece il bit 2 viene resettato (posto a “0”) in modo automatico dal μC , significa che la conversione è terminata e che è possibile prelevare il dato dai registri ADRESH e ADRESL.

In merito a questi ultimi due registri, tramite il bit 7 del registro ADCON1, è possibile scegliere se adottare una giustificazione sinistra o destra (vedi Figura 4.3).

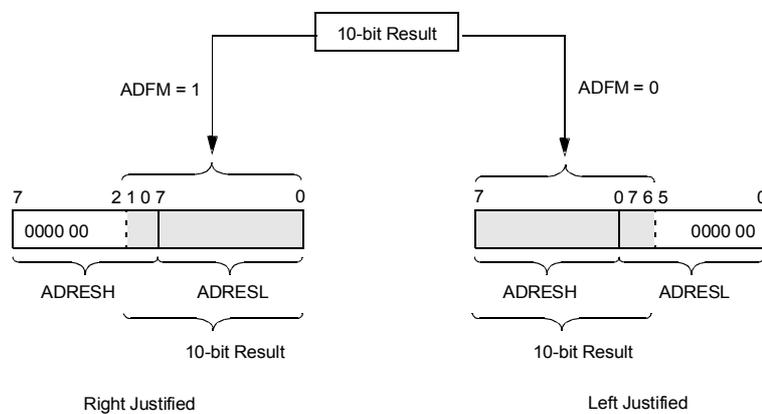


Figura 4.3: Giustificazione dei dati a sinistra e a destra

Nel nostro caso adotteremo una giustificazione destra pertanto il bit ADFM del registro ADCON1 sarà posto a 1; inoltre, come detto nel §3.3, vengono utilizzati solo due ingressi analogici. La configurazione di partenza del registro ADCON1 sarà quindi 10000100.

Ognuna delle grandezze elencate nel §4.1, ad eccezione del senso della corrente della batteria, viene acquisita con la medesima procedura; essa è esprimibile tramite il seguente diagramma di flusso:

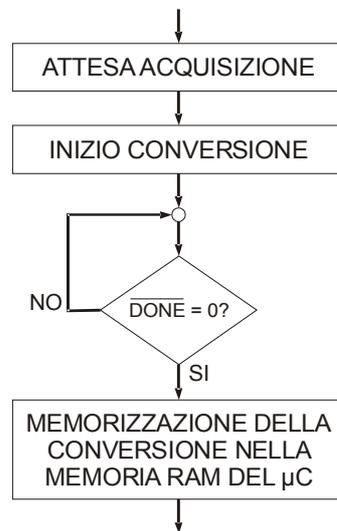


Figura 4.4: Procedura di acquisizione del canale

Per quanto riguarda il senso della batteria, il μC provvede semplicemente ad effettuare un test sul livello logico della linea RA2.

Una volta terminate tutte le acquisizioni, si procede alla memorizzazione nella memoria di massa. A tal proposito si può anticipare l'adozione di due tecniche differenti di memorizzazione: quella "standard" e quella differenziale (vedi §5.1.1 e §5.1.2). La procedura di memorizzazione viene anche detta *routine* di memorizzazione.

La fase di acquisizione di tutte le grandezze riportate nel §4.1, può essere descritta dal seguente diagramma temporale:

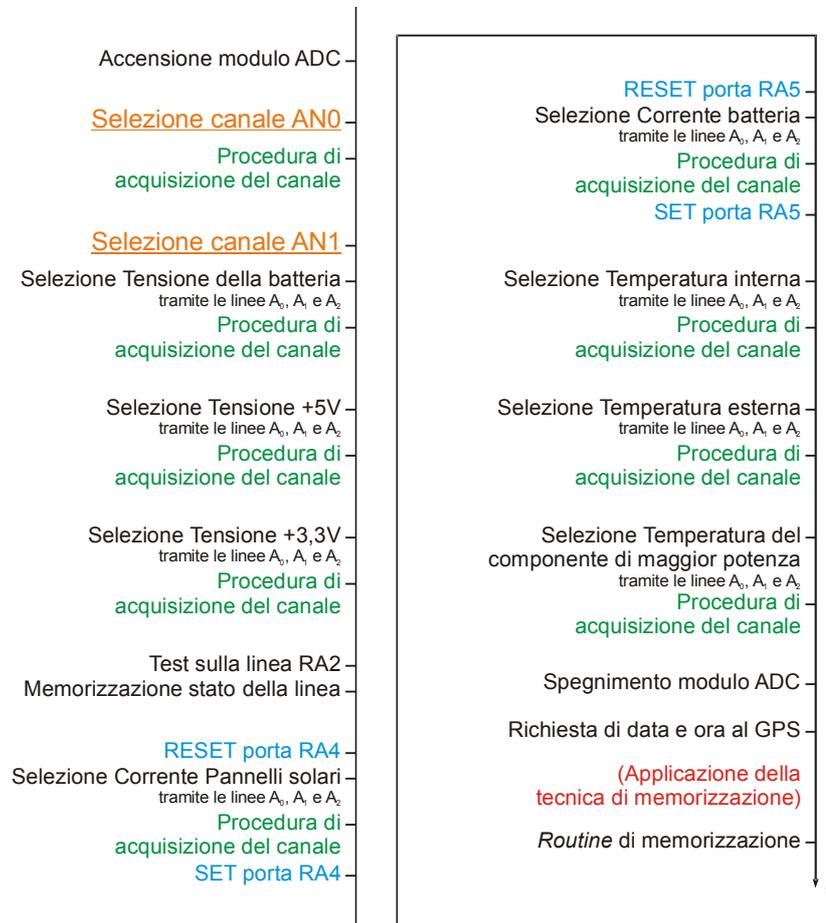


Figura 4.5: Procedura di acquisizione e memorizzazione tensioni, correnti e temperature

4.1.2 Tracciatura della memoria residua e dei dati strumentali

La memoria residua rappresenta il residuo totale della memoria di massa. Essa consiste in una variabile presente all'interno del μC_{MASTER} che viene continuamente aggiornata ogni qualvolta occorre accedere alla memoria per scrivere o cancellare le informazioni (sia di *telemetria* che dei dati strumentali).

Per quanto riguarda l'occupazione di memoria di tale dato, essa ammonta a 3 byte. Questa dimensione consente di conteggiare fino a $2^{24}=16.777.216$ byte. Due byte infatti, non sono sufficienti: in quanto è possibile conteggiare solo 65.536 locazioni di memoria. Tale valore è difatti una quantità inferiore a quella con cui si avrà a che fare.

Nel momento in cui avverrà la richiesta di memorizzazione del valore della memoria residua, il $\mu\text{C MASTER}$ preleverà il contenuto della variabile nominata precedentemente e, avviando la *routine* di memorizzazione (vedi Figura 4.7), lo collocherà nella memoria di massa, non prima però di averlo decrementato di 3 byte (occupazione di memoria del dato in esame).

Per quanto riguarda la memoria totale dei dati strumentali, anche in questo caso è presente nel $\mu\text{C MASTER}$ un'opportuna variabile che viene aggiornata ogniqualvolta il $\mu\text{C MASTER}$ riceve dal $\mu\text{C SLAVE}$ l'incarico di collocare nella memoria di massa i dati degli strumenti di bordo. Nel momento in cui è necessario memorizzare il dato di *telemetria* riguardante lo stato di tale variabile, il $\mu\text{C MASTER}$ preleva il suo contenuto e lo colloca, avviando la procedura di memorizzazione, nella memoria di massa. Per gli stessi motivi presentati prima, l'occupazione di memoria della variabile (quindi del dato di *telemetria*) ammonta a 3 byte.

4.1.3 Potenza media nelle trasmissioni e ricezioni

La potenza media - utilizzata per l'invio e la ricezione dei pacchetti - viene calcolata una volta terminato il *tempo di accesso*. Per ogni pacchetto infatti, viene determinato o misurato il livello di potenza utilizzato e sommato con il precedente; una volta terminato tale tempo, viene operata una media.

Per il calcolo della potenza utilizzata in trasmissione, nonché per quello della potenza misurata in ricezione, vengono utilizzati 4 byte della memoria RAM nel $\mu\text{C MASTER}$ (4 byte per la potenza in trasmissione e 4 byte per la potenza in ricezione). Due di questi byte sono destinati all'immagazzinamento della somma delle potenze mentre i restanti procedono al conteggio del numero di pacchetti. Una volta terminato il *tempo di accesso*, viene operata la media: il risultato finale viene posto in due ulteriori byte (della memoria RAM) e collocati al termine della TMB o della TME, a seconda dei settaggi effettuati tramite *telecomando* (vedi §1.5, §1.6 e il Capitolo 2).

4.1.4 Numero di accessi non autorizzati

Tale dato di *telemetria*, come già accennato nei capitoli precedenti, consiste in un contatore il cui contenuto rappresenta gli accessi effettuati al satellite ad opera di utenti a noi sconosciuti. Nell'accesso al satellite ATMOCUBE vi sarà infatti una fase di autenticazione, senza la quale non si potrà eseguire nessun comando e quindi nessuna trasmissione a Terra. Si suppone che durante il *tempo di attesa* (tempo nel quale vengono conteggiati gli accessi sconosciuti) non avvengano molti tentativi non autorizzati; si imposta quindi l'occupazione di memoria della variabile di conteggio a bordo del $\mu\text{C}_{\text{MASTER}}$ pari a 1 byte. Con tale dimensione è possibile conteggiare fino a 255 accessi. Supponendo che ne avvengano in numero maggiore, tale variabile, una volta incrementata, ricomincerebbe il conteggio partendo da zero: tramite il *firmware* del μC ciò viene impedito.

4.2 Interazione del $\mu\text{C}_{\text{MASTER}}$

Nei paragrafi che seguono, verranno espone le procedure cui deve sottostare il $\mu\text{C}_{\text{MASTER}}$ per comunicare con l'integrato CC1100, con la memoria di massa e con il $\mu\text{C}_{\text{SLAVE}}$. Prima di procedere, si vuole sottolineare il fatto che il $\mu\text{C}_{\text{MASTER}}$ è l'effettivo “padrone” del bus SPI, quindi è colui che decide, in seguito ad una richiesta, se consentire l'uso da parte degli altri sistemi.

Per quanto riguarda la comunicazione tra $\mu\text{C}_{\text{MASTER}}$ e l'integrato CC1100, al momento attuale non può essere definita in modo completo, in quanto il funzionamento dell'integrato è ancora sotto esame.

4.2.1 Interazione $\mu\text{C}_{\text{MASTER}} \rightarrow$ Memoria di massa

Il questo paragrafo verrà illustrata la comunicazione tra il $\mu\text{C}_{\text{MASTER}}$ e la memoria di massa nel momento in cui occorre memorizzare oppure prelevare i dati. La memoria utilizzata è la 25LC1024, prodotta dalla Microchip.

Verranno analizzati i casi di memorizzazione/lettura di un singolo e di una molteplicità di byte; verrà inoltre descritta anche la cancellazione delle locazioni di memoria.

Nei diagrammi che seguono, si presume che prima dell'invio dei dati, sia avvenuta la configurazione della memoria; tale procedura viene eseguita un'unica volta (nel *boot* del satellite) e varia da memoria a memoria.

4.2.1.1 Scrittura dei dati

Se il μC MASTER deve memorizzare un singolo byte, esso si dovrà attenere alla procedura seguente.

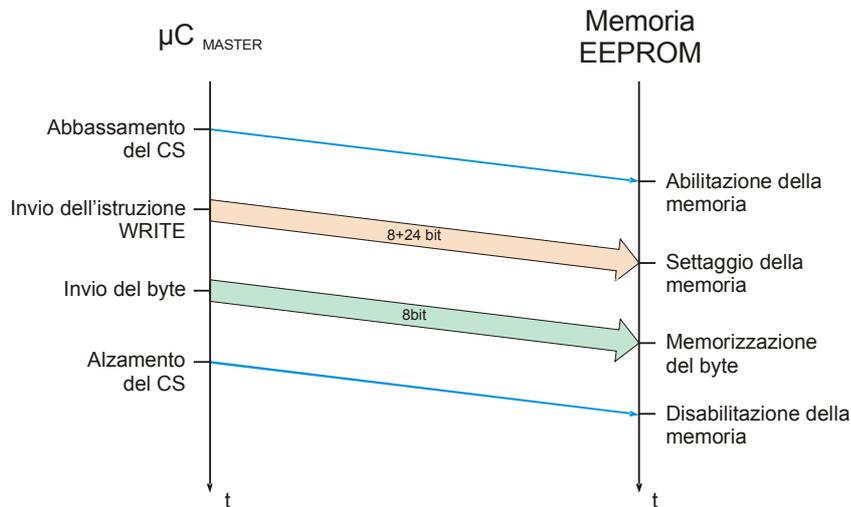


Figura 4.6: Memorizzazione di un singolo byte

L'abbassamento del livello del CS ad opera del μC MASTER avviene ogni qualvolta occorre accedere alla memoria; tale linea deve essere necessariamente rialzata nel momento in cui l'accesso non è più necessario (quando è stata terminata la scrittura, la lettura o la cancellazione). L'invio dell'istruzione WRITE, è composta da 8+24 bit: 8 bit identificano l'istruzione mentre 24 sono destinati all'identificazione della locazione di memoria nella quale memorizzare il dato (a 8 bit) inviato immediatamente dopo.

Tra una memorizzazione di un singolo byte ed un'altra dello stesso tipo, occorre attendere un preciso intervallo di tempo: la memoria infatti deve avere il tempo, oltre che per memorizzare il dato, anche per rendersi conto della fine di una scrittura e l'inizio di una nuova (nel caso della 25LC1024 tale tempo è pari a 5ms).

Nel caso in cui occorre invece inviare due o più byte consecutivi (generalmente è il nostro caso), la procedura avviene in modo simile alla precedente. Supponendo di dover memorizzare N byte consecutivi, abbiamo:

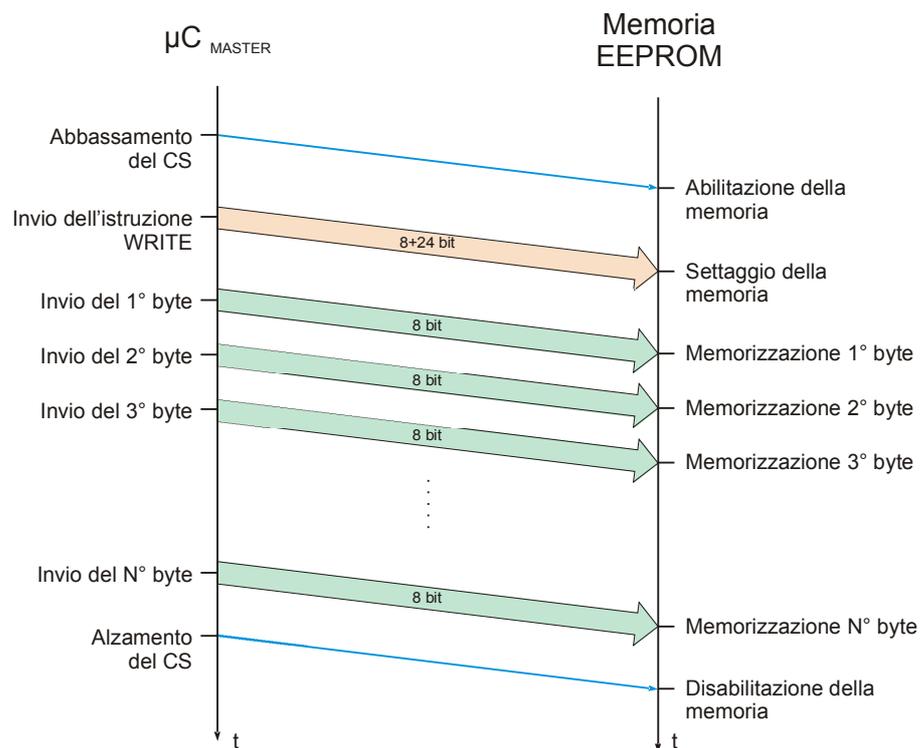


Figura 4.7: Procedura di memorizzazione di N byte consecutivi

Nel caso in cui N sia pari a 256, si ha la così detta “memorizzazione di pagina” (*page write*).

Come si osserva dalla Figura 4.7, l'abbassamento del CS (*chip select*) della memoria avviene solamente all'inizio dell'invio di tutti i dati da memorizzare, e non ad ognuno di questi. Inoltre, tramite l'istruzione WRITE è possibile memorizzare tutti i byte inviati a partire da una determinata locazione di memoria: in questo modo si ha un notevole risparmio di tempo, in quanto non occorre inviare ogni volta l'indirizzo della locazione di memoria.

Con la memoria presa in esame, è possibile operare una memorizzazione di massimo 256 byte consecutivi; nel caso i byte da memorizzare fossero in numero maggiore, occorrerà eseguire più volte la procedura di Figura 4.7.

Dal punto di vista dei segnali che viaggiano sulle linee di comunicazione SDI, SDO e CS, abbiamo i seguenti diagrammi:

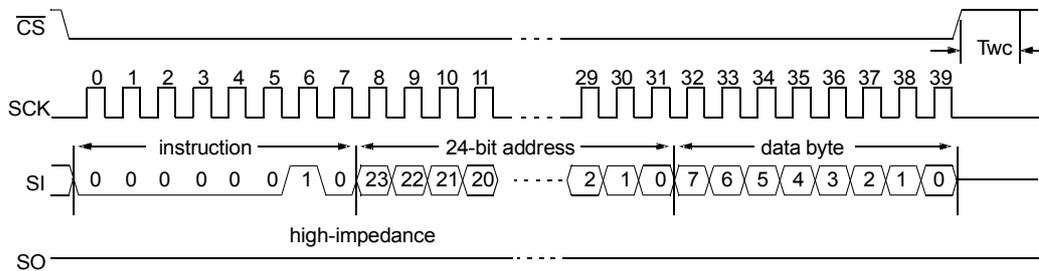


Figura 4.8: Segnali di memorizzazione di un singolo byte

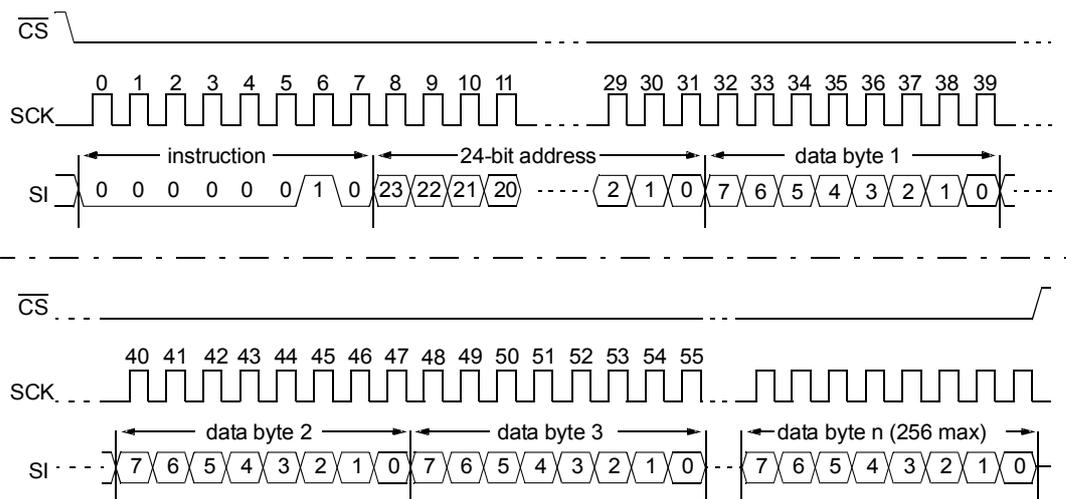


Figura 4.9: Segnali di memorizzazione di due o più byte (max 256)

4.2.1.2 Prelievo dei dati

Per quanto riguarda la ricezione dei dati, dopo l'abbassamento del CS e l'invio dell'istruzione READ (composta da 8+24 bit), la memoria mi restituisce tutti i dati contenuti in essa a partire dalla locazione indicata tramite l'istruzione. Per interrompere la trasmissione basta riportare a livello logico alto il CS. Il diagramma temporale è quindi il seguente:

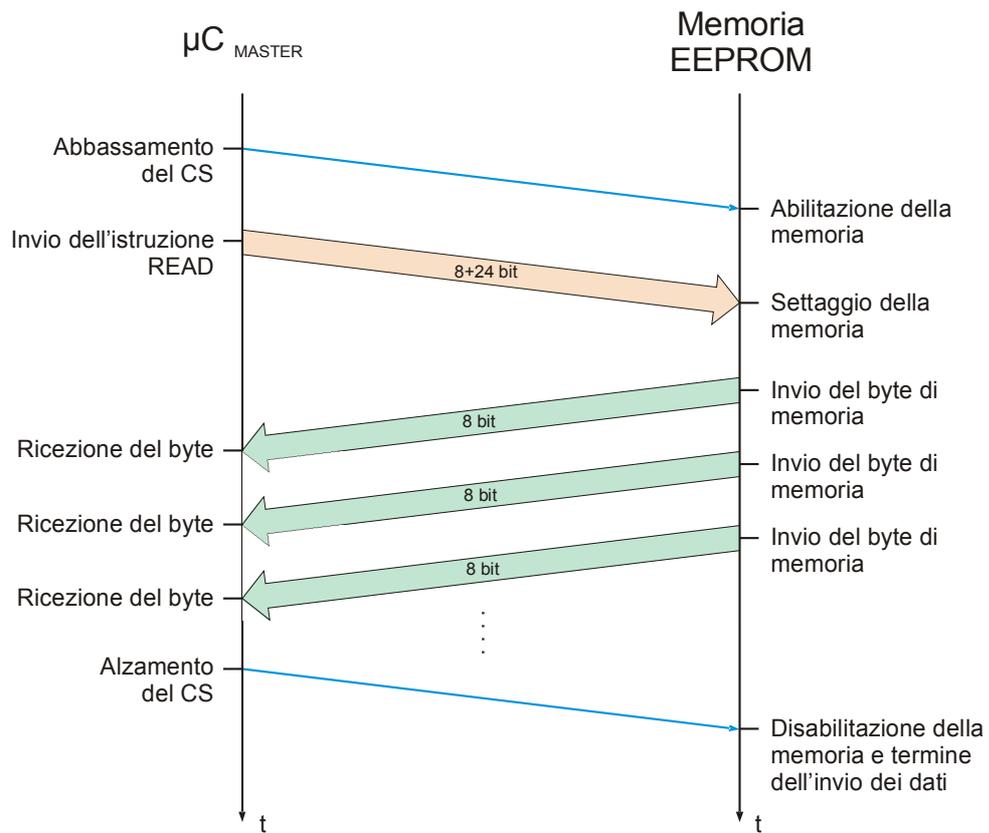


Figura 4.10: Prelievo di dati dalla memoria

I segnali presenti sulle linee di comunicazione avranno la seguente evoluzione temporale:

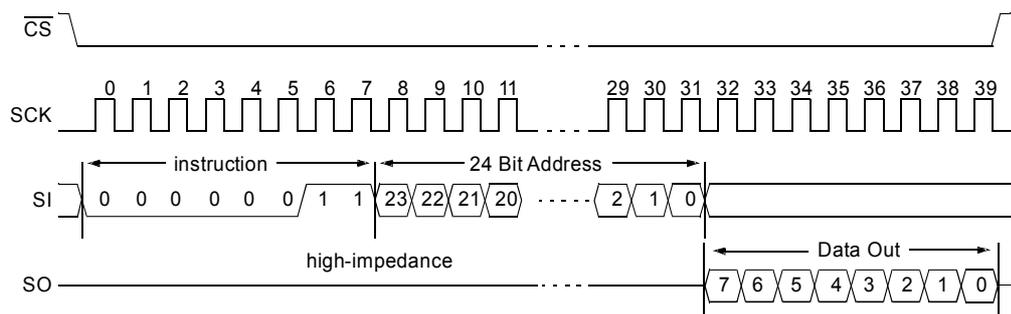


Figura 4.11: Segnali di prelievo dati dalla memoria

4.2.1.3 Cancellazione dei dati

Per quanto riguarda infine la cancellazione della memoria possiamo avere tre alternative:

- **cancellazione della page (*page erase*):** cancella 256 locazioni di memoria a partire da quella inviata (5ms) – vedi Figura 4.12;
- **cancellazione del settore (*sector erase*):** cancella il settore indicato dall'istruzione (2s) – vedi Figura 4.13;
- **cancellazione del chip (*chip erase*):** cancella l'intero contenuto del chip (4s) – vedi Figura 4.14;

Le sequenze di bit atte a far eseguire queste istruzioni sono rappresentate di seguito:

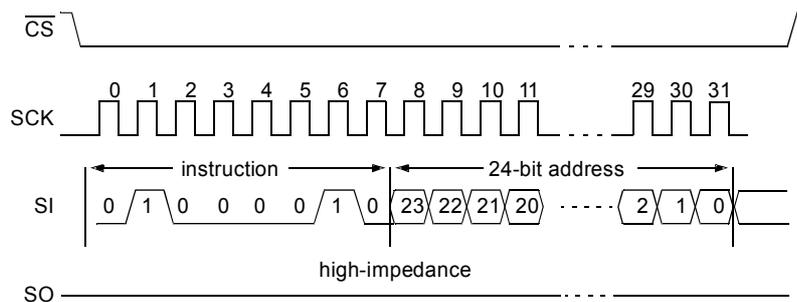


Figura 4.12: Istruzione di *page erase*

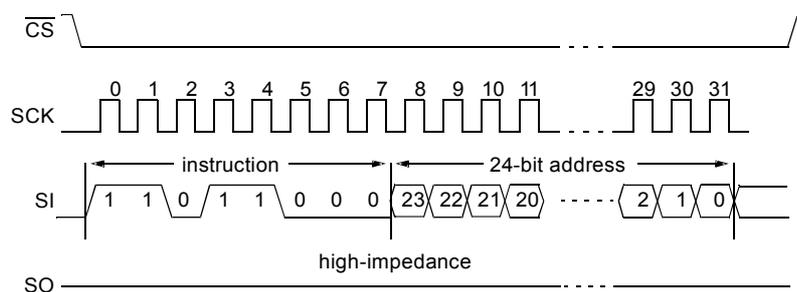


Figura 4.13: Istruzione di *sector erase*

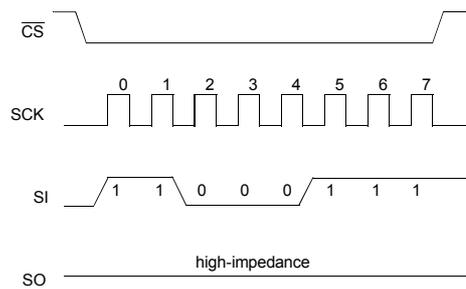


Figura 4.14: Istruzione di chip erase

4.2.2 Interazione $\mu\text{C MASTER} \rightarrow \text{CC1100}$ (configurazione)

Nel §3.5 è stato detto che l'integrato CC1100 svolge sia la funzione di trasmettitore, sia quella di ricevitore. Nel passaggio quindi da una modalità all'altra, occorre ogni volta configurare i suoi registri interni tramite l'invio di opportuni messaggi attraverso il bus SPI. Al momento attuale non si conosce la composizione del messaggio, in quanto l'integrato in esame è ancora in fase di studio.

Nel §3.1, è stato detto che i μC utilizzati hanno al loro interno una memoria EEPROM da 256 byte. Tale memoria ha una dimensione esigua per i nostri scopi, in quanto la *telemetria* e gli strumenti di bordo producono una quantità di dati maggiore. La memoria in esame può essere invece utilizzata per immagazzinare tutte le impostazioni di *default* del sistema; al momento della programmazione del μC , è possibile infatti inviare alla memoria in esame tali configurazioni e, tramite il bit $\text{EECON1}\langle\text{WREN}\rangle$, proteggerla da cancellazioni accidentali. Nel momento in cui si deve quindi procedere ad un ripristino delle configurazioni di *default*, tramite l'invio di un *telecomando*, il $\mu\text{C MASTER}$ preleverà il contenuto della sua memoria EEPROM e lo sostituirà ai suoi registri di configurazione (anche quelli creati da noi). Allo stesso modo può avvenire la riconfigurazione dell'integrato CC1100, nel momento in cui si deve passare dalla modalità trasmissione a quella ricezione.

La procedura di configurazione del CC1100 può essere rappresentata dai seguenti diagrammi temporali: uno per la configurazione come trasmettitore e uno per la configurazione come ricevitore. Nella tracciatura dei diagrammi si è tenuto conto delle Figure 3.23 e 3.24.

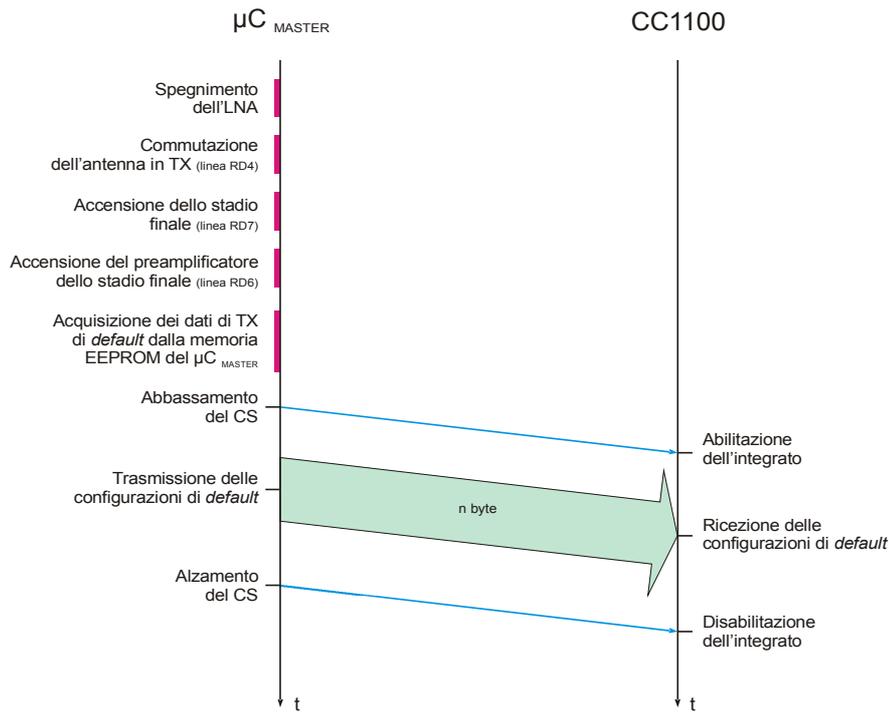


Figura 4.15: Procedura di invio delle configurazioni di default per la modalità TX

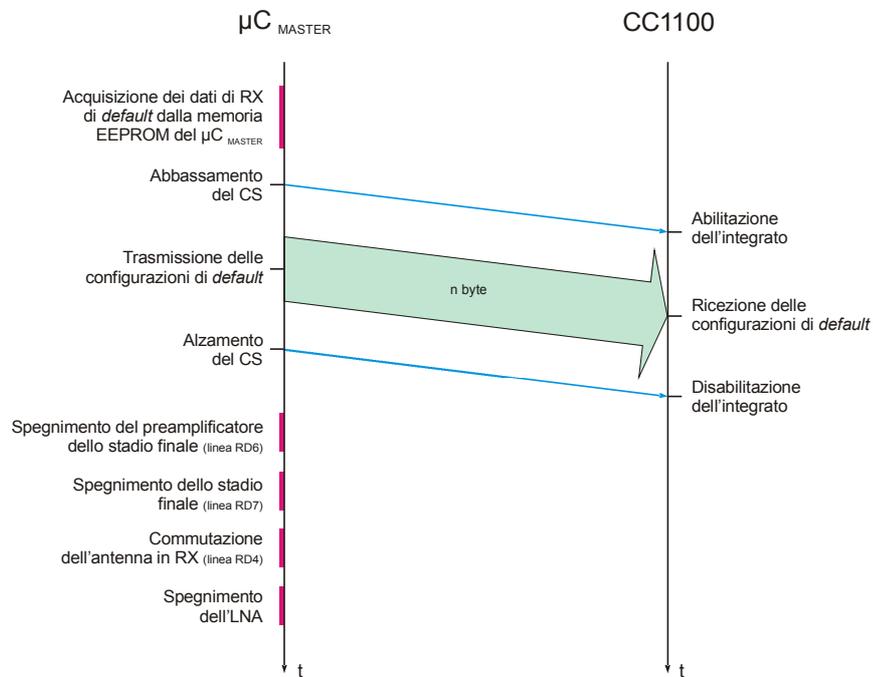


Figura 4.16: Procedura di invio delle configurazioni di default per la modalità RX

Al termine della configurazione in modalità TX/RX, avverrà la fase di trasmissione/ricezione dei dati (descritta nel paragrafo successivo)

4.2.3 Interazione $\mu\text{C MASTER} \rightarrow \text{CC1100}$ (trasmissione e ricezione dei dati)

Come detto in precedenza, al momento attuale non si conoscono le modalità con cui inviare i dati all'integrato CC1100, tanto meno quelle per riceverli; ciò di cui si è sicuri è la necessità di alzare ed abbassare la linea CS in modo da orientare la comunicazione verso l'integrato CC1100. Per garantire la massima flessibilità dell'integrato in questione, sono state connesse anche le linee GDO0 e GDO1 alla porta USART del μC .

Il $\mu\text{C MASTER}$ inoltre, con molta probabilità (è ancora da verificare), dovrà preoccuparsi della formazione del pacchetto di dati secondo lo standard AX.25 raffigurato sotto.

| Flag | Address | Control | Pid | Info | FCS | Flag |
|----------|-------------|---------|-------|-----------|--------|----------|
| 01111110 | 112-560 bit | 8 bit | 8 bit | N · 8 bit | 16 bit | 01111110 |

Il μC dovrà quindi occuparsi, oltre alla formazione dei campi sopra riportati, anche dell'inserimento del *bit di stuffing* (per evitare la presenza della sequenza di Flag all'interno dei dati) e del calcolo dell'FCS (*Frame Check Sequence* a 16 bit).

4.2.4 Interazione $\mu\text{C MASTER} \rightarrow \text{DAC AD5300}$

L'inserimento del DAC proviene dalla necessità di regolare la potenza di trasmissione in modo da operare (laddove sia possibile, affinché avvenga la ricezione a Tera) un risparmio di potenza. L'uscita del DAC andrà infatti a comandare un circuito di retroazione sull'amplificatore di potenza.

Siccome il DAC scelto comunica con il $\mu\text{C MASTER}$ sul bus SPI, avremo quattro linee di collegamento tra i due: SDI, SDO, SCLK e $\overline{\text{SYNC}}$ (quest'ultimo ha lo stesso comportamento del CS). La linea SDO in particolare sarà sempre inutilizzata infatti, l'AD5300 ha il solo compito di convertire ciò che riceve in ingresso (in forma digitale e seriale) all'uscita.

Per come è fatto il DAC AD5300, affinché venga prodotto in uscita il segnale desiderato (una tensione continua), occorre inviare una particolare sequenza composta da 16 bit (vedi Figura 4.11).

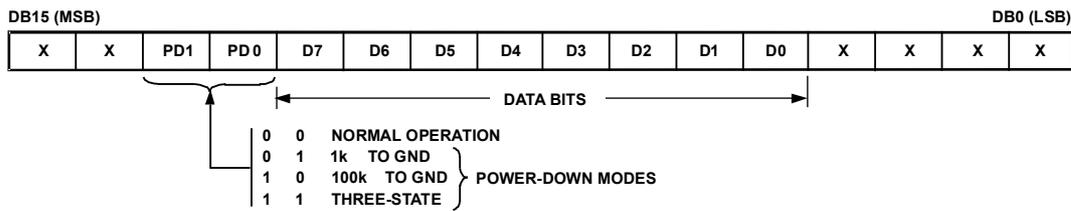


Figura 4.17: Pacchetto DAC AD5300

Come si può notare, la combinazione dei bit PD1 e PD0, mi consente di impostare o la modalità di risparmio energetico (consumo di 200nA) o quella di normale funzionamento. Si stabilisce che la modalità di funzionamento abituale del DAC sia quella di basso consumo mentre, nel momento in cui occorre trasmettere a Terra il *beacon* (durante il *tempo di attesa*) oppure i dati (durante il *tempo di accesso*), viene attivata la modalità di funzionamento normale.

Il significato della linea \overline{SYNC} è il medesimo della linea CS (*chip select*) vista nei casi in precedenza. Nel momento in cui tale linea viene abbassata di livello, l'integrato viene selezionato; viceversa, quando la linea viene portata a livello logico alto, l'integrato viene deselezionato oppure, nel caso fosse stata in atto una trasmissione di dati, si ha l'immediata interruzione (vedi Figura).

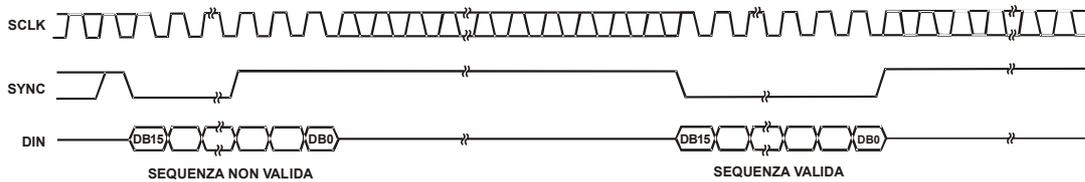


Figura 4.18: Sequenza valida/non valida nella trasmissione al DAC AD5300

4.2.5 Interazione $\mu\text{C MASTER} \rightarrow \mu\text{C SLAVE}$

Nel momento in cui occorre la necessità della comunicazione con il $\mu\text{C SLAVE}$ da parte del $\mu\text{C MASTER}$ attraverso il bus SPI, si ha l'esecuzione delle fasi illustrate dalla Figura 4.10. La comunicazione tra i due può essere scaturita o dall'invio (dalla stazione di Terra) del *telecomando* atto ad alterare la cadenza delle misurazioni strumentali, o da una precedente richiesta di utilizzo del bus da parte del $\mu\text{C SLAVE}$. Nel caso in cui, una volta instaurata la comunicazione tra i due microcontrollori, il $\mu\text{C MASTER}$ debba, per esempio, orientare in un secondo momento il bus per comunicazione con la sezione di TX/RX, basterà che questi generi un impulso (passaggio livello logico alto e subito dopo basso) sulla sua linea RD5.

La procedura di instaurazione della comunicazione è la seguente:

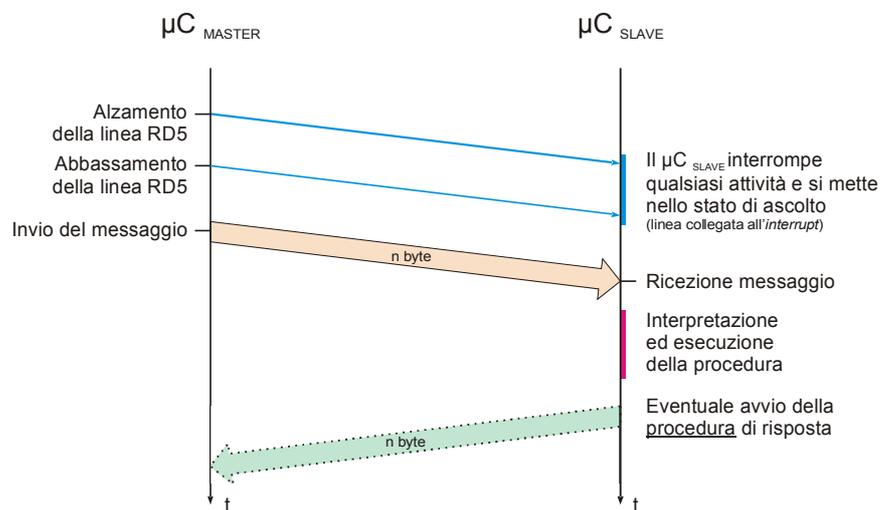


Figura 4.19: Procedura di colloqui tra $\mu\text{C MASTER}$ e $\mu\text{C SLAVE}$

La linea RD5 del $\mu\text{C MASTER}$, come mostrato in Figura 3.25, è collegata al pin di *interrupt* esterno del $\mu\text{C SLAVE}$. Quando tale linea viene portata a livello logico alto (per poi essere riportata a livello logico basso), si avvisa il $\mu\text{C SLAVE}$ dell'imminente invio di dati sul bus (a lui indirizzati). Il $\mu\text{C SLAVE}$ provvede quindi a inibire l'istruzione in esecuzione, a commutare lo switch (ADG736) verso il $\mu\text{C MASTER}$ e a prelevare i dati dal modulo SPI posto al suo interno: una volta ottenuti, avverrà la loro interpretazione ed esecuzione, nonché, a seconda della "storia passata", l'avvio della procedura di risposta. Se infatti prima di tale comunicazione è avvenuta una richiesta di invio di dati da parte del $\mu\text{C SLAVE}$, questo provvederà prima ad ottenerli dal GPS e successivamente ad eseguire la *subroutine* dedicata alla trasmissione sul BUS; se invece i dati costituiscono un *telecomando* inviato dalla stazione di Terra, il $\mu\text{C SLAVE}$ provvederà a eseguirlo e ad alterare la sua configurazione interna.

Al termine del colloquio tra i due μC , lo switch ADG736 viene riportato nella posizione di *default* (orientato verso il magnetometro).

4.3 Interazione del μC SLAVE

Verranno ora descritte le possibili interazioni del μC SLAVE con i sottosistemi a lui connessi, in particolare gli strumenti di bordo, nonché il μC MASTER (parte della comunicazione è già stata analizzata nel §4.2.5).

Per quanto riguarda la comunicazione del μC SLAVE con il GPS e il dosimetro, occorre configurare in modo appropriato la porta USART dell'integrato in esame: occorre prestare particolare attenzione alle velocità a cui i due strumenti possono lavorare, altrimenti si corre il rischio di interpretare in modo errato i dati scambiati.

Tutti i dati ricevuti dagli strumenti di bordo vengono momentaneamente immagazzinati all'interno della memoria RAM del μC SLAVE; successivamente verrà avviata la procedura di richiesta dell'utilizzo del bus (descritta nel paragrafo che segue), in modo da avviare a sua volta la procedura di immagazzinamento nella memoria di massa.

Si osserva, infine, che nel momento in cui viene terminata l'acquisizione dei dati dagli strumenti di bordo e il loro invio al μC MASTER, il μC SLAVE deve porsi nella sua modalità di basso consumo ovvero eseguendo l'istruzione assembler SLEEP.

4.3.1 Interazione μC SLAVE \rightarrow μC MASTER (invio dati)

Nella trattazione che segue, si suppone che il μC SLAVE abbia già acquisito i dati dagli strumenti di bordo: deve quindi preoccuparsi del loro immagazzinamento nella memoria di massa.

Per comunicare attraverso il bus, il μC SLAVE deve obbligatoriamente inviare una richiesta al μC MASTER, il quale potrà acconsentire oppure negare il permesso ad usufruirne.

La richiesta di invio di dati al μC MASTER è dovuta o all'invio dei dati strumentali, o all'invio, in seguito ad una richiesta da parte del μC MASTER, dei dati relativi all'ora e alla data.

La fase di richiesta di utilizzo del bus per la trasmissione avviene secondo la seguente procedura:

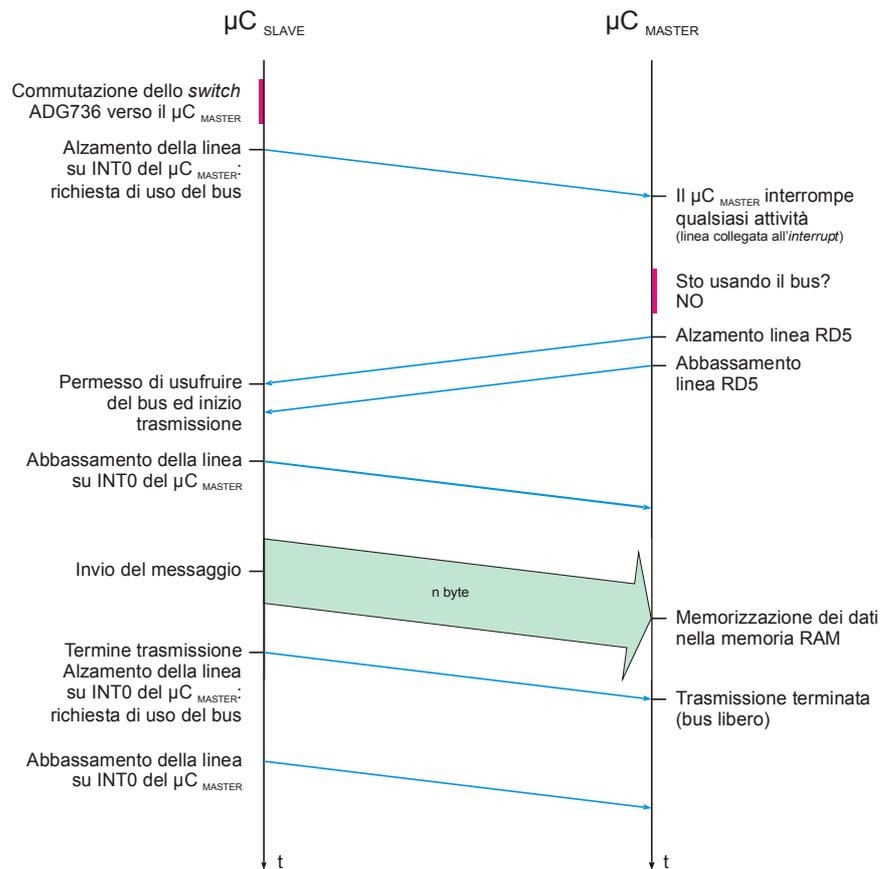


Figura 4.20: Richiesta ed invio di dati sul bus destinati al $\mu\text{C}_{\text{MASTER}}$ ad opera del $\mu\text{C}_{\text{SLAVE}}$

Il $\mu\text{C}_{\text{SLAVE}}$, per comunicare sul bus, dopo aver opportunamente commutato lo *switch* ADG736, porta a livello logico alto la linea collegata al pin INT0 del $\mu\text{C}_{\text{MASTER}}$; quest'ultimo, una volta percepito il cambiamento dello stato della linea (*interrupt*), verifica l'utilizzo da parte sua del bus: in caso negativo porta a livello logico alto la sua linea RD5 , dando quindi il permesso al $\mu\text{C}_{\text{SLAVE}}$ di avviare la trasmissione.

Una volta avuta la conferma, il $\mu\text{C}_{\text{SLAVE}}$ provvede ad avviare la trasmissione dei dati. Nel momento in cui questi ultimi sono terminati, il $\mu\text{C}_{\text{SLAVE}}$ alza ed abbassa la linea collegata al pin INT0 del $\mu\text{C}_{\text{MASTER}}$: tale segnale viene interpretato da quest'ultimo come il termine della trasmissione.

Nel caso in cui, a causa dell'inizio del *tempo di accesso*, il $\mu\text{C}_{\text{MASTER}}$ deve riprendere il controllo sul bus, basta che questi generi un impulso (passaggio livello logico alto e subito dopo basso) sulla sua linea RD5 : così facendo viene scatenato un *interrupt* nel $\mu\text{C}_{\text{SLAVE}}$ il quale bloccherà qualsiasi operazione in atto sul bus e provvederà a commutare lo *switch* ADG736 nella sua posizione naturale. Il controllo del bus da parte del $\mu\text{C}_{\text{SLAVE}}$, viene ripreso solo su comunicazione del $\mu\text{C}_{\text{MASTER}}$ tramite un'opportuna trasmissione (vedi §4.2.5).

Nel caso in cui avvenga una richiesta di utilizzo da parte del μC SLAVE, e il bus è impegnato dal μC MASTER, questi, terminato l'utilizzo, si incaricherà di avvisare il μC SLAVE secondo la procedura descritta in 4.2.5, consentendogli quindi il controllo.

4.3.2 Interazione μC SLAVE \rightarrow GPS

Nella descrizione che segue, si suppone di conoscere le modalità con cui interpellare il GPS ovvero la struttura del pacchetto da inviargli.

Dalla Figura 3.25, si osserva che, affinché avvenga la comunicazione con lo strumento in esame, occorre operare sull'integrato DG9415 ed in particolare sulla linea SELEZIONE STRUMENTO.

Una volta configurata in modo opportuno la porta USART del μC SLAVE, occorre procedere con le seguenti fasi:

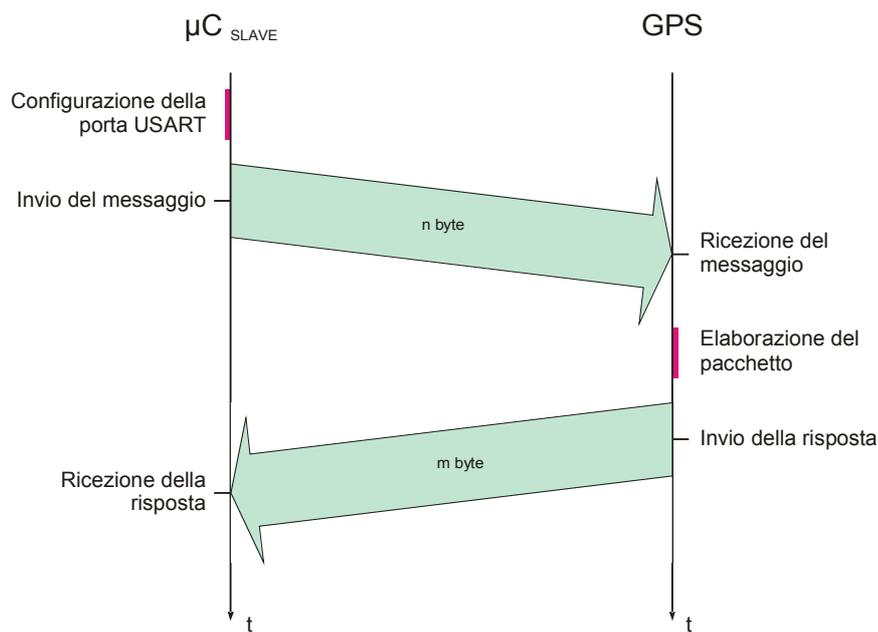


Figura 4.21: Procedura di colloquio tra μC SLAVE e GPS

4.3.3 Interazione $\mu\text{C}_{\text{SLAVE}} \rightarrow$ Dosimetro

La procedura che consente l'interazione tra $\mu\text{C}_{\text{SLAVE}}$ e dosimetro, è simile a quella descritta nella Figura 4.9, ovvero:

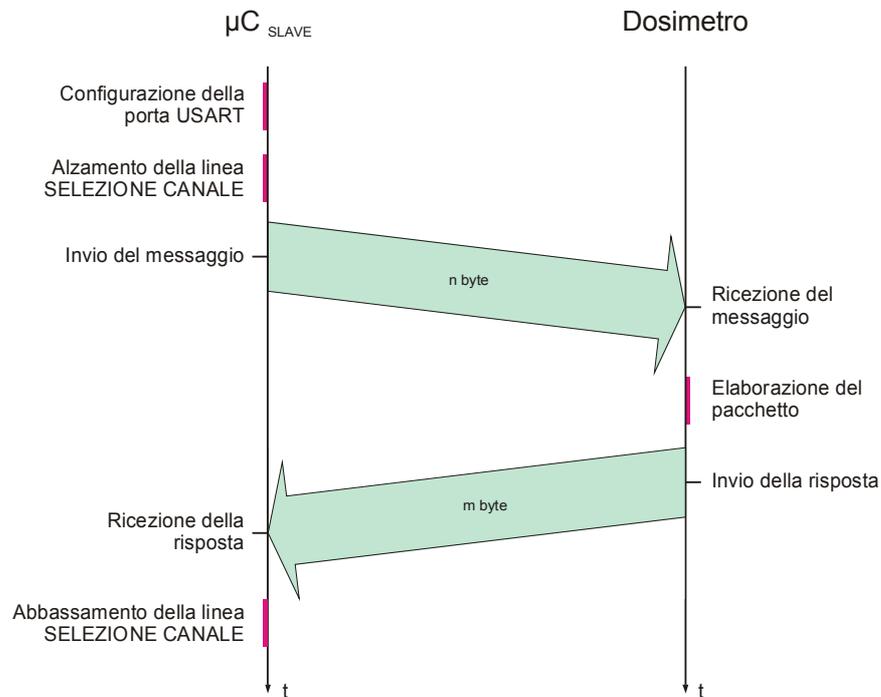


Figura 4.22: Procedura di colloquio tra $\mu\text{C}_{\text{SLAVE}}$ e Dosimetro

L'alzamento della linea SELEZIONE DEL CANALE ha lo scopo di permettere il collegamento tra la linea TX del dosimetro e la linea RX del $\mu\text{C}_{\text{SLAVE}}$, nonché viceversa. Al termine della ricezione dei dati, tale linea viene riposta a livello logico basso, in quanto si stabilisce che il GPS è per *default* sempre collegato al $\mu\text{C}_{\text{SLAVE}}$. Questa scelta è stata adottata per i motivi di richiesta dell'ora e della data dal $\mu\text{C}_{\text{MASTER}}$ per la tracciatura temporale dei dati di *telemetria* (vedi §3.3); infatti, se tale richiesta avviene, il $\mu\text{C}_{\text{SLAVE}}$ potrà subito procedere alla richiesta senza dover prima selezionare lo strumento.

La struttura interna dell'interno del DG9415 è proposta sotto.

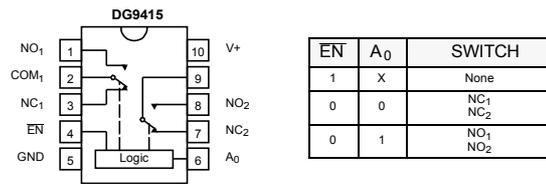


Figura 4.23: Struttura interna DG9415

4.3.4 Interazione μC SLAVE \rightarrow Magnetometro

La struttura interna del ADC ADS8343 è riportata nella figura sotto:

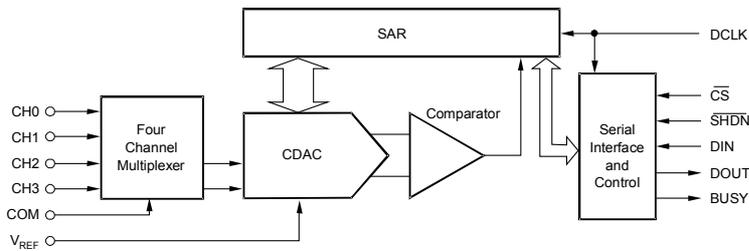


Figura 4.24: Schema interno DAC ADS8343

Tramite il CS, attivo a livello logico basso, si abilita l'integrato alla trasmissione/ricezione.

La configurazione dell'integrato in esame avviene per mezzo di un opportuno byte di controllo inviato all'inizio della comunicazione con il μC SLAVE. Tale byte è così composto:

| Bit 7 (MSB) | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) |
|-------------|-------|-------|-------|-------|---------|-------|-------------|
| S | A2 | A1 | A0 | — | SGL/DIF | PD1 | PD0 |

Figura 4.25: Byte di controllo dell'ADS8343

In particolare il bit S deve essere sempre posto a “1” in modo da avvisare l'inizio della comunicazione. Il bit A2, A1 e A0, consentono di selezionare il canale su cui effettuare la conversione; a tal proposito si può dire che, a seconda del bit SGL/\overline{DIF} , è possibile stabilire se l'acquisizione deve essere effettuata in modo differenziale (vedi Figura 4.8) oppure in modalità *single-ended*. Nel nostro caso verrà adottata quest'ultima modalità.

| A2 | A1 | A0 | CH0 | CH1 | CH2 | CH3 | COM |
|----|----|----|-----|-----|-----|-----|-----|
| 0 | 0 | 1 | +IN | | | | -IN |
| 1 | 0 | 1 | | +IN | | | -IN |
| 0 | 1 | 0 | | | +IN | | -IN |
| 1 | 1 | 0 | | | | +IN | -IN |

Tabella 4.1: Configurazione dei bit A2, A1 e A0

Gli ultimi due bit (PD1-PD0) consentono di impostare la modalità *power down mode*. Se entrambi sono posti a 1 la modalità basso consumo non entra mai in funzione, viceversa, se entrambi i bit sono invece a livello logico basso, tra una conversione e l'altra, la modalità viene attivata. Quando viene avviata una nuova conversione si ha l'uscita dalla modalità in esame. La combinazione 01 non viene implementata mentre la 10 serve ad abilitare la modalità *internal clock*. Quest'ultima si contrappone a quella *external clock* (che si ha con le combinazioni P1-P0 = 00 o 11). Nella prima, una volta inviato il byte di controllo, l'ADS8343 procede alla conversione e alla memorizzazione del risultato in un registro: il suo contenuto può essere prelevato in qualsiasi momento (vedi Figura 4.6).

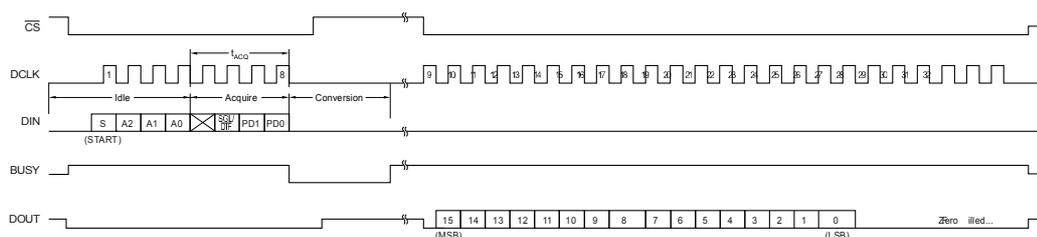


Figura 4.26: Modalità internal clock

Nella seconda invece, terminato l'invio della configurazione, viene portata a livello logico alto la linea BUSY per la durata di un ciclo di *clock*, in modo tale da consentire l'inserimento dell'ultimo bit (il P0); successivamente si ha l'invio, sulla linea SDO dell'integrato in questione, dei dati della conversione ottenuti per approssimazioni successive. Tale modalità può essere adottata in due distinte maniere riportate nelle Figure 4.4 e 4.5.

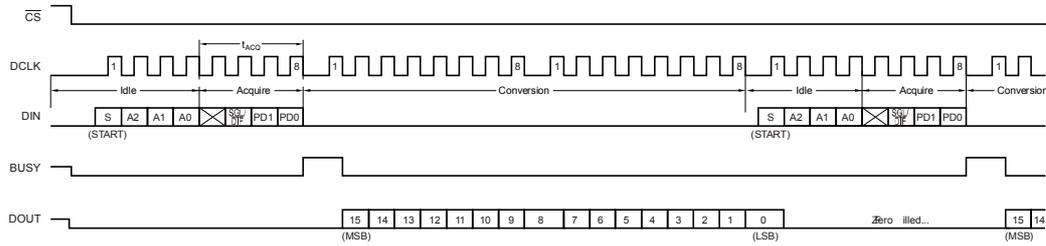


Figura 4.27: Modalità external clock: conversione a 24 clock

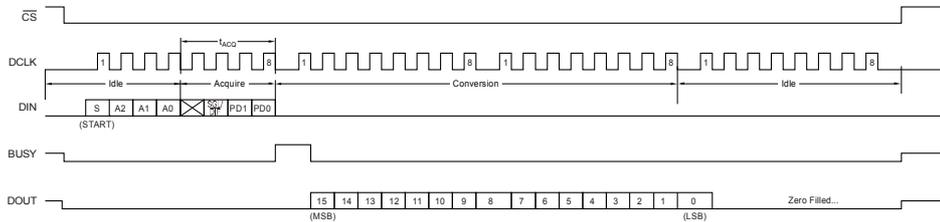


Figura 4.28: Modalità external clock: conversione a 32 clock

Nella Figura 4.4 in particolare, viene rappresentata la conversione a 24 colpi di *clock*. Infatti, all'invio dell'LSB dell'acquisizione, viene inviato un nuovo byte di controllo: tale byte potrà contenere la richiesta di acquisizione di un altro canale. Nella comunicazione tra μC SLAVE e ADS8343 verrà utilizzata tale modalità.

La configurazione iniziale del byte di controllo dell'ADS8343 è: 10010100. Ad ogni termine di prelievo dei dati, saranno modificati i bit A2, A1 e A0 (bit sottolineati) secondo la Tabella 4.1, in modo da procedere all'acquisizione di tre canali (X, Y e Z). La combinazione 110 non verrà mai utilizzata.

Nella raffigurazione che segue, si supporrà che non avvenga nessuna richiesta di invio di dati da parte del μC MASTER. Nel caso in cui questa avvenga, il μC SLAVE provvederà ad interrompere la procedura in corso di svolgimento, ad orientare lo *switch* verso il μC MASTER e a “mettersi in ascolto” del messaggio. Una volta terminata quindi la sequenza di istruzioni esposta in 4.2.5, il μC SLAVE completerà la procedura rimasta in sospeso.

L'evoluzione nel tempo della procedura di acquisizione dei segnali provenienti dal magnetometro è rappresentata nel seguente diagramma:

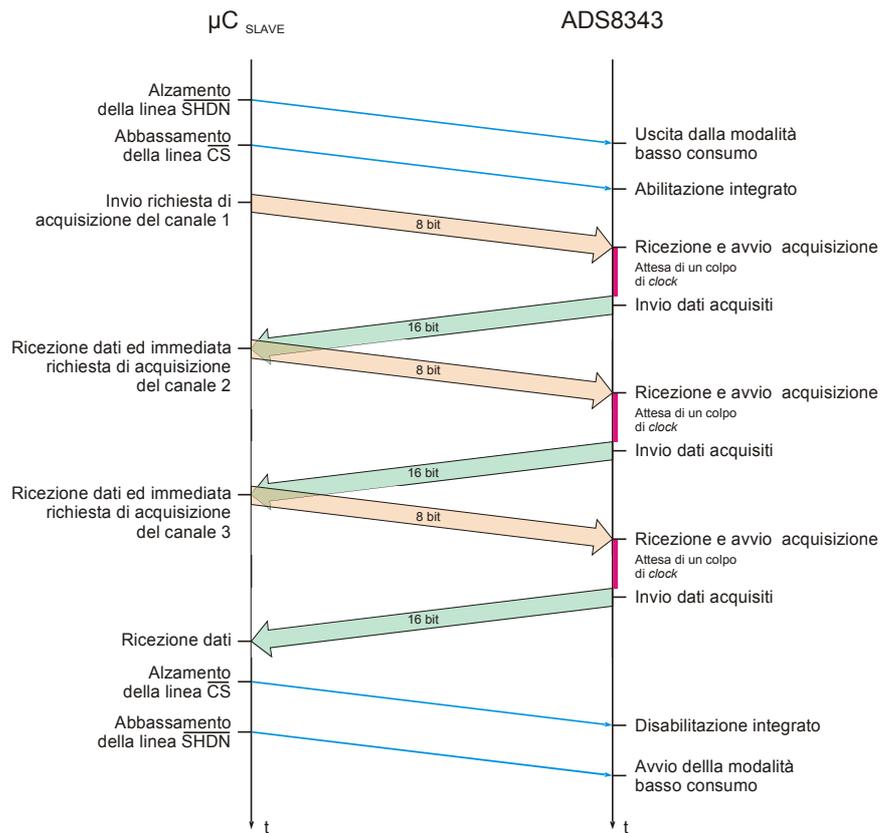


Figura 4.29: Procedura di acquisizione dei segnali dal convertitore AD ADS8343

Capitolo 5

Stima ed organizzazione della memoria di Telemetria

Come è stato detto nel Capitolo 1, possiamo avere due tipi di *telemetria*: quella base (TMB) e quella estesa (TME). Ognuna di queste tipologie ha inoltre una variante: la *telemetria addizionata*. Per ciascuna di queste classificazioni avremo un'occupazione di memoria differente. Per quanto riguarda lo spazio necessario per memorizzare i dati di *telemetria addizionata*, possiamo dire fin da ora che essi avranno la stessa occupazione di memoria, sia che si tratti della TMB Addizionata sia della TME Addizionata. Ciò che differenzia queste varianti infatti, è solo la collocazione dei dati al termine di una o dell'altra.

Dopo un'introduzione sui metodi di acquisizione e memorizzazione delle grandezze di *telemetria*, ci si concentrerà sul calcolo della memoria necessaria all'immagazzinamento dei dati prodotti da ciascun tipo di *telemetria*. Successivamente si procederà a stimare la memoria necessaria per ciascuna classificazione di *telemetria*.

Al termine del capitolo inoltre, verrà fatta una stima dei dati prodotti dagli strumenti di bordo.

Poiché per l'interconnessione dei blocchi funzionali interni al satellite è stato scelto un microcontrollore a otto bit, ciascuna locazione di memoria varrà per l'appunto otto bit e sarà rappresentata dal termine byte. Quest'ultimo avrà come bit più significativo (MSB) il bit 7 mentre come bit meno significativo (LSB) il bit 0.

5.1 Metodi di acquisizione e memorizzazione dei dati di *telemetria*

Siccome il μC (incaricato ad acquisire i dati di *telemetria*) è un'unità che lavora a otto bit (1 byte), il convertitore analogico-digitale presente al suo interno produrrà, come risultato finale, due byte, uno dei quali conterrà sicuramente sei zeri.

Il formato dei dati in uscita all'ADC inoltre, può essere di due tipi: giustificato a destra o giustificato a sinistra (come mostra la Figura 5.1). Per convenzione stabiliamo una giustificazione destra.

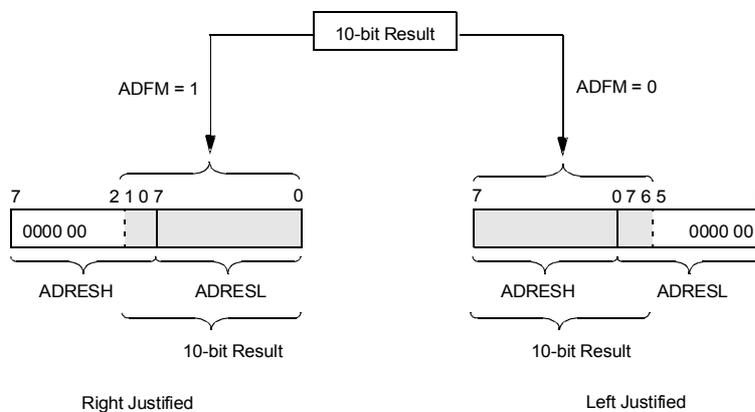


Figura 5.1: Giustificazione dei dati a destra e a sinistra

Una volta effettuata l'acquisizione della grandezze di *telemetria* e ottenuta la coppia di byte, il μC provvederà ad adottare due distinte tecniche di memorizzazione dei dati:

- **tecnica di memorizzazione "standard"**
- **tecnica di memorizzazione differenziale**

5.1.1 Tecnica di memorizzazione "standard"

Tale tecnica non opera nessun cambiamento al contenuto della coppia di byte in uscita dall'ADC. Il μC provvede solamente ad avviare la procedura di memorizzazione, collocando nella memoria EEPROM di massa: prima il byte contenente i bit più significativi e successivamente quelli meno significativi. Se quindi abbiamo N acquisizioni consecutive, avremo un'occupazione di memoria di $2 \cdot N$ byte.

Si capisce subito che, memorizzare un numero elevato di campioni composto da due byte con al loro interno bit perennemente nulli, risulta essere uno spreco di memoria. Si adotta quindi la tecnica differenziale.

5.1.2 Tecnica di memorizzazione differenziale

Con questa tecnica si può già dire che è possibile ottenere un'occupazione di memoria minore rispetto all'adozione della precedente.

In questa tecnica il primo dato, composto dai due byte in uscita dall'ADC, viene memorizzato nella memoria come descritto nella tecnica "standard". Tale dato inoltre viene considerato come **riferimento**.

Nelle acquisizioni successive invece, non viene memorizzata la coppia di byte prodotta dal convertitore, bensì facendo uso di un byte, il suo discostamento dal dato preso come riferimento. L'utilizzo di tale locazione di memoria avviene però secondo una logica precisa. L'MSB del byte in esame, sta ad indicare se il discostamento è positivo o negativo (per convenzione 1 = è avvenuto un incremento rispetto al dato di riferimento, 0 = è avvenuto un decremento rispetto al dato di riferimento), mentre i rimanenti sette bit mi esprimono la differenza tra i due campioni. Se occorre per esempio dover trattare le seguenti sequenze, in cui il primo dato è quello di riferimento,

| | | |
|----------|--------------------------|----------|
| 1° dato: | <u>000000</u> 1100101011 | (10 bit) |
| 2° dato: | <u>000000</u> 1100110101 | (10 bit) |
| 3° dato: | <u>000000</u> 1100100100 | (10 bit) |

il discostamento del secondo dato è pari alla combinazione di bit 0001010. Quindi, siccome il dato è maggiore del riferimento, verrà memorizzato il byte 10001010. Per quanto riguarda invece il discostamento del terzo dato, abbiamo la combinazione di bit 0000111 e, poiché il dato è minore del riferimento, dovremo memorizzare il byte 00000111.

Per il momento si è supposto che l'oscillazione del segnale delle grandezze in esame sia contenuta, ovverosia la tensione all'ingresso del convertitore AD non oscilli più di $\pm 0,3V$ (pari a circa 6,5 bit). Nel caso in cui questo accada, ovverosia il discostamento della grandezza in esame è maggiore della sequenza 1111110 (7 bit), vengono inseriti nella memoria due byte consecutivi composti da soli "1", seguiti da un nuovo dato di riferimento (quindi 2 byte). La scelta di inserire questi due particolari byte è stata effettuata in modo da poter permettere al ricevitore di identificare il nuovo dato, a cui si riferiranno i campioni successivi. Adottando quindi tale tecnica di memorizzazione, si corre il rischio – nel caso in cui la tensione all'ingresso dell'ADC vari continuamente oltre i limiti di $\pm 0,3V$ – di avere un'occupazione di memoria maggiore rispetto alla

tecnica di memorizzazione “standard”, in quanto si dovrebbero memorizzare 4 byte anziché 2. Generalmente nella pratica tali oscillazioni non avvengono con frequenza; nel caso in cui queste si verificassero, per evitare la saturazione della memoria, viene posto un limite alla quantità di dati memorizzabili nell'orbita in corso. Tale limite viene stabilito come l'occupazione di memoria che si avrebbe se si adottasse la tecnica di memorizzazione “standard”. L'inconveniente in questo caso sarebbe la mancata conoscenza dei dati di alcune misure lungo una singola orbita.

Ad ogni inizio orbita inoltre, viene effettuata una nuova acquisizione del dato di riferimento.

Con tale tecnica quindi, se si hanno N acquisizioni consecutive che non eccedono i limiti imposti, avremo un'occupazione di memoria pari a $2+(N-1)$ byte: un netto risparmio in confronto alla precedente tecnica.

5.2 Calcolo della memoria occupata per la singola orbita

Viene ora illustrata la metodologia per il calcolo dell'occupazione di memoria per ciascuna grandezza di *telemetria* nell'intervallo di tempo di un'orbita, secondo quindi la Tabella 1.3. Nel caso in cui si desideri diminuire l'intervallo di acquisizione, ovvero si lavori con le tempistiche della Tabella 1.5, oppure ridurre le grandezze di cui tener traccia (vedi §1.4), si avrà un volume di dati inferiore a quelli valutati in questo paragrafo.

5.2.1 Tensioni e correnti

Queste due grandezze vengono raggruppate sotto un'unica metodologia di calcolo di memoria, in quanto, ogni volta che devono essere misurate le tensioni, vengono determinate immediatamente dopo anche le correnti; inoltre entrambe le grandezze vengono prelevate con la medesima cadenza (vedi Tabella 1.3).

Le tensioni e le correnti di cui vogliamo conoscere il valore sono le seguenti:

- **TENSIONI DI ALIMENTAZIONE**
 - tensione dei pannelli solari (+ 7V)
 - tensione della batteria (+ 10,8V)
 - tensione + 5V
 - tensione + 3,3V
- **il SENSO DELLA CORRENTE DELLA BATTERIA**
- **CORRENTI DI ALIMENTAZIONE**
 - corrente dei pannelli solari
 - corrente della batteria

Abbiamo quindi quattro tensioni e due correnti da dover acquisire tramite il convertitore AD del $\mu\text{C}_{\text{MASTER}}$. Quest'ultimo dovrà inoltre individuare il senso della corrente della batteria, operazione che si realizza con un semplice test della porta RA2.

Dallo schema riportato in Figura 3.15, si osserva che le precedenti grandezze di *telemetria*, ad eccezione della tensione dei pannelli solari, vengono multiplexate in un'unica linea analogica. Nel momento in cui occorrerà acquisire i dati, si dovrà selezionare l'opportuna grandezza di *telemetria* secondo l'ordine riportato sopra.

Facendo riferimento alla Tabella 1.3, e adottando entrambe le tecniche di memorizzazione riportate precedentemente, abbiamo i seguenti risultati:

| Grandezza | n° misure per orbita | n° byte prima acquisizione | n° byte acquisizioni successive | n° byte per orbita |
|-------------------------------------|----------------------|----------------------------|---------------------------------|--------------------|
| Tensione dei pannelli solari | 126 | 2 | 2 | 252 |
| Tensione della batteria | 126 | 2 | 2 | 252 |
| Tensione + 5V | 126 | 2 | 2 | 252 |
| Tensione + 3,3V | 126 | 2 | 2 | 252 |
| Senso della corrente della batteria | 126 | 1 | 1 | 126 |
| Corrente dei pannelli solari | 126 | 2 | 2 | 252 |
| Corrente della batteria | 126 | 2 | 2 | 252 |

Totale 1.638

Tabella 5.1: Calcolo dell'occupazione di memoria delle tensioni e delle correnti, utilizzando la tecnica di memorizzazione "standard"

| Grandezza | n° misure per orbita | n° byte prima acquisizione | n° byte acquisizioni successive | n° byte per orbita |
|-------------------------------------|----------------------|----------------------------|---------------------------------|--------------------|
| Tensione dei pannelli solari | 126 | 4 | 1 | 129 |
| Tensione della batteria | 126 | 4 | 1 | 129 |
| Tensione + 5V | 126 | 4 | 1 | 129 |
| Tensione + 3,3V | 126 | 4 | 1 | 129 |
| Senso della corrente della batteria | 126 | 1 | 1 | 126 |
| Corrente dei pannelli solari | 126 | 4 | 1 | 129 |
| Corrente della batteria | 126 | 4 | 1 | 129 |

Totale 900

Tabella 5.2: Calcolo dell'occupazione di memoria delle tensioni e delle correnti, utilizzando la tecnica di memorizzazione differenziale

Quindi si ha un'occupazione minima di 900 byte e massima di 1.638 byte (la massima si ha secondo il ragionamento fatto in 5.2.2).

Si osserva che nella pratica la tensione dei pannelli solari viene acquisita con una cadenza maggiore rispetto a quella indicata nella Tabella 1.3, in quanto tale tensione viene utilizzata per il controllo del MPP. La cadenza di Tabella 1.3 si riferisce alle memorizzazioni delle grandezze di *telemetria*.

Per quanto riguarda il byte che mi rappresenta il senso della corrente della batteria, la sua occupazione di memoria è sottoutilizzata: nel senso che al suo interno, ad ogni acquisizione e memorizzazione, viene utilizzato solamente un bit. Un possibile rimedio a questo inconveniente (non adottato in questa trattazione) è quello di introdurre una periodicità su tale byte, in modo tale che ad ogni acquisizione venga settato un bit del byte in questione, e dopo otto acquisizioni ne venga inserito uno nuovo. Così facendo un generico byte del senso della corrente della batteria, conterrà al suo interno i sensi della corrente delle successive otto acquisizioni.

5.2.2 Temperature

Dalla Tabella 1.3, si apprende che la cadenza con cui vengono acquisite le temperature è minore di quella delle tensioni e correnti; si avrà quindi un'occupazione di memoria inferiore.

Le temperature di nostro interesse sono:

- temperatura interna
- temperatura esterna
- temperatura del componente di maggior potenza

Come descritto nel §3.3, le temperature sono multiplexate, assieme alle correnti e ad alcune tensioni, sulla linea analogica AN1; occorrerà pertanto agire sulle linee digitali RE0, RE1 e RE2 (vedi Tabella 3.1.2) in modo da selezionare i canali dell'ADG708 di nostro interesse.

Come nel §5.2.1, poniamo a confronto la memorizzazione delle grandezze adottando le due differenti tecniche. Abbiamo quindi le seguenti tabelle:

| Grandezza | n° misure per orbita | n° byte prima acquisizione | n° byte acquisizioni successive | n° byte per orbita |
|---|----------------------|----------------------------|---------------------------------|--------------------|
| Temperatura interna | 63 | 2 | 2 | 126 |
| Temperatura esterna | 63 | 2 | 2 | 126 |
| Temperatura del componente di maggior potenza | 63 | 2 | 2 | 126 |
| Totale | | | | 378 |

Tabella 5.3: Calcolo dell'occupazione di memoria della temperatura, utilizzando la tecnica di memorizzazione "standard"

| Grandezza | n° misure per orbita | n° byte prima acquisizione | n° byte acquisizioni successive | n° byte per orbita |
|---|----------------------|----------------------------|---------------------------------|--------------------|
| Temperatura interna | 63 | 4 | 1 | 66 |
| Temperatura esterna | 63 | 4 | 1 | 66 |
| Temperatura del componente di maggior potenza | 63 | 4 | 1 | 66 |

Totale 198

Tabella 5.4: Calcolo dell'occupazione di memoria della temperatura utilizzando, la tecnica di memorizzazione differenziale

Quindi si ha un'occupazione minima di 198 byte e massima di 378 byte (la massima si ha secondo il ragionamento fatto in 5.1.2).

5.2.3 Memoria residua

L'occupazione di memoria di tale dato, ammonta a 3 byte. Questa dimensione mi consente di conteggiare fino a $2^{24}=16.777.216$ byte. Due byte infatti, non sono sufficienti in quanto è possibile conteggiare solo 65.536 locazioni di memoria; tale valore è difatti una quantità inferiore a quella con cui si avrà a che fare.

Nel momento in cui si avrà la richiesta di memorizzare il valore della memoria residua, il $\mu\text{C}_{\text{MASTER}}$ preleverà il contenuto della variabile nominata precedentemente e lo collocherà nella memoria di massa, non prima però di averlo decrementato di 3 byte (occupazione di memoria di tale dato in esame).

5.2.4 Memoria totale dei dati strumentali

Ogni volta che devono essere memorizzati i dati provenienti dagli strumenti di bordo, il $\mu\text{C}_{\text{MASTER}}$ provvede ad incrementare un'opportuna variabile posta al suo interno. Nel momento in cui è necessario memorizzare il dato di *telemetria* in esame, il $\mu\text{C}_{\text{MASTER}}$ preleva il contenuto di tale variabile lo colloca nella memoria di massa. Per gli stessi motivi presentati prima, l'occupazione di memoria della variabile (quindi del dato di *telemetria*) ammonta a 3 byte.

5.2.5 Potenza media nelle trasmissioni e ricezioni

La potenza media utilizzata per l'invio e la ricezione dei pacchetti, viene calcolata una volta terminato il *tempo di accesso*. Per ogni pacchetto infatti, viene determinato o misurato il livello di potenza utilizzato e sommato con il precedente; una volta terminato il *tempo di accesso*, viene operata una media. Al termine del §3.5, è stato proposto un metodo di calcolo della potenza utilizzata in trasmissione. Per quanto riguarda la potenza in ricezione invece, tale compito potrebbe essere assolto dall'integrato CC1100 (è ancora da vedere se è dotato di questa caratteristica).

Affinché avvenga il calcolo delle potenze in esame, all'interno del $\mu\text{C MASTER}$ verranno riservate delle opportune locazioni di memoria RAM in modo da poter effettuare la media dei dati. Un insieme di queste locazioni sarà dedicato al conteggio dei pacchetti inviati, mentre un altro gruppo sarà dedicato al il conteggio di quelli ricevuti.

Lo spazio di memoria per immagazzinare i dati finali è pari a 4 byte: 2 per la potenza utilizzata in trasmissione e 2 per quella impiegata in ricezione.

5.2.6 Numero di accessi non autorizzati

Tale dato di *telemetria*, come già accennato nei capitoli precedenti, consiste in un contatore il cui contenuto rappresenta gli accessi effettuati al satellite ad opera di utenti a noi sconosciuti. Si suppone che durante il *tempo di attesa* non avvengano molti tentativi non autorizzati, quindi si ritiene che sia sufficiente 1 byte. Con tale dimensione è possibile conteggiare fino a 255 accessi. Supponendo che ne avvengano in numero maggiore, tale variabile, una volta incrementata, ricomincerebbe il conteggio partendo da zero: tramite il *firmware* del μC ciò viene impedito.

5.3 Calcolo dell'occupazione di memoria della *telemetria*

Viene ora effettuata la stima dell'occupazione di memoria per ciascun tipo di *telemetria*. Verrà inoltre considerato il caso limite, ovverosia il caso in cui sono presenti tutte le grandezze di *telemetria*. Al termine di questi calcoli si potrà dunque dichiarare le occupazioni di memoria minima e massima dei dati nella memoria di massa. Tale occupazione nella memoria di massa non riguarda la TMB, in quanto l'occupazione di quest'ultima risiede nella memoria RAM del $\mu\text{C MASTER}$: in ogni caso è saggio considerarla ai fini della stesura del *firmware*.

I calcoli che seguiranno saranno effettuati considerando la Tabella 1.3.

Per quanto concerne l'occupazione di memoria da parte delle telemetrie addizionate, verrà eseguito un unico calcolo, in quanto tali tipologie di *telemetria* si differenziano solamente per la loro collocazione, al termine cioè della TMB o della TME. La loro occupazione totale potrà essere calcolata con le seguenti espressioni:

$$\text{Occupazione di memoria della telemetria base addizionata} = \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{del primo pacchetto} \\ \text{della telemetria base} \end{array} \right) + \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{della telemetria addizionata} \end{array} \right)$$

Formula 5.1: Calcolo dell'occupazione di memoria della telemetria base addizionata

$$\text{Occupazione di memoria della telemetria estesa addizionata} = \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{della telemetria estesa} \end{array} \right) + \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{della telemetria addizionata} \end{array} \right)$$

Formula 5.2: Calcolo dell'occupazione di memoria della telemetria estesa addizionata

Nel caso si adotti una cadenza inferiore (vedi Tabella 1.5), si avrà necessariamente un'occupazione di memoria inferiore.

5.3.1 Calcolo dell'occupazione di memoria della *telemetria base*

Siccome la *telemetria base* viene calcolata e inviata al primo accesso con la stazione di Terra e ad intervalli regolari successivi, non ha senso memorizzarla nella memoria di massa per prelevarla immediatamente dopo. I dati di *telemetria base* vengono quindi momentaneamente immagazzinati all'interno della memoria RAM del $\mu\text{C MASTER}$. Nella *telemetria base* viene utilizzata la tecnica di memorizzazione (nella memoria RAM) "standard" e quindi, ad ogni invio dei dati, la memoria RAM può essere liberata. Ad ogni invio della *telemetria base*, ad eccezione del primo, si avrà quindi lo stesso numero di byte. Si ricorda infatti, che nella *telemetria base*, al secondo invio dei dati, sono presenti solamente tensioni, correnti e temperature.

Abbiamo quindi due tipi di trasmissione:

| | Grandezza | Occupazione di memoria [byte] |
|-------------|---|----------------------------------|
| | Memoria totale dei dati strumentali | 3 |
| Tensioni | Tensione dei pannelli solari | 2 |
| | Tensione della batteria | 2 |
| | Tensione dei +5V | 2 |
| | Tensione dei +3,3V | 2 |
| Correnti | Senso della corrente della batteria | 1 |
| | Corrente dei pannelli solari | 2 |
| | Corrente della batteria | 2 |
| Temperature | Temperatura interna | 2 |
| | Temperatura esterna | 2 |
| | Temperatura del componente di maggior potenza | 2 |
| | Memoria residua | 3 |
| Totale | | 25 |

Tabella 5.5: Occupazione di memoria del primo pacchetto di TMB

| | Grandezza | Occupazione di memoria [byte] |
|-------------|---|-------------------------------|
| Tensioni | Tensione dei pannelli solari | 2 |
| | Tensione della batteria | 2 |
| | Tensione dei + 5V | 2 |
| | Tensione dei + 3,3V | 2 |
| Correnti | Senso della corrente della batteria | 1 |
| | Corrente dei pannelli solari | 2 |
| | Corrente della batteria | 2 |
| Temperature | Temperatura interna | 2 |
| | Temperatura esterna | 2 |
| | Temperatura del componente di maggior potenza | 2 |
| Totale | | 19 |

Tabella 5.6: Occupazione di memoria dei pacchetti di TMB successivi al primo

Considerando i campi di intestazione iniziali, descritti nel §1.3, i quali occupano 2 byte e vengono posti all'inizio di ogni pacchetto di TMB, abbiamo che l'occupazione del primo invio e degli invii successivi sono rispettivamente **27** e **21** byte.

Per quanto riguarda l'organizzazione del pacchetto si rimanda al §5.4.1.

5.3.2 Calcolo dell'occupazione di memoria della telemetria estesa

Nel calcolo della TME verrà dapprima determinato il volume di dati generato durante la singola orbita e successivamente quello prodotto da sette consecutive. Si esegue quindi un semplice prodotto tra la quantità di memoria necessaria per la singola orbita e il numero di orbite totali (si aggiungono inoltre 2 byte pari alle occupazioni di memoria dei campi TME_CONFIG e TIPO_TME); ciò è possibile in quanto per ciascuna orbita viene memorizzata la medesima quantità di dati.

Si ricorda che nella TME si adotta la tecnica di memorizzazione differenziale.

5.3.2.1 Calcolo dell'occupazione di memoria per la singola orbita

Dalle Tabelle 5.2 e 5.4, e dai §5.2.3 e §5.2.4, è possibile riassumere le seguenti occupazioni di memoria riferite ad una singola orbita:

- tensioni e correnti: **900** byte
- temperature: **198** byte
- memoria residua: **3** byte
- memoria dei dati strumentali: **3** byte

Considerando i volumi di memoria appena elencati e il fatto che per ogni orbita è necessario memorizzare i dati relativi all'ora e alla data forniti dal GPS (6 byte), l'occupazione di memoria per la singola orbita è desumibile dall'applicazione della seguente espressione:

$$\begin{aligned}
 \text{Occupazione di memoria} &= \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{della singola orbita} \end{array} \right) = \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{tensioni e correnti per orbita} \end{array} \right) + \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{temperature per orbita} \end{array} \right) + \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{Memoria residua} \end{array} \right) \\
 &+ \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{Memoria totale strumenti} \end{array} \right) + \left(\begin{array}{c} \text{Memoria dei} \\ \text{dati del GPS} \end{array} \right) \\
 &= 900 + 198 + 3 + 3 + 6 \\
 &= 1.110 \text{ byte}
 \end{aligned}$$

Formula 5.3: Occupazione di memoria della TMB per la singola orbita

Pertanto, nella singola orbita, si ha un'occupazione di memoria minima pari a **1.110** byte.

$$\begin{aligned}
 \text{Occupazione di} \\
 \text{memoria della TME} &= \left(\text{Occupazione di memoria} \right. \\
 &\quad \left. \text{della singola orbita} \times n^{\circ} \text{ di orbite} \right) + \text{byte di} \\
 &\quad \text{intestazione} \\
 &= (1.110 \times 7) + 2 \\
 &= 7.772 \text{ byte}
 \end{aligned}$$

Formula 5.5: Occupazione di memoria della TME

Pertanto, ogni sette orbite, si avrà un'occupazione di memoria pari a 7.772 byte.

Siccome vengono memorizzate due TME consecutive, si dovrà disporre, nella memoria di massa, di almeno 15.544 byte.

Come nel caso del calcolo della memoria della singola orbita, viene ora calcolata anche la massima occupazione di memoria. Si ha quindi che:

$$\begin{aligned}
 \text{Massima occupazione di} \\
 \text{memoria della TME} &= \left(\text{Massima occupazione di} \right. \\
 &\quad \left. \text{memoria della singola orbita} \times n^{\circ} \text{ di orbite} \right) + \left(\text{byte di} \right. \\
 &\quad \left. \text{intestazione} \right) \\
 &= (2.028 \times 7) + 2 \\
 &= 14.198 \text{ byte}
 \end{aligned}$$

Formula 5.6: Occupazione massima di memoria della TME

Pertanto, dopo sette orbite, avremo un'occupazione di memoria massima pari a 14.198 byte.

Siccome vengono memorizzate due TME consecutive, si dovrà disporre, nella memoria di massa, di almeno 28.396 byte.

5.3.4 Calcolo dell'occupazione di memoria delle telemetrie aggiuntive

Per il calcolo della *telemetria* aggiuntiva occorre considerare l'occupazione di memoria da parte della potenza media utilizzata per la trasmissione e ricezione, nonché il numero di accessi non autorizzati. Nei §5.2.3, §5.2.4 e §4.1.4, è stato detto che l'occupazione di tali grandezze ammonta a:

| Grandezze | n° di byte totali |
|--|-------------------|
| Potenza media utilizzata in trasmissione | 2 |
| Potenza media utilizzata in ricezione | 2 |
| Numero di accessi non autorizzati | 1 |
| Totale | 5 |

Tabella 5.7: Occupazione di memoria delle telemetrie addizionali

Quindi, l'insieme dei dati di ciascuna *telemetria* addizionale occupa uno spazio di memoria pari a **5** byte.

Si ricorda che del caso della *telemetria* base addizionata, tali dati addizionali vengono trasmessi solamente all'interno del primo pacchetto.

Per il calcolo della memoria totale occupata dalle due telemetrie, vale a dire la TMB Addizionata e la TME Addizionata, si applicano rispettivamente le Formule 5.1 e 5.2.

$$\begin{aligned}
 \text{Occupazione di memoria della TMB addizionata} &= \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{del primo pacchetto} \\ \text{della TMB} \end{array} \right) + \left(\begin{array}{c} \text{Occupazione di memoria della} \\ \text{telemetria addizionata} \end{array} \right) \\
 &= 27 + 5 \\
 &= 32 \text{ byte}
 \end{aligned}$$

Formula 5.7: Occupazione di memoria della TMB addizionata

$$\begin{aligned}
 \text{Occupazione di memoria della TME addizionata} &= \left(\begin{array}{c} \text{Occupazione di memoria} \\ \text{della TME} \end{array} \right) + \left(\begin{array}{c} \text{Occupazione di memoria della} \\ \text{telemetria addizionata} \end{array} \right) \\
 &= 7.772 + 5 \\
 &= 7.777 \text{ byte}
 \end{aligned}$$

Formula 5.8: Occupazione di memoria della TME addizionata

Siccome vengono memorizzate due TME consecutive, si dovrà disporre, nella memoria di massa, di almeno **15.554** byte.

$$\begin{aligned}
 \text{Massima occupazione di} \\
 \text{memoria della TME addizionata} &= \left(\begin{array}{c} \text{Massima occupazione di} \\ \text{memoria della TME} \end{array} \right) + \left(\begin{array}{c} \text{Occupazione di memoria della} \\ \text{telemetria addizionata} \end{array} \right) \\
 &= 14.198 + 5 \\
 &= 14.203 \text{ byte}
 \end{aligned}$$

Formula 5.9: Massima occupazione di memoria della TME addizionata

Siccome vengono memorizzate due TME consecutive, si dovrà disporre, nella memoria di massa, di almeno **28.406** byte.

5.3.5 Calcolo dell'occupazione di memoria della TME a cadenza ridotta

Considerando la Tabella 1.5 e seguendo i ragionamenti fatti fino ad ora, è possibile determinare l'occupazione di memoria nel caso in cui venga adottata la cadenza ridotta.

Si ricorda che la scelta di lavorare con una cadenza ridotta, influenza esclusivamente la TME. Inoltre, per quanto riguarda i dati di *telemetria* addizionata, non si ha alcuna variazione rispetto a quelli calcolati nel §5.3.4, in quanto vi è assoluta estraneità tra le misure di potenza (e gli accessi non autorizzati) e la procedura di acquisizione da parte dell'ADC.

Si giunge quindi ai seguenti risultati:

- **Memoria orbitale**
 - ◆ occupazione di memoria della singola orbita: **573 byte**
 - ◆ massima occupazione di memoria della singola orbita: **1.017 byte**
- **Memoria della TME**
 - ◆ occupazione di memoria della TME: **4.013 byte**
 - ◆ occupazione di memoria di due TME consecutive: **8.026 byte**
 - ◆ massima occupazione di memoria della TME: **7.121 byte**
 - ◆ massima occupazione di memoria di due TME consecutive: **14.242 byte**
- **Memoria addizionale**
 - ◆ occupazione di memoria della TME Addizionata: **4.018 byte**
 - ◆ occupazione di memoria di due TME Addizionate consecutive: **8.036 byte**
 - ◆ massima occupazione di memoria della TME Addizionata: **7.126 byte**
 - ◆ massima occupazione di memoria di due TME Addizionate consecutive: **14.252 byte**

5.4 Organizzazione della memoria

Si procede ora alla rappresentazione in forma grafica dell'organizzazione dei dati di *telemetria* all'interno della memoria di massa, nel caso della TME, nonché in quella RAM (nel caso della TMB). Verranno considerati sia i casi di TMB e TME “standard”, sia quelli addizionali.

Nella rappresentazione che segue si suppone che sia impostato l'invio a Terra di tutte le grandezze di *telemetria* che sono state analizzate.

5.4.1 Organizzazione della memoria di TMB

Come già discusso nel §5.3.1, nella TMB abbiamo una lieve differenza tra il primo pacchetto inviato a Terra (dopo l'inizio del *tempo di accesso*) e quelli inviati negli istanti successivi - l'intervallo di tempo e l'invio sono impostabili tramite il registro TMB_CONFIG -. Il primo infatti, si differenzia dal secondo per la presenza dei dati relativi alla memoria residua e quella dei dati strumentali immagazzinati. Siccome il volume di dati di queste grandezze ammonta a 6 byte (vedi §5.2.3 e §5.2.4), ci aspettiamo che tale quantità di memoria sia pari alla differenza di dimensione tra il primo pacchetto e il secondo.

5.4.1.1 Organizzazione della memoria del primo pacchetto

Nel primo pacchetto saranno presenti sia i campi di intestazione TIPO_TMB e TMB_CONFIG, sia i dati di *telemetria* descritti nella Tabella 5.5. Affinché la stazione di Terra non interpreti in modo errato le sequenze di byte ricevute, occorre rispettare rigorosamente l'ordine riportato nella figura seguente.



Figura 5.2: Organizzazione della memoria del primo pacchetto di TMB

A sinistra della figura sopra è raffigurata l'organizzazione del primo pacchetto di TMB, mentre a destra quella del primo pacchetto di TMB Addizionato.

5.4.1.2 Organizzazione della memoria dei pacchetti successivi al primo

Nei pacchetti successivi al primo saranno presenti sia i campi di intestazione TIPO_TMB e TMB_CONFIG, sia i dati di *telemetria* descritti nella Tabella 5.6. Affinché la stazione di Terra non interpreti in modo errato le sequenze di byte ricevute, occorre rispettare rigorosamente l'ordine riportato nella figura seguente.

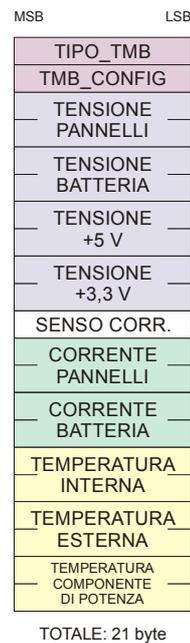


Figura 5.3: Organizzazione della memoria dei pacchetti di TMB successivi al primo

5.4.2 Organizzazione della memoria di TME

Come detto nel §5.3.3, la memoria di TME è calcolata considerando sette volte la memoria della singola orbita, due byte di intestazione (TME_CONFIG e TIPO_TME – §1.4 e §1.6) ed eventualmente la quantità di memoria destinata alla *telemetria* addizionata. Viene quindi descritta prima l'organizzazione della memoria per la singola orbita e successivamente l'organizzazione della TME (anche nella sua variante di TME Addizionata).

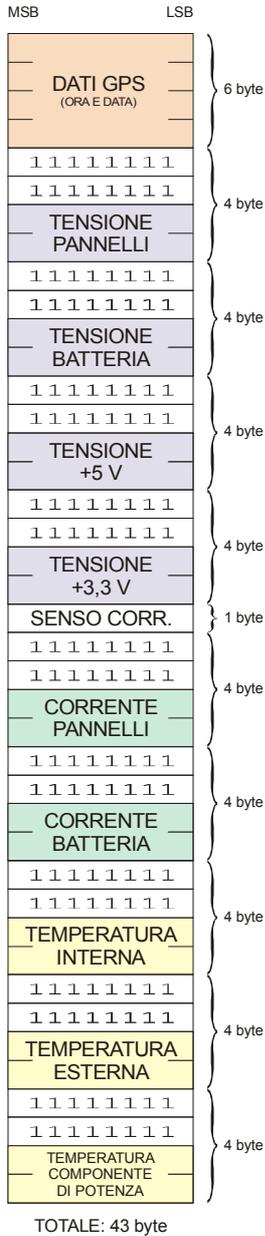
Nel raffigurare le organizzazioni di memoria, si suppone che le variazioni dei segnali delle grandezze di *telemetria* siano contenute entro i limiti imposti in §5.1.2, ovvero non si proceda all'acquisizione di un nuovo dato di riferimento.

5.4.2.1 Organizzazione della memoria per la singola orbita

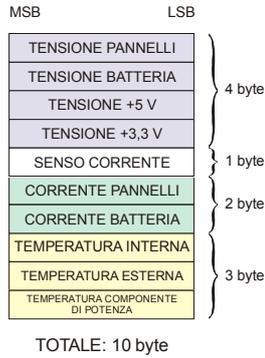
Dalla Figura 1.1 e dalla Tabella 1.3, si è potuto apprendere che durante una singola orbita si ha la successione di 63 blocchi di dati di tipo A alternati con 62 blocchi di tipo B; la successione viene terminata con il blocco di dati di tipo C. Tali blocchi, in virtù del fatto che contengono distinte combinazioni di grandezze di *telemetria*, avranno organizzazioni di memoria differenti.

Poiché viene adottata la tecnica di memorizzazione differenziale (vedi §5.2.1), ogni prima acquisizione di ciascuna grandezza di *telemetria* ad inizio orbita, viene preceduta, ad eccezione del senso della corrente della batteria, dalla coppia di byte composta da soli “1” descritta in §5.1.2 (vedi Tabella 5.2 e 5.4). Ciò è dovuto al fatto che ad ogni inizio orbita, si è stabilito di acquisire un nuovo dato di riferimento, specie se si tratta della prima acquisizione dopo il termine del *tempo di accesso* (all'inizio del *tempo di attesa* si ha l'inizio delle acquisizioni delle grandezze di TME). Per questo motivo quindi, possiamo identificare due differenti blocchi di tipo A denominati A0 e A1. Il primo in particolare si presenterà solo all'inizio di ogni orbita, mentre il secondo si alternerà per 62 volte con il blocco di dati di tipo B. Il blocco di dati di tipo A0, è l'unico che contiene al suo interno i dati di ora e data del GPS. Il blocco di tipo A0 sarà quindi organizzato in modo differente rispetto a quello A1; ciò è visibile dalla Figura 5.3, nella quale vengono riportate le quattro organizzazioni di memoria (blocchi A0, A1, B e C).

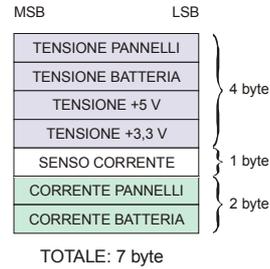
BLOCCO DI DATI DI TIPO A0



BLOCCO DI DATI DI TIPO A1



BLOCCO DI DATI DI TIPO B



BLOCCO DI DATI DI TIPO C

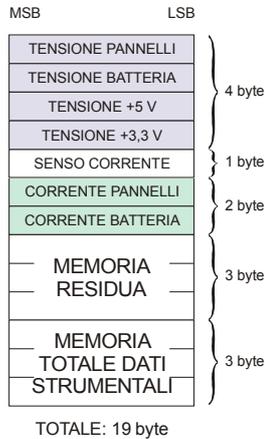


Figura 5.4: Blocchi di dati di tipo A0, A1, B e C

Se volessimo effettuare una verifica sull'occupazione di memoria per la singola orbita avremmo:

$$\begin{aligned}
 \text{Occupazione di memoria} &= \left(\begin{matrix} \text{Memoria} \\ \text{blocco A0} \end{matrix} \right) + \left(62 \times \begin{matrix} \text{Memoria} \\ \text{blocco A1} \end{matrix} \right) + \left(62 \times \begin{matrix} \text{Memoria} \\ \text{blocco B} \end{matrix} \right) \\
 \text{per la singola orbita} &+ \left(\begin{matrix} \text{Memoria} \\ \text{blocco C} \end{matrix} \right) - (\text{n}^\circ \text{ byte GPS}) \\
 &= 43 + (62 \times 10) + (62 \times 7) + 19 - 6 \\
 &= 1.110 \text{ byte}
 \end{aligned}$$

Formula 5.10: Verifica dell'occupazione di memoria della singola orbita

che è pari al risultato ottenuto in 5.7.

Rappresentando nel tempo la successione dei blocchi sopra raffigurati abbiamo:

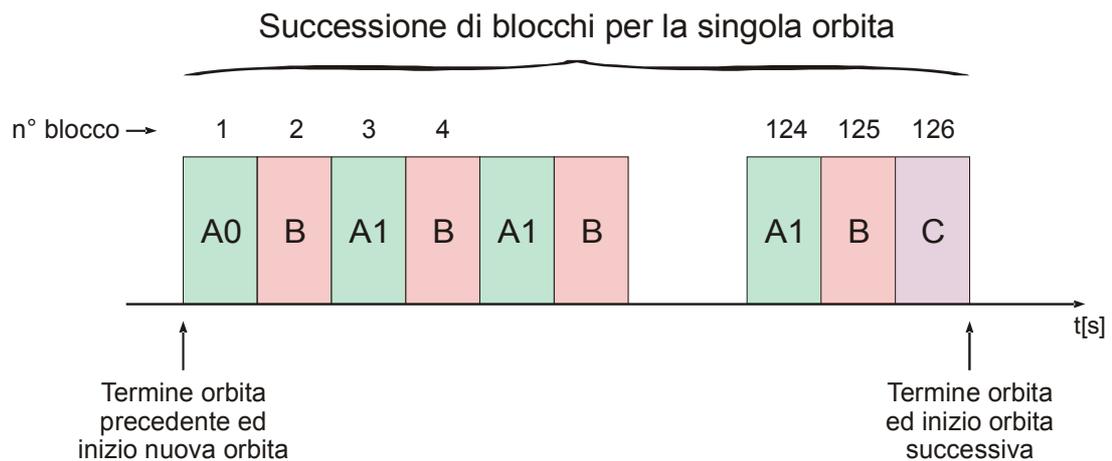


Figura 5.5: Successione dei blocchi di dati A0, A1, B e C durante la singola orbita

5.4.2.2 Organizzazione della memoria di TME

Si procede ora alla rappresentazione grafica dell'organizzazione della memoria per la TME, quindi con sette orbite consecutive. Si fa presente che, come detto nel §1.4, la settima orbita potrebbe avere una quantità di dati inferiore a quelle precedenti, in quanto la TME si interrompe nel momento in cui il satellite riceve il segnale dalla stazione di Terra (nella Figura 5.2 viene indicata l'occupazione massima che si può avere).

Nella rappresentazione che segue, vengono considerati i campi di intestazione (prelevati dai registri interni al $\mu\text{C MASTER}$), nonché, nel caso in cui siano stati attivati, i dati di *telemetria* addizionali (hanno occupazione fissa e pari a 5 byte - vedi §5.3.4 e Tabella 5.3).



Figura 5.6: Organizzazione della memoria di TME

5.5 Occupazione di memoria dei dati strumentali

Come detto nel §3.6, durante le orbite del satellite verranno effettuate le misurazioni tramite gli strumenti di bordo ogni 237 secondi (tempo di *default* e modificabile tramite *telecomando*). I dati acquisiti verranno collocati nella memoria di massa.

Considerando un tempo di acquisizione dei dati strumentali pari alla somma dei tempi massimi di *accesso* e di *attesa*, siamo in grado di sapere, assegnata la cadenza delle misure, la dimensione massima che si dovrà riservare nella memoria di bordo.

$$\begin{aligned} \text{Tempo massimo totale} &= \text{Tempo massimo di accesso} + \text{Tempo massimo di attesa} \\ &= 301 + 46.620 = 46.921 \text{ secondi} \end{aligned}$$

Formula 5.11: Calcolo del tempo massimo tra due accessi consecutivi

$$\begin{aligned} \text{Numero massimo} &= \left\lfloor \frac{\text{Tempo massimo totale}}{\text{Tempo della cadenza} \times 7} \right\rfloor \\ \text{di misure per orbita} &= \left\lfloor \frac{46.921}{237 \times 7} \right\rfloor = 29 \end{aligned}$$

Formula 5.12: Calcolo del numero massimo di misure per orbita

Il volume di dati prodotti per la singola orbita ammonta a:

| Strumento di misura | n° di byte |
|---------------------|------------|
| GPS | 2.320 |
| Dosimetro | 59.392 |
| Magnetometro | 174 |

Totale 61.886

Tabella 5.8: Quantità di memoria prodotta dagli strumenti nel periodo di un'orbita

Dopo sette orbite avremo quindi un'occupazione di memoria pari a

$$\text{Occupazione di memoria per sette orbite} = 61.886 \times 7 = 433.202 \text{ byte}$$

Formola 5.13: Calcolo della occupazione di memoria degli strumenti dopo sette orbite

5.6 Considerazioni

L'occupazione di memoria, da parte dei dati di *telemetria*, è stata calcolata sulla base del tempo medio dell'orbita. Per stimare la massima occupazione di memoria si sarebbe dovuto considerare il tempo massimo della singola orbita. Dal momento che dalla Formola 1.1 e dai ragionamenti conseguenti, si è deciso di considerare sette orbite, si è effettuata per l'appunto una sovrastima. Si impone inoltre che, nel caso si verificano più di sette orbite, il $\mu\text{C}_{\text{MASTER}}$ non provvede ad avviare le acquisizioni dei dati: in questo modo l'occupazione massima di memoria calcolata nella Formola 5.8 è valida.

Considerando l'occupazione massima di memoria della TME (Formola 5.8) e degli strumenti di bordo (Formola 5.7), desumiamo che la memoria di bordo dovrà disporre di almeno

$$\begin{aligned} \text{Occupazione totale di memoria} &= \left(\begin{array}{l} \text{Massima occupazione di} \\ \text{per sette orbite} \end{array} \right) + \left(\begin{array}{l} \text{Massima occupazione di} \\ \text{memoria dei dati di TME} \end{array} \right) + \left(\begin{array}{l} \text{Massima occupazione di} \\ \text{memoria degli strumenti} \end{array} \right) \\ &= 28.406 + 433.202 = 461.608 \text{ byte} \end{aligned}$$

Formula 5.14: Calcolo della occupazione di memoria degli strumenti dopo sette orbite

Se quindi all'interno del satellite inseriamo quattro memorie del tipo 25LC1024, dalla capacità complessiva di $131.072 \cdot 4 = 524.288$ byte, abbiamo un avanzo di 62.680 byte.

Da tener presente che questi dati sono ottenuti considerando una cadenza di acquisizione dei dati degli strumenti di bordo di 237 secondi.

Capitolo 6

Applicazione del telecomando

Viene infine riportato un esempio di utilizzo del *telecomando* sul microcontrollore considerato in questo elaborato.

La stesura del *firmware* del μC , è stata resa possibile tramite l'utilizzo di un kit di sviluppo software fornito gratuitamente dalla Microchip: l'MPLAB IDE (Figura 6.2). La programmazione invece dell'integrato è avvenuta tramite l'ICD2 (*InCircuit Debugger 2* - Figura 6.1). Tale oggetto svolge sia la funzione di programmatore, sia quella di debugger; quest'ultima modalità in particolare consente di leggere, tramite l'applicativo MPLAB IDE, tutti i settaggi dei registri del μC , consentendo una più semplice individuazione dei bug (errori del *firmware*).



Figura 6.1: ICD2

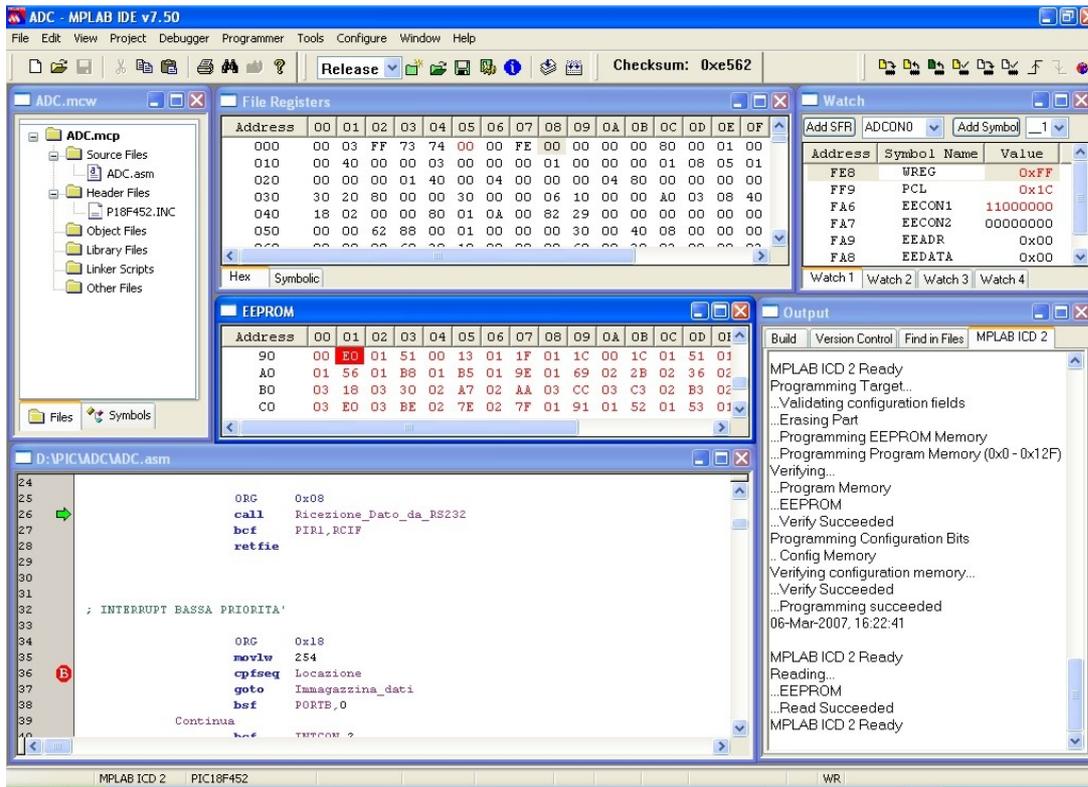


Figura 6.2: Screenshot dell'MPLAB IDE

Si è fatto inoltre uso di una *Demoboard* (prodotta sempre dalla Microchip e raffigurata sotto).

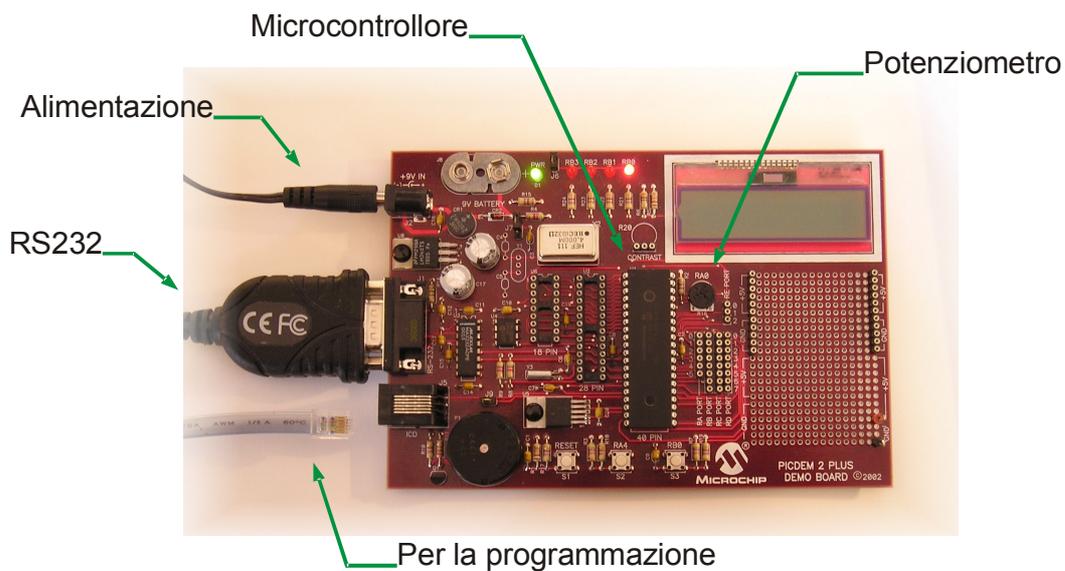


Figura 6.3: Demoboard della Microchip

Nelle pagine seguenti viene riportato il listato del *firmware*:

```

1  ; Firmware esemplificativo del telecomando
2  ;
3  ; Premendo il tasto "s" della tastiera si avvia l'acquisizione
4  ; dall'ADC (tensione 0-5 V), premendolo di nuovo la si termina
5  ; Con il tasto "t" si procede al trasferimento al Pc tramite RS232
6
7
8          PROCESSOR    18F452
9          RADIX        DEC
10         INCLUDE      "P18F452.INC"
11
12
13         ORG          0x00
14 Delay          RES    1
15 AD_MSB         RES    1
16 AD_LSB         RES    1
17 Car_Start      RES    1
18 Car_di_invio_dati RES 1
19 Locazione      RES    1
20 Avanzamento_Locaz RES 1
21
22
23         ORG          0x00
24         goto         Start
25
26
27
28 ; INTERRUPT ALTA PRIORITA'
29
30         ORG          0x08
31         call         Ricezione_Dato_da_RS232
32         bcf          PIR1,RCIF
33         retfie
34
35
36
37 ; INTERRUPT BASSA PRIORITA'
38
39         ORG          0x18
40         movlw        254
41         cpfseq       Locazione
42         goto         Immagazzina_dati
43         bsf          PORTB,0
44         Continua
45         bcf          INTCON,2
46         retfie
47
48
49
50 Ricezione_Dato_da_RS232
51         movf         RCREG,W
52         cpfseq       Car_di_invio_dati           ; Se è uguale salta
53         goto         Continua_ricezione_dato_da_RS232
54         bcf          TOCON,TMR0ON
55         clrf        Avanzamento_Locaz
56         call         Invio_dati
57
58         Continua_ricezione_dato_da_RS232
59         cpfseq       Car_Start
60         return
61         btfs         TOCON,TMR0ON                ; Il timer era già attivo?
62         goto         Avvio_Timer
63         bcf          TOCON,TMR0ON                ; Spengo il timer
64         return
65
66
67 Avvio_Timer
68         bsf          TOCON,TMR0ON
69         clrf        Locazione
70         clrf        PORTB
71         return
72
73
74
75

```

```

76
77 ;*****
78 ;*****
79
80
81 ; *** INIZIO PROGRAMMA ***
82
83 Start ; Configurazioni porte
84     movlw B'11111111'
85     movwf TRISA
86
87     movlw B'00000000'
88     movwf TRISB
89
90     movlw B'11111111'
91     movwf TRISC
92
93     movlw B'11111111'
94     movwf TRISD
95
96     movlw B'11111111'
97     movwf TRISE
98
99     clrf  PORTB
100
101
102 ; *** CONFIGURAZIONE ADC ***
103
104     movlw B'00000000'
105     movwf ADCON0
106
107     movlw B'10001110'
108     movwf ADCON1
109
110 ; *** FINE CONFIGURAZIONE ADC ***
111
112
113
114 ; *** IMPOSTAZIONE TIMER 0 ***
115
116     movlw B'00000011' ; Overflow ogni 1,048 secondi
117     movwf TOCON
118
119 ; *** FINE CONFIGURAZIONE TIMER 0 ***
120
121
122
123 ; *** CONFIGURAZIONE INTERRUPT ***
124
125     movlw B'01100000'
126     movwf INTCON
127     bcf  INTCON2,TMR0IP ; Interrupt Timer 0 bassa priorità
128
129     bsf  PIE1,RCIE
130     bcf  PIR1,RCIF
131     bsf  IPRI,RCIP ; Interrupt RX USART alta priorità
132
133     bsf  RCON,IPEN ; Abilitazione priorità
134
135 ; *** FINE CONFIGURAZIONE INTERRUPT ***
136
137
138
139 ; *** CONFIGURAZIONE PORTA USART ***
140
141     movlw 12
142     movwf SPBRG
143
144     bsf  TXSTA,BRGH ; High baud rate
145     bcf  TXSTA,SYNC
146     bcf  TXSTA,TX9 ; No 9° bit
147     bsf  TXSTA,TXEN ; Abilitazione trasmissione
148
149     bsf  RCSTA,SPEN
150     bsf  RCSTA,CREN

```

```

151
152             ; *** FINE CONFIGURAZIONE PORTA USART ***
153
154
155
156             movlw   's'
157             movwf   Car_Start
158
159             movlw   't'
160             movwf   Car_di_invio_dati
161
162             clrfl   Locazione
163             clrfl   Avanzamento_Locaz
164
165             bsf     INTCON,GIE             ; Abilitazione interrupt
166
167
168
169 Loop
170             goto    Loop
171
172
173
174
175 ; *** SUBROUTINE ***
176
177
178 AD_Converzion
179             bsf     ADCON0,ADON
180             movlw   15
181             movwf   Delay
182             decfsz  Delay
183             goto    $-1
184
185             bsf     ADCON0,GO
186             btfsc  ADCON0,GO
187             goto    $-1
188
189             movf   ADRESH,W
190             movwf  AD_MSB
191             movf   ADRESL,W
192             movwf  AD_LSB
193
194             bcf     ADCON0,ADON
195
196             return
197
198
199 Immagazzina_dati
200             call    AD_Converzion
201
202             movf   Locazione,W
203             movwf  EEADR
204
205             movf   AD_MSB,W
206             movwf  EEDATA
207             bcf     EECON1,EEPGD
208             bcf     EECON1,CFGS
209             bsf     EECON1,WREN
210
211             bcf     INTCON,GIE
212
213             movlw   0x55
214             movwf  EECON2
215             movlw   0xAA
216             movwf  EECON2
217
218             bsf     EECON1,WR
219
220             btfsc  EECON1,WR
221             goto    $-1
222
223             incf   Locazione,1
224
225             movf   Locazione,W

```

```

226          movwf  EEADR
227
228          movf   AD_LSB,W
229          movwf  EEDATA
230          bsf   EECON1,WREN
231
232          movlw  0x55
233          movwf  EECON2
234          movlw  0xAA
235          movwf  EECON2
236
237          bsf   EECON1,WR
238
239          btfsc EECON1,WR
240          goto  $-1
241
242          incf   Locazione,1
243
244          bsf   INTCON,GIE
245
246          goto  Continua
247
248
249
250  Invio_dati                                     ; Invio dei dati acquisiti
251                                               ; con l'ADC alla EEPROM
252          movf   Avanzamento_Locaz,W
253          movwf  EEADR
254          bsf   EECON1,RD
255          movf   EEDATA,W
256          movwf  TXREG
257
258          movlw  250
259          movwf  Delay
260          decfsz Delay
261          goto  $-1
262
263          incf   Avanzamento_Locaz,1
264          movf   Locazione,W
265          cpfseq Avanzamento_Locaz
266          goto  Invio_dati
267          clrf   Locazione
268
269          return
270
271
272          END
273

```

Una volta programmato il μC , occorre collegare la Demoboard come indicato nella figura sottostante. Come si può notare la comunicazione tra il PC e il circuito stampato dimostrativo, è realizzata tramite la porta RS232 (la Demoboard è dotata di un traslatore di livello).

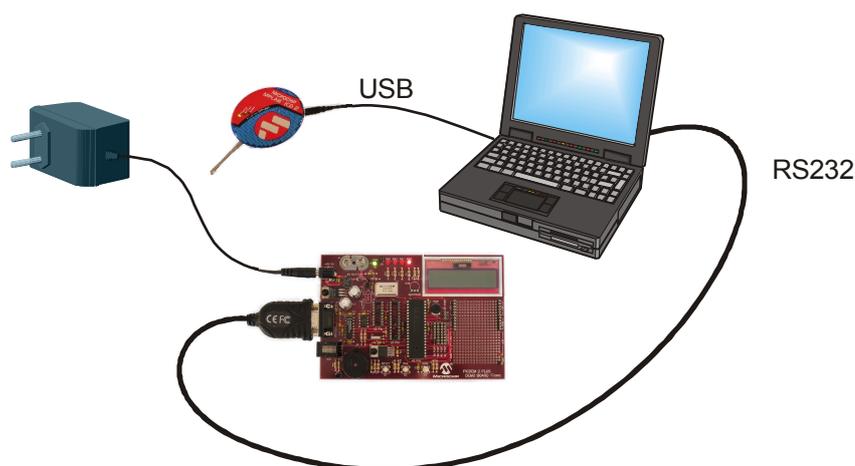


Figura 6.4: Collegamento Demoboard

Una volta configurata la porta RS232 del PC (a 19200 baud, con dati a 8 bit e senza bit di parità), si può procedere all'utilizzo del μC . Tramite la trasmissione del carattere "s", quest'ultimo acquisisce e memorizza (nella sua memoria EEPROM), con cadenza di circa 1 secondo, la tensione proveniente dal potenziometro (vedi Figura 6.3). Nel momento in cui viene inviato nuovamente il carattere "s", tale acquisizione viene interrotta. Tramite invece l'invio del carattere "t", il μC provvede all'invio dei dati immagazzinati.

Nel caso in cui la memoria venisse saturata, il μC provvederà ad interrompere la procedura di immagazzinamento; quest'ultima situazione si può constatare tramite l'accensione di un LED rosso (vedi Figura 6.3).

Per l'invio dei caratteri di comando, è stato utilizzato il software TeraTerm (un software videoterminale), mentre, per il monitoraggio del transito di byte sulla porta seriale (come *sniffer*), è stato utilizzato il software Serial Port Monitor. Gli *screenshot* di entrambi i programmi sono visibili nelle seguenti Figure.

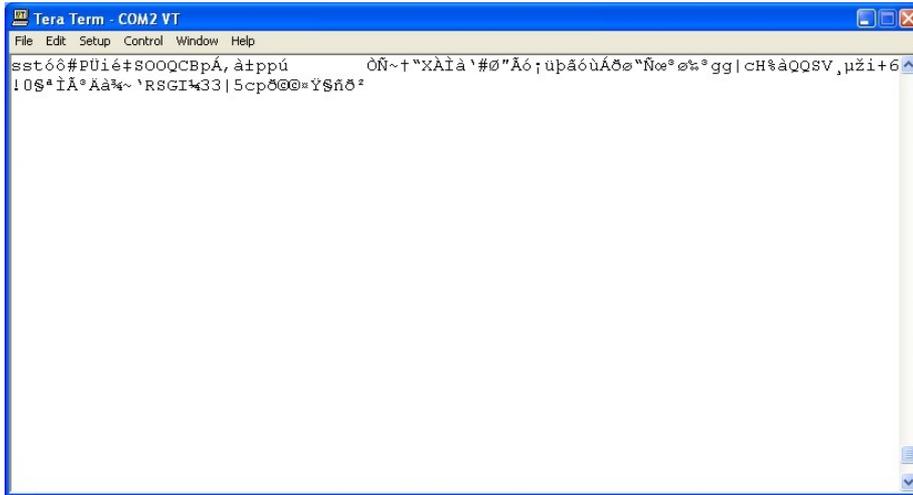


Figura 6.5: Screenshot del software TeraTerm

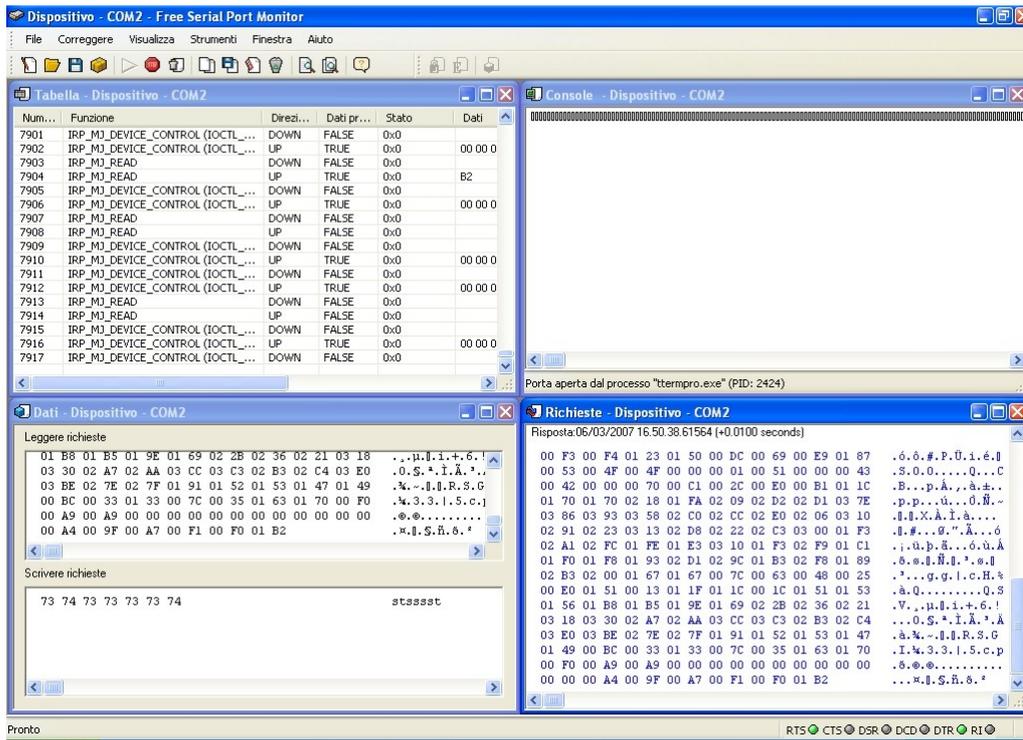


Figura 6.6: Screenshot del software Serial Port Monitor

Per quanto riguarda la Figura 6.6, si osserva che i dati ricevuti (finestra in basso a destra) sono espressi sia tramite codice esadecimale, sia per mezzo del codice ASCII.

Capitolo 7

Conclusioni

Durante l'elaborazione del presente documento, sono emerse alcune problematiche le quali hanno avuto, in alcuni casi, una parziale soluzione. L'esempio lampante è il problema dell'inserimento dello *switch* AD736 per commutare il bus SPI del $\mu\text{C}_{\text{SLAVE}}$ o verso il magnetometro o verso il $\mu\text{C}_{\text{MASTER}}$. Si ricorda che tale commutatore è stato inserito per permettere la misura del campo magnetico anche nei momenti in cui il bus SPI è occupato dal $\mu\text{C}_{\text{MASTER}}$ per la comunicazione con i rimanenti sottosistemi. L'inserimento di questo integrato viene valutato come una problematica in quanto, nella norma, il bus SPI non viene sezionato. Una possibile soluzione, come detto nel capitolo 3 è quella di rimandare la misura del magnetometro nel momento in cui il bus viene liberato. La misura dilazionata comporta però un errore di posizione in quanto, durante il tempo in cui il bus non è disponibile, il satellite compie un arco di orbita non trascurabile (il satellite viaggia a 7 Km/s) quindi, le coordinate ricavate attraverso il GPS, non sarebbero corrette per la misura effettuata. Un'ulteriore alternativa è quella di implementare il modulo SPI sulle rimanenti porte inutilizzate del μC . Quest'ultima sembrerebbe la soluzione ottimale, ma tale possibilità non è di immediata realizzazione.

Un altro problema (di facile soluzione) in cui ci si è imbattuti, era la scelta dei componenti: occorre infatti ricercare integrati a basso consumo, in grado di essere posti in modalità *shutdown* e di operare alla tensione di 3,3V. Come si può notare infatti dallo schema generale, questa tensione è utilizzata per la maggior parte degli integrati, consentendo un abbassamento della potenza impiegata.

Per quanto riguarda la scelta della memoria, si è visto dal capitolo 5, che l'inserimento di quattro memorie del tipo 25LC1024 sono sufficienti ad immagazzinare sia i dati di *telemetria*, sia quelli provenienti dagli strumenti di bordo (per la cadenza di 237 secondi). Tale memoria inoltre, risulta essere quella di maggiore capacità individuata sul mercato che consenta una comunicazione sul bus SPI. Si è pensato anche di inserire al loro posto una memoria RAM di maggior dimensione. Questa però avrebbe richiesto l'inserimento di un'ulteriore circuiteria dedicata alla sua gestione in quanto, questo tipo di memoria, è generalmente reperibile ad accesso parallelo (integrati da almeno 40 pin con *footprint* BGA).

Giunti a tal punto del lavoro, è possibile operare alcune considerazioni sui possibili argomenti da approfondire affinché il sistema complessivo tenda sempre più ad un lavoro finito e funzionale. Vengono di seguito riportati i possibili spunti di riflessione e sviluppo.

◆ **Divisione della memoria**

Al termine del capitolo 5, si è giunti alla conclusione che per immagazzinare la *telemetria* e i dati strumentali sono necessarie quattro memoria 25LC1024 (con un avanzo di 62.680 byte). A tal punto occorre stabilire con che ordine devono essere inseriti i dati al loro interno ovvero come “spartire” le locazioni alle due tipologie di dati. Occorre inoltre procedere alla stesura del *firmware* del $\mu\text{C MASTER}$ per l'immagazzinamento dei dati (verranno usate due porte del $\mu\text{C MASTER}$, e un multiplexer, per selezionare la memoria corretta).

◆ **Instaurazione della comunicazione**

Al primo accesso della stazione di Terra con il satellite, è necessaria l'autenticazione tramite l'invio di opportune credenziali. Si tratta di stabilire come deve essere composto il pacchetto ovvero cosa, e come, il campo dati del pacchetto AX.25 deve contenere. Occorre stabilire inoltre come il satellite risponde all'arrivo del precedente messaggio e quando ha inizio la comunicazione vera e propria dei dati della memoria di bordo.

◆ **Trasmissione a Terra**

Occorre stabilire come avviene la trasmissione dei dati a Terra ovvero quanti dati devono essere inviati per poi configurare l'integrato CC1100 come ricevitore (si tenga presente il limitato, e variabile, *tempo di accesso*). Occorre inoltre definire come avviene la comunicazione al satellite dell'avvenuta ricezione (corretta o errata) dei pacchetti. Si pensava di numerare i pacchetti e trasmettere al satellite, come risposta, l'avviso di quelli che sono giunti errati. Nel caso in cui si verifici tale situazione, il satellite provvederà al loro rinvio (i dati di TME meno recenti e tutti quelli strumentali vengono cancellati ad inizio del *tempo di attesa*).

◆ **Controllo automatico di frequenza**

A causa dell'effetto doppler dovuto all'avvicinamento del satellite alla stazione di Terra e, successivamente, il suo allontanamento, si ha uno scostamento della frequenza. Occorre implementare un algoritmo che provveda ad alterare l'oscillatore del ricevitore e a riportarlo nella giusto punto di lavoro (è da vedere se quello di Terra o quello del satellite).

- ◆ **Costruzione del pacchetto AX.25**

È necessario procedere alla stesura del *firmware* dedicato alla costruzione del pacchetto secondo il protocollo AX.25. Tale compito dovrà assolverlo il μC MASTER. Occorre quindi, oltre all'immissione dei dati e al completamento di alcuni campi, calcolare il FCS (*Frame Check Sequence*) nonché provvedere all'inserimento del bit di *stuffing*.

- ◆ **Acquisizione dei dati**

Occorre procedere alla stesura del *firmware* per l'automatizzazione delle acquisizioni dei dati di *telemetria* nonché quelli dagli strumenti di bordo. Nella stesura del *firmware* relativo al μC SLAVE, occorre tenere presente che, all'inizio della comunicazione master e slave, sarà contenuto un campo che identificherà la tipologia dei dati che verranno trasmessi di seguito, ovverosia se questi si trattano di un *telecomando* o di una richiesta di data e ora dal GPS.

Bibliografia

Pubblicazioni

- Bruce R. Elbert - The Satellite Communication Applications Handbook - Artech House, Inc.
- Michael O. Kolawole - Satellite Communication Engineering - CRC Press
- Tim Prat - Satellite Communication - IEE ILP
- **Ray E. Sheriff and Y. Fun Hu** - Mobile satellite communication networks - Wiley
- Tanzilli - Programmazione dei PIC
- Andrea Sbrana - La programmazione dei microcontrollori PIC - Ed. ETS
- Randall Hyde - The Art Of Assembly Language - 2002 Hyde
- Tesi - Marco De Din - Progetto preliminare di un sistema di gestione energetica per il satellite ATMOCUBE
- Tesi - Stefano Punis - Progetto di un circuito a microcontrollore per la gestione del ricetrasmittitore impiegato nel satellite ATMOCUBE

Siti WEB

- www.rscomponents.it - RS Components
- www.distrelec.it - Distrelec
- www.farnellinone.it - Farnell InOne
- www.microchip.com - Microchip
- www.kevin.it - Kevin Schurter Spa
- www.analog.com - Analog Devices
- www.ti.com - Texas Instruments

Datasheet

- *Datasheet* P18F452 – Microchip
- *Datasheet* 25LC1024 – Microchip
- *Datasheet* MAX471 – Maxim
- *Datasheet* MAX333A – Maxim
- *Datasheet* ADG708 – Analog Devices
- *Datasheet* AD5300 – Analog Devices
- *Datasheet* ADG7360 – Analog Devices
- *Datasheet* ADS8343 – Burr-Brown
- *Datasheet* DG9415 – Vishai
- *Datasheet* CC1100 – Chipcon

Ringraziamenti

Un doveroso ringraziamento va al prof. Fragiacomò per avermi seguito durante lo sviluppo della mia tesina e per avermi fornito i suoi consigli oltre che le sue conoscenze.

Un ringraziamento particolare va a Marzia per avermi sopportato e confortato in questi mesi un po' più pesanti del solito.

Alla lista non possono mancare i miei genitori, che mi hanno sempre spronato a darci dentro, i miei fratelli – e sorella – (in particolare Guglielmo), i nonni e gli zii.

I miei ringraziamenti vanno anche all'A.P.E. Research S.r.l. (in particolare agli elettronici – Stefano C., Paolo e Massimo), per avermi permesso di frequentare le lezioni e dato la possibilità di dedicarmi alla tesi a pieno ritmo nell'ultimo periodo.

Infine volevo ringraziare tutti i miei amici per il loro sostegno nel corso di questi anni, con l'invito a fare altrettanto nei prossimi in attesa della seconda tornata di esami e di tesi.