

**POLITECNICO DI MILANO**

Facoltà di Ingegneria Industriale

Corso di Laurea specialistica in Ingegneria aeronautica



Progettazione e realizzazione della nuova  
generazione di un sistema di strumentazione per le  
prove di volo di velivoli ultraleggeri

Relatore: Prof. Alberto Rolando

Tesi di Laurea di:

Roberto Riva Matr. 724552

Anno Accademico 2012 – 2013



# Sommario

Questa tesi tratta della progettazione e realizzazione del prototipo della nuova generazione del sistema FTI Mnemosine MK IV.

Lo sviluppo di questo nuovo FTI è stato voluto dal Dipartimento di Scienze e Tecnologie Aerospaziali del Politecnico di Milano a seguito dell'esperienza operativa maturata con la versione precedente.

Mnemosine MK IV come tutti i suoi predecessori, è stato progettato specificatamente per essere utilizzato durante la sperimentazione di velivoli della classe ULM (Ultra Light Machine).

I requisiti della nuova generazione MK IV sono stati pensati e scelti a seguito di un lavoro di sviluppo e ottimizzazione che ha interessato il sistema Mnemosine MKIII per diversi anni.

La principale innovazione riguarda la presenza di un unico nodo che integra tutti quei moduli che nella vecchia generazione erano collocati all'interno del cockpit del velivolo.

La realizzazione del prototipo hardware del modulo madre è partita con la scelta del microcontrollore e la progettazione degli schematici, è proseguita con il disegno e realizzazione dei PCB ed è terminata con il montaggio dei componenti elettronici sulle schede stesse.

Sono stati poi eseguiti vari test con lo scopo di verificare il corretto funzionamento dell'hardware che hanno portato ad alcune modifiche per correggere errori di progettazione e per implementare ulteriori funzionalità sul prototipo.

È stato poi concluso il progetto degli schematici del nuovo nodo dati aria.

Il presente documento è composto da un'introduzione in cui si descrive la storia degli FTI e la categoria ULM, dal progetto vero e proprio del nodo principale, dal progetto preliminare degli schematici di un nuovo nodo dati aria, da una descrizione del software Real time che verrà eseguito dal microcontrollore e dei vari task che lo compongono, e da un capitolo conclusivo in cui vengono descritti i possibili futuri sviluppi del progetto del Sistema FTI Mnemosine MK IV



# Abstract

This thesis presents the design and realization of a prototype of the FTI Mnemosine MK IV.

The FTI development was required by “Dipartimento di scienze e tecnologie aerospaziali” the “Politecnico di Milano” in consideration of the good experience gained with the previous release.

This device, as all its predecessors, was specifically designed to be used during the tests of ultralight motorized aircraft.

All requirements for the prototype MK IV are chosen and met through the optimization of Mnemosine MKIII, which has been done since 2005 to the present day.

The most important advanced development concerns the presence of a central node that integrates all modules. In the older generation these modules were placed inside the cockpit of the aircraft.

The prototyping of the principal module started with the choice of the microcontroller and the design of schematics, followed by the design and manufacture of PCBs. Finally the realization was finished with the installation of the electronic components on these cards.

Some tests were performed to verify the correct work of the hardware, which brought to changes in design and implementing additional functionality in the prototype.

There is also a preliminary draft of the schematic of the new AirDataComputer with the first dimension sizing of PCB.

This document consist of five parts: an introduction in which is described the history of the FTI and the ultralight category; a project of the main node; a preliminary draft of the schematic of the new air data computer; a description of the software "Real time" that will be executed by the microcontroller and the various tasks that compose it; and a conclusive chapter which describes the future development of the project.

# Indice

<b>1</b>	<b>Introduzione .....</b>	<b>1</b>
1.1	Storia dell'FTI.....	1
1.2	UltraLeggero a Motore (ULM) .....	1
1.3	Flight Test Instrumentation (FTI).....	2
<b>2</b>	<b>Mnemosine MK III.....</b>	<b>4</b>
2.1	Definizione dei requisiti per Mnemosine MK IV.....	7
<b>3</b>	<b>Il modulo principale .....</b>	<b>10</b>
3.1	Architettura e dimensioni delle schede.....	10
3.2	Power Supply Unit .....	12
3.3	Scheda Olimex STM32-E407 .....	15
3.4	Ingressi seriali .....	17
3.4.1	RS232 (EIA RS232) .....	18
3.4.2	RS485 (EIA RS485) .....	19
3.4.3	RS422 (EIA RS422) .....	20
3.4.4	Circuito per le porte seriali .....	21
3.5	Ingressi analogici.....	25
3.5.1	Porta di comunicazione per gli ingressi analogici.....	25
3.5.2	Il circuito di condizionamento del segnale .....	26
3.5.3	Disaccoppiatore ottico .....	26
3.6	Interfaccia Can-Bus .....	29
3.7	Modifiche e correzioni .....	31
3.7.1	Implementazione del circuito Driver per il Can-Bus.....	31
3.7.2	Correzioni e completamento degli stampati .....	34
<b>4</b>	<b>Progetto del nodo dati aria .....</b>	<b>37</b>
4.1	Architettura del nodo dati aria .....	37
4.1.1	Il microcontrollore del nodo dati aria: Olimex STM32-H107 .....	39
4.1.2	I sensori di pressione .....	40
4.1.3	Sensori per AOA e SSA.....	42
4.1.4	Sensore di temperatura totale .....	43
4.1.5	Interfaccia Can-Bus del modulo dati aria .....	43
4.2	Definizione dell'architettura interna e degli ingombri .....	44
<b>5</b>	<b>Sistema Operativo Real-Time (RTOS).....</b>	<b>48</b>
5.1	Scelta dell'RTOS .....	49

<b>5.2</b>	<b>Descrizione dei vari Threads .....</b>	<b>50</b>
5.2.1	Main .....	51
5.2.2	Time Scheduler.....	51
5.2.3	SD Thread .....	52
5.2.4	Ethernet Thread .....	53
5.2.5	Can Thread .....	53
5.2.6	GPS Thread .....	54
5.2.7	UART Threads: AHRS, CDU e Stick Force .....	54
5.2.8	Control surface position Thread.....	56
5.2.9	Air Data Thread .....	57
<b>6</b>	<b>Conclusioni e sviluppi futuri.....</b>	<b>58</b>
	<b>Bibliografia.....</b>	<b>59</b>

## Indice delle figure

Figura 2.1 - Architettura del sistema FTI Mnemosine MK3.....	5
Figura 2.2 - Alcuni moduli di Mnemosine MK3 installati nella cockpit .....	6
Figura 2.3 - Architettura di Mnemosine MK4 .....	9
Figura 3.1 - Architettura interna del modulo madre di Mnemosine MK IV .....	11
Figura 3.2 - Schematico Power Supply Unit.....	12
Figura 3.3 - Schema a blocchi della Power Supply Unit.....	13
Figura 3.4 - Olimex STM32-E407 layout .....	15
Figura 3.5 - Esempio del protocollo di trasmissione Rs232.....	18
Figura 3.6 - Ricevitore GPS U-Blox Lea4T in tecnologia SMD .....	21
Figura 3.7 - Schema concettuale per il multimode serial port.....	22
Figura 3.8 - Schematico multimode serial port A .....	24
Figura 3.9 - Layout del frontale .....	25
Figura 3.10 - Andamento degli errori dei canali analogici.....	29
Figura 3.11 - Schematico Driver Can-Bus .....	30
Figura 3.12 - Realizzazione sperimentale del circuito Driver Can-Bus .....	31
Figura 3.13 - Schema del circuito Driver Can-Bus progettato inizialmente .....	32
Figura 3.14 - Schema del circuito driver Can-Bus realizzato .....	33
Figura 3.15 - Schede interconnesse con il flat-cable 40 pin.....	34
Figura 3.16 - Vista della parte inferiore della Masterboard con modifiche .....	36
Figura 4.1 - Schema a blocchi del nodo dati aria .....	38
Figura 4.2 - Struttura di un Pitot-Boom .....	38
Figura 4.3 - Total Temperature Probe .....	39
Figura 4.4 - Scheda del microcontrollore STM32-H107 .....	39
Figura 4.5 - Sensore HCLA005EU .....	40
Figura 4.6 - Sensore HCA0611ARHS .....	41
Figura 4.7 - Schematico del collegamento dei sensori di pressione.....	41
Figura 4.8 - Circuito di condizionamento del segnale per AOA e SSA.....	42
Figura 4.9 - Total temperatur probe .....	43
Figura 4.10 - Driver Can-Bus nodo dati aria.....	43
Figura 4.11 - Architettura modulo dati aria.....	44
Figura 4.12 - Organizzazione dei componendi del nodo dati aria all'interno del case...	45
Figura 4.13 - Vista posteriore del nodo dati aria con i due connettori CAN-BUS.....	46
Figura 4.14 - Vista superiore del PCB del nodo dati aria.....	46
Figura 4.15 - Vista inferiore del PCB del nodo dati aria.....	47
Figura 5.1 - Diagramma agli stati del driver SDC .....	52
Figura 5.2 - Diagramma agli stati del Driver CAN.....	53
Figura 5.3 - Diagramma a blocchi per il Driver di comunicazione seriale .....	54
Figura 5.4 - Diagramma a blocchi per comunicazione UART.....	55
Figura 5.5 - Diagramma per la trasmissione UART .....	55
Figura 5.6 - Diagramma per la Ricezione UART .....	55
Figura 5.7 - Diagramma per la gestione ADC .....	56
Figura 5.8 - Diagramma agli stati del driver I <sup>2</sup> C.....	57



## Indice delle tabelle

Tabella 3.1 - Dimensioni standard Eurocard .....	10
Tabella 3.2 - Configurazione modalità di comunicazione .....	17
Tabella 3.3 - Collegamenti ricevitore GPS .....	21
Tabella 3.4 - Pin del microcontrollore dedicati alla multimode serial interface Port A	23
Tabella 3.5 - Pin del microcontrollore dedicati alla multimode serial interface Port B.	23
Tabella 3.6 - Pin del microcontrollore dedicati alla multimode serial interface Port C.	23
Tabella 3.7 - Pinout delle porte seriali .....	24
Tabella 3.8 - Taratura ingressi analogici.....	27
Tabella 3.9 - Tensioni previste ipotizzando la linearità del circuito. ....	27
Tabella 3.10 - Test di linearità degli ingressi analogici .....	28
Tabella 3.11 - Errori di linearità dei canali analogici.....	28
Tabella 3.12 - Pinout del connettore del Can-Bus .....	30
Tabella 3.13 - Pinout del connettore 40 pin. ....	33

## Lista degli acronimi

ADC	Air Data Computer
ADC	Analog to Digital Converter
AeCI	Aero Club Italia
AHRS	Attitude Heading Reference System
AOA	Angle Of Attack
API	Application Program Interface
ARM	Advanced RISC Machine
ATZ	Air Track Zone
CAFFE	CAn For Flight-test Equipment
CAN	Controller Area Network
CCM	Core Coupled Memory
CDU	Command and Display Unit
CPU	Central Processing Unit
CS-LSA	Certification Specification Light Sport Aircraft
CS-VLA	Certification Specification Very Light Aircraft
CTS	Clear To Send
DAC	Digital to Analog Converter
DCE	Data Communication Equipment
DTE	Data Terminal Equipment
EGT	Exhaust Gas Temperature
ELT	Emergency Locator Transmitter
FPU	Floating Point Unit
FT	Flight Test
FTE	Flight Test Engineer
FATFS	File Allocation Table File System
FTI	Flight Test Instrumentation
GND	Ground
GNU	Gnu's Not Unix
GPL3	General Purpose Licence V3
GPS	Global Positioning System
HAL	Hardware Abstraction Layer
PC	Inter Integrated Circuit
ISR	Interrupt Service Routine
JTAG	Joint Test Action Group
lwIP	light-weight Internet Protocol
LSB	Least Significant Byte
MPU	Memory Protection Unit
MSB	Most Significant Byte

MSPS	Mega-Sample Per Second
PCB	Printed Circuit Board
PLC	Programmable Logic Controller
PROFet	PROtected Fet
PSU	Power Supply unit
PTH	Plating Through Holes
PWM	Pulse Width Modulation
RAM	Random Access Memory
RISC	Reduced Instruction Set Code
RMS	Rate Monotonic Scheduling
ROM	Read-Only Memory
RPM	Revolutions Per Minute
RTOS	Real Time Operating System
RTS	Ready To Send
SCL	Serial CLock Line
SDA	Serial DATA line
SDCard	Secure Digital Card
SDIO	Secure Digital Input/Output
SFTE	Society of Flight Test Engineer
SMD	Surface Mounting Device
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SDIO	Secure Digital Input Output
SSA	Side Slip Angle
TCP/IP	Transmission Control Protocol / Internet Protocol
UART	Universal Asynchronous Receiver-Transmitter
ULM	Ultra Light Machine
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
USB OTG	Universal Serial Bus On The Go
VFR	Visual Flight Rules
WKUP	WaKeUP



# 1 Introduzione

## 1.1 Storia dell'FTI

Nello scorso secolo l'aviazione ha avuto un'evoluzione impressionante. Dal primo successo dei fratelli Wright nel 1903, si è arrivati a mettere piede sulla luna nel 1969. 66 anni di evoluzione che evidenziano che lo sviluppo dell'aeronautica è stato qualcosa di incredibile.

Ma perché quello dei fratelli Wright è considerato il primo volo? In realtà già prima di loro alcuni avevano effettuato esperienze simili, ma la cosa che rende unico il lavoro dei fratelli Wright è l'aver tenuto traccia "scientifica" delle loro azioni e delle scelte tecniche effettuate. Nei loro diari troviamo accurata descrizione delle prove, dei dati e delle tecniche di volo effettuate. I primi voli sono stati in effetti le prime prove di volo. Secondo la Society of Flight Test Engineers (SFTE), "Le prove di volo sono un processo attraverso il quale si ottengono dati ed informazioni da un velivolo o da un sistema progettato per operare nell'atmosfera" [1].

Otto Lilienthal, considerato il costruttore del primo alante in grado di sollevarsi da terra trasportando un essere umano, sosteneva la necessità di raccogliere in modo sistematico le impressioni, i dati di volo e gli insegnamenti derivanti da ogni esperienza di volo al fine di tenere traccia dei difetti riscontrati, delle modifiche effettuate e dei risultati ottenuti durante lo sviluppo di un velivolo.

Inizialmente, come nel caso dei fratelli Wright, gli unici dati forniti da un'esperienza di Flight Test (FT) erano esclusivamente le sensazioni e le indicazioni che il Pilota era in grado di riportare al termine del volo.

Le prime vere e proprie registrazioni di dati di volo vennero ottenute con l'utilizzo di cineprese che riprendevano la strumentazione del pilota durante il volo.

Successivamente con l'avvento dell'elettronica, è stato possibile lo sviluppo di vere e proprie Flight Test Instrumentation (FTI) atte a monitorare, misurare e registrare i parametri di volo per la progettazione, lo sviluppo e la realizzazione dei velivoli.

In realtà lo sviluppo del mezzo aereo è stato fortemente incentivato dalle due guerre mondiali, ed in mezzo a loro l'epoca delle grandi imprese. Le motivazioni di questo sviluppo sono passate da "più veloce, più in alto, più lontano" alla tecnologia della sicurezza e della prestazione ecocompatibile di oggi.

La continua evoluzione dell'aeronautica e degli obiettivi perseguiti nello sviluppo del velivolo ha portato alla progettazione di specifici FTI per differenti scopi. In particolare l'FTI di cui tratta questa tesi è dedicato alla categoria dei velivoli Ultra Light Machines (ULM), che sono regolamentate dalle normative CS-VLA (Very Light Aircraft) e CS-LSA (Light Sport Aeroplane).

## 1.2 UltraLeggero a Motore (ULM)

L'acronimo ULM sta a indicare la categoria degli ultraleggeri motorizzati [2] e comprende quei velivoli destinati esclusivamente al volo da diporto con limiti di peso regolamentati inferiori a quelli facenti parte dell'aviazione generale.

Gli ultraleggeri motorizzati vengono generalmente suddivisi in:

- motoalianti;
- pendolari (comandati su due assi con lo spostamento del peso);
- paramotore (parapendio con imbrago, dotato di motore);
- tre assi (dotati di comandi sui tre assi spaziali: rollio, beccheggio ed imbardata);
- autogiri;
- elicotteri.

Sebbene gli ultraleggeri siano per la maggior parte di piccole dimensioni, semplici e lenti, alcuni possiedono il carrello retrattile, l'elica a passo variabile e possono raggiungere velocità intorno ai 350 km/h.

Con il termine di ultraleggero avanzato, si indica una categoria di ultraleggeri particolare avente caratteristiche tecniche conformi a standard tecnici almeno equivalenti a quelli riportati negli allegati tecnici II, III, IV e V del DECRETO DEL PRESIDENTE DELLA REPUBBLICA 9 luglio 2010, n. 133; devono inoltre essere dotati di radio aeronautica, transponder in modalità A/C ed E.L.T. (Emergency Locator Transmitter), e devono essere registrati presso l'AeCI come mezzo avanzato [3].

Rispetto ai normali ultraleggeri, gli ultraleggeri avanzati possono fruire dei servizi di assistenza alla navigazione aerea e non sono soggetti a restrizioni riguardanti la quota di volo ma hanno gli stessi obblighi degli altri aeromobili. Hanno solo l'ulteriore obbligo di rimanere fuori dagli spazi aerei controllati e dalle zone di traffico aeroportuale, a distanza di sicurezza dagli ostacoli e a distanza non inferiore a cinque chilometri dagli aeroporti. È evidente quindi che un ultraleggero avanzato, registrato come tale, può volare in tutto lo spazio aereo di classe G (spazio non controllato) con regole equivalenti a quelle dell'aviazione generale, relativamente al volo a vista (VFR).

Dall'introduzione della categoria "Ultraleggero avanzato" nel 2010, è nata quindi la necessità di dimostrare che un determinato velivolo con i limiti di peso relativi alla categoria ultraleggero, soddisfa le caratteristiche tecniche contenute nel DPR133.

Per poterlo dimostrare, i produttori di velivoli dovranno quindi eseguire una campagna di voli test per dimostrare che in fase di volo tutte le richieste contenute nel Decreto 133 siano soddisfatte.

Dal 2010 è quindi sorta la necessità di un FTI con particolari caratteristiche in fatto di peso che sia in grado di acquisire e registrare i parametri necessari alla dimostrazione che il velivolo in prova è certificabile come ULM Avanzato.

### **1.3 Flight Test Instrumentation (FTI)**

Con il termine FTI si indicano tutti quei sistemi di misurazione [4], di visualizzazione, di comunicazione e di registrazione che vengono utilizzati su velivoli sperimentali con lo scopo di monitorare il comportamento del velivolo.

Alcuni di questi parametri possono essere mostrati direttamente ai Test Pilot in fase di prova per avere un controllo più completo della manovra in corso. Una delle particolarità degli FTI di ultima generazione è quello di avere una Command Display Unit (CDU) configurabile in modo da poter scegliere i parametri visualizzati e aver la possibilità di tener d'occhio i giusti parametri durante ogni singola manovra Test.

Tutti i parametri misurati, sono anche costantemente registrati in modo da poter effettuare una più completa analisi delle varie manovre in fase di post processing.

Un sistema FTI tipicamente può monitorare da 10 a 120000 parametri di volo. Questi possono essere di svariato tipo:

- temperature di particolari componenti (motori, freni, ecc)
- distribuzione di pressione su superfici aerodinamiche
- stato di sforzo di particolari componenti critici
- tensioni e correnti che circolano in particolari dispositivi di bordo,
- informazioni che circolano sul databus proprio del velivolo
- immagini per verificare il movimento di particolari superfici visivamente

La scelta dei parametri registrati nelle varie prove può variare a seconda della prova stessa in quanto solitamente il costo di un FTI aumenta esponenzialmente con il numero di dati registrabili. Infatti nelle prime prove di volo si monitorano dati di sforzo di particolari componenti critici o informazioni riguardo l'aerodinamica del velivolo, successivamente quando la conoscenza di questi parametri è stata consolidata, si passerà al controllo di altri dati riguardanti per esempio le qualità di volo.

Talvolta le FTI prevedono un sistema di telemetria che permette di inviare informazioni a una Ground Station dove verranno fatte delle analisi preliminari in real-time con lo scopo di capire se una manovra test è andata a buon fine o se sarebbe meglio ripeterla risparmiando così tempo e denaro per lo sviluppo stesso del velivolo.

## 2 Mnemosine MK III

Mnemosine MK III è un FTI destinato alla categoria ULM avanzati progettato in collaborazione col Dipartimento di scienze e tecnologie Aerospaziali (ex DIA) del Politecnico di Milano [5]. Il suo sviluppo è stato molto influenzato dalla natura accademica del progetto.

La caratteristica principale del sistema FTI Mnemosine MK3, è quella di essere delocalizzato, infatti è composto da svariati nodi indipendenti provvisti di microcontrollore che misurano convertono e inviano i dati su un Can-Bus unico. A questo Can-Bus è collegato un Can-Recorder che è un dispositivo commerciale che registra tutte le informazioni che transitano sul Can-Bus.

Il protocollo di comunicazione tra i nodi è denominato CAFFE (CAAn For Flight test Equipment) ed è una evoluzione del protocollo Can aerospace (Controller Area Network aerospace) che a sua volta deriva dal Can ideato e sviluppato da Bosch. Il Can originario è un protocollo di comunicazione seriale creato per gestire sistemi real-time delocalizzati. Il progetto Can è nato in ambito automobilistico, ma come confermato dalla sua evoluzione, ha trovato notevole sviluppo anche in campo aeronautico.

I vari nodi che compongono Mnemosine MK III sono:

- Erato analizza i dati che circolano sul Can-Bus e li invia alla Ground Station;
- Eutherpe acquisisce mediante dei potenziometri la posizione dei comandi di volo e indirettamente delle superfici di controllo;
- Klios è il nodo dedito alla memorizzazione di tutti i dati che circolano sul Can-Bus;
- Melete ha il compito di alimentare tutto l'FTI;
- Polimnia è composto principalmente dal modulo GPS che oltre a fornire la posizione del velivolo, consente di fornire il segnale (Hearthbit) di sincronizzazione;
- Talia acquisisce i dati motore;
- Terpiscore è collegato alla piattaforma inerziale gestisce i dati di assetto del velivolo;
- Urania è il modulo montato in prossimità del Pitot-boom che gestisce i dati aria.

L'architettura di Mnemosine MK III è rappresentata nello schema in Figura 2.1.

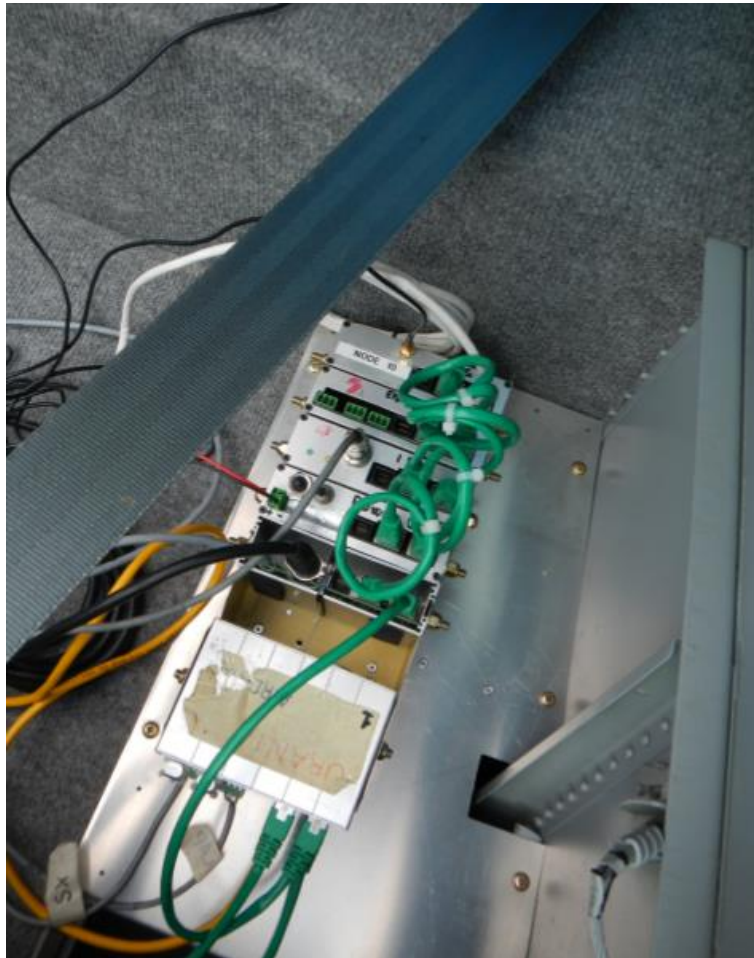




**Figura 2.1 - Architettura del sistema FTI Mnemosine MK3**

I principali pregi di questo tipo di architettura sono:

- semplicità dei singoli nodi;
- versatilità e semplicità di installazione del sistema;
- elevate possibilità di sviluppo tramite l'adozione di nuovi moduli progettati a doc;
- elevata affidabilità in quanto il malfunzionamento di un singolo nodo non compromette il funzionamento dell'intero FTI permettendo di acquisire comunque una parte dei parametri di volo.



**Figura 2.2 - Alcuni moduli di Mnemosine MK3 installati nella cockpit**

Nella Figura 2.2 possiamo contare sei moduli di Mnemosine MK III installati nella cockpit del velivolo XL di Nando Groppo. Si nota anche la presenza di un cavo verde che interconnette tutti i moduli, questo è il Can-Bus.

Questa immagine evidenzia come alla facilità di installare i singoli nodi vicino ai componenti dei quali si deve monitorare qualche parametro, si contrappone la presenza di un cavo per il Can-Bus che interconnette tutti i nodi e che spesso non è di semplice posizionamento. Un episodio chiarificatore dell'importanza di questo fattore è avvenuto durante una campagna di voli didattici. Durante le procedure di test, un Flight Test Engineer (FTE) salendo a bordo del velivolo calpestò accidentalmente il cavo del Can-Bus rompendone il connettore e rendendo impossibile l'acquisizione dei dati relativi alla posizione dei comandi di volo durante l'intera prova.

Un altro difetto che si è notato è che nonostante molti nodi siano montati spesso vicini tra loro essendo posizionati tutti nel cockpit, essi, per le caratteristiche originali del progetto, presentano un sistema di alimentazione proprio che in realtà potrebbe essere condiviso migliorando così ingombri, pesi e costi.

Ricapitolando, i principali svantaggi che gravano sull'architettura di Mnemosine MK III sono:

- Ridondanza di parti comuni ai vari nodi;
- Ingombri necessari ad ogni nodo e ai relativi cablaggi.
- Necessità di un nodo esclusivamente per la memorizzazione dei dati acquisiti.

Saranno proprio questi i punti da cui partirà la progettazione della nuova generazione.

## **2.1 Definizione dei requisiti per Mnemosine MK IV**

A seguito delle considerazioni fatte nel capitolo precedente, si è quindi deciso di stilare un nuovo pacchetto di requisiti da soddisfare in fase di progettazione del nuovo FTI che prenderà il nome di MNEMOSINE MK IV.

Il nuovo sistema FTI verrà quindi progettato con:

- un architettura centralizzata (per quanto possibile) composta da:
  - un nodo principale dotato di un “potente” microcontrollore che gestisca l'acquisizione di vari sensori e che garantisca la corretta successione e memorizzazione dei dati acquisiti;
  - un modulo dati aria dotato di processore in grado di fornire dei dati calcolati dai valori di pressione acquisiti dal pitot e corretti in funzione degli angoli AOA e SSA e di temperatura forniti dai dispositivi presenti sul Pitot-Boom.
  - Eventuali nuovi moduli implementabili per successivi miglioramenti.
- una doppia alimentazione separata per la parte analogica e digitale del circuito;
- una batteria di backup che permetta il salvataggio dei dati acquisiti anche in caso di arresto critico;
- un Can-Bus che permetterà di comunicare con nodi periferici, inizialmente il nodo dati aria;
- un ricevitore GPS onboard;
- tre canali seriali per poter connettere AHRS, CDU e Stickforce;
- otto ingressi analogici con relativi circuiti di adattamento per essere compatibili con l'ADC e essere disaccoppiati elettricamente;
- la predisposizione per implementare una trasmissione dati a un eventuale Ground Station;

La presenza del Can-Bus garantisce un'estrema versatilità al nostro FTI garantendo così ampie possibilità di sviluppo, particolarità che è sempre stato un punto di forza di Mnemosine.

Non verrà utilizzato il vecchio modulo Urania per l'acquisizione dei dati aria perché è previsto un nuovo nodo dotato di una notevole capacità di calcolo. Questo per avere la possibilità di effettuare una elaborazione dei dati in loco in maniera da correggere i dati di pressione totale e dinamica misurati dal Pitot-Boom utilizzando le misure di direzione del vento relativo e di temperatura dell'aria stessa.

In conclusione il sistema sarà composto da un modulo centrale che gestirà gran parte delle acquisizioni dall'interno della cockpit, e pochi nodi delocalizzati collegati tramite il Can-Bus.

Per come è stato concepito, il nodo centrale di Mnemosine MK IV sarà costituito dai seguenti componenti:

- la scheda del micro controllore dotata di slot per scheda microSD;
- l'alloggiamento della scheda del controllore e le relative interconnessioni;
- il modulo GPS;
- tre connettori per la comunicazione seriale con moduli esterni (es. AHRS);
- i componenti elettronici necessari per la gestione (Pwr etc.);
- l'elettronica di condizionamento di otto segnali per dare modo al microcontrollore di poter acquisire i segnali analogici in arrivo da sensori esterni al modulo (potenziometri, termocoppie, ecc).

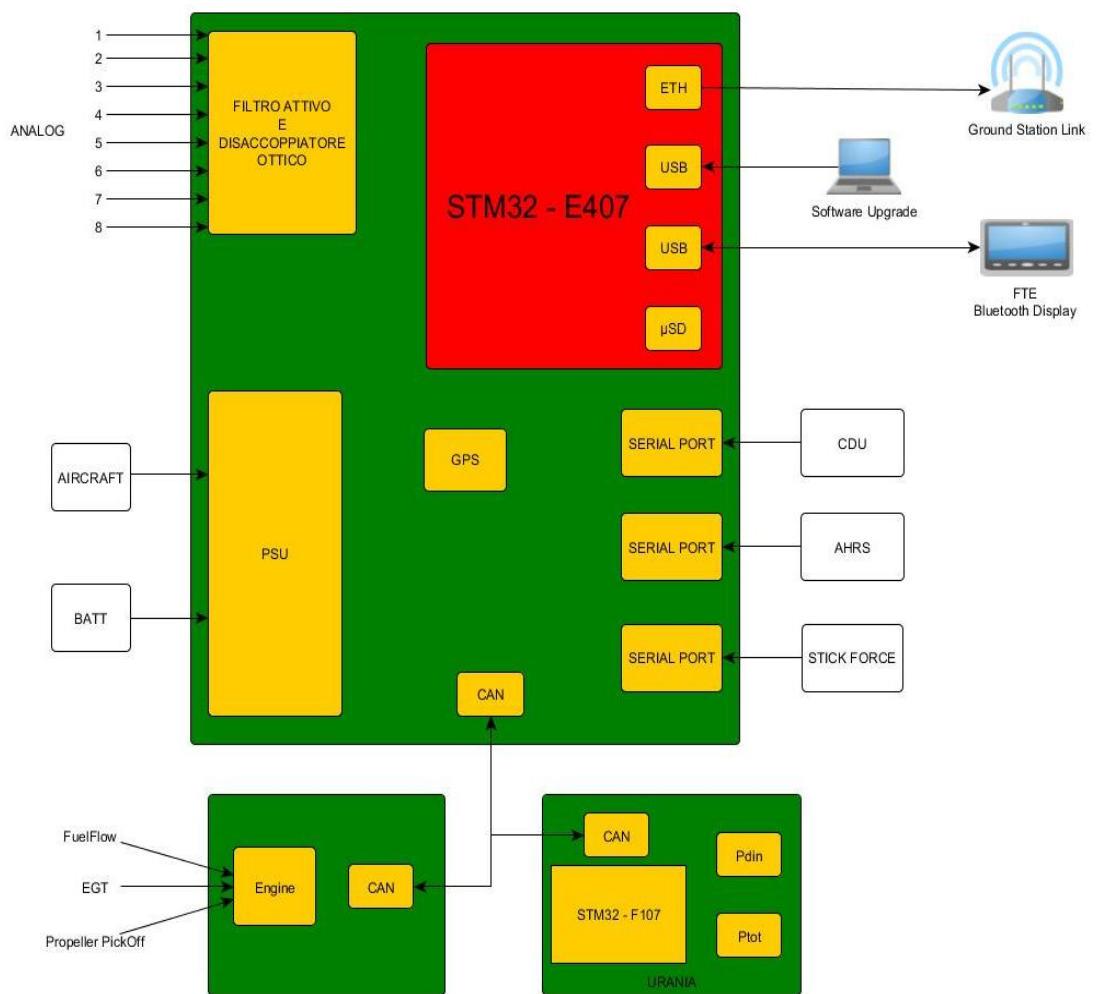
I moduli dislocati collegati tramite il Can-Bus che potranno anche non essere tutti presenti saranno invece:

- modulo dati aria;
- ingressi analogici aggiuntivi per diversi possibili scopi:
  - dati motore;
  - posizione comandi;
  - sforzi di barra;
- moduli di espansione futuri.

Per quanto riguarda i moduli periferici collegati al Can-Bus, essi saranno costituiti da 4 importanti componenti:

- eventuale circuito per la ricezione di segnali digitali
- eventuale circuito per il condizionamento e l'acquisizione di segnali analogici;
- un microcontrollore per l'elaborazione digitale dei dati;
- un interfaccia Can-Bus per connettere il modulo periferico all'FTI.

L'architettura finale di Mnemosine MK IV può essere quindi schematizzata come rappresentato in Figura 2.3.



**Figura 2.3 - Architettura di Mnemosine MK4**

## 3 Il modulo principale

In fase di progettazione si è deciso di realizzare il modulo madre di Mnemosine in modo che fosse facilmente posizionabile all'interno della cockpit del velivolo. Per ottenere questo risultato, è stato deciso di posizionare tutti i connettori e gli eventuali collegamenti su un solo lato del modulo principale dell'FTI. Questo permette il posizionamento in un angolo della cockpit in modo da non creare intralcio durante i test in volo e semplificare le operazioni di allestimento.

Per realizzare un device dalle dimensioni accettabili con tutti i componenti indicati nel capitolo precedente, l'elettronica è stata separata su due schede che chiameremo Masterboard e Slaveboard. Si pensa di montare le due schede con il lato componenti affacciato verso l'interno e di connetterle tra di loro con un flat cable 40 pin.

Sulla Masterboard collocheremo:

- l'elettronica dell'alimentazione con i relativi connettori;
- la scheda OLIMEX STM32-E407 con il processore ARM Cortex-M4F;
- il modulo GPS;
- l'elettronica per collegare 3 canali seriali e quindi i relativi driver;
- il connettore per connettere la slaveboard.

Sulla Slaveboard troverà posto invece:

- l'elettronica relativa agli ingressi analogici
- il circuito di pilotaggio del Can-Bus (Sperimentale)
- il connettore per la connessione con la masterboard.

### 3.1 Architettura e dimensioni delle schede

Per scegliere le dimensioni del dispositivo, il dimensionamento delle schede è stato fatto tenendo conto dello standard Eurocard.

Le misure dello standard Eurocard sono riportate in Tabella 3.1

Dimensioni Eurocard per circuiti stampati			
Formato	Sigla	Altezza	Larghezza
Eurocard	3U - Db160	100	160
Extended Eurocard	3U - Db220	100	220
Super Extended Eurocard	3U - Db250	100	250
Hyper Extended Eurocard	3U - Db280	100	280
Normal Double Eurocard	6U - Db160	233.4	160
Extended Double Eurocard	6U - Db220	233.4	220
Super Extended Double Eurocard	6U - Db250	233.4	250
Hyper Extended Double Eurocard	6U - Db280	233.4	280
Normal Triple Eurocard	9U - Db160	366.7	160
Extended Triple Eurocard	9U - Db220	366.7	220
Super Extended Triple Eurocard	9U - Db250	366.7	250
Hyper Extended Triple Eurocard	9U - Db280	366.7	280

Tabella 3.1 - Dimensioni standard Eurocard

La dimensione definitiva delle schede è stata scelta di 233.4mm x160mm Per garantire un adeguata lunghezza del lato connettori e non creare eccessivo ingombro.

Lo spazio sulle schede non occupato da componenti elettronici sarà configurato con una superficie millefori a passo 2.54mm per permettere l'eventuale futura realizzazione di circuiti per l'implementazione di nuove funzioni.

Nella Figura 3.1 si può vedere l'organizzazione interna del modulo madre con tutti i componenti presenti e la loro ubicazione sulle due schede.

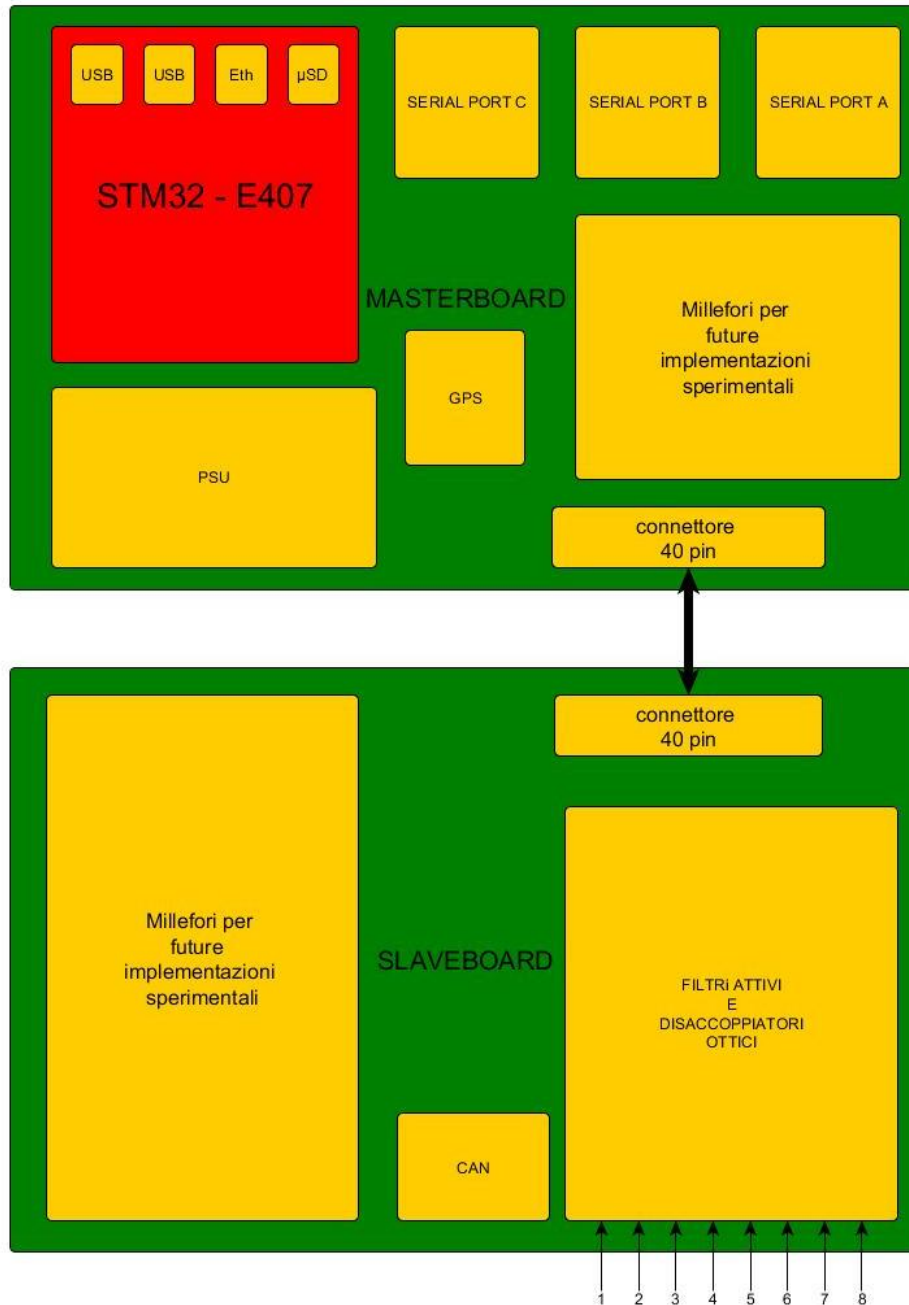


Figura 3.1 - Architettura interna del modulo madre di Mnemosine MK IV

## 3.2 Power Supply Unit

Nella Figura 3.2 si vede lo schematico riguardante l'elettronica necessaria ad alimentare l'intero FTI.

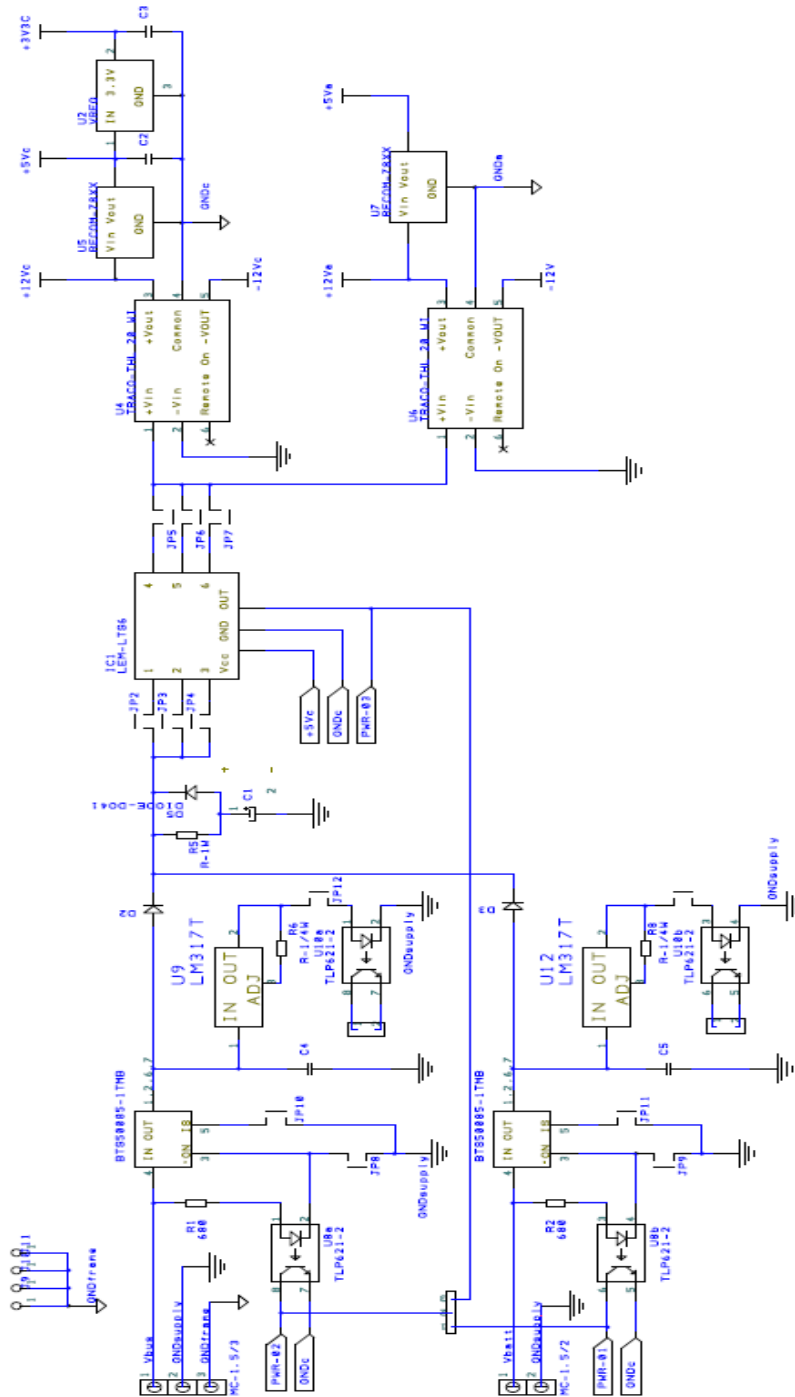


Figura 3.2 - Schematic Power Supply Unit



Per capire meglio come funziona questa unità, possiamo analizzare lo schema a blocchi nella Figura 3.3.

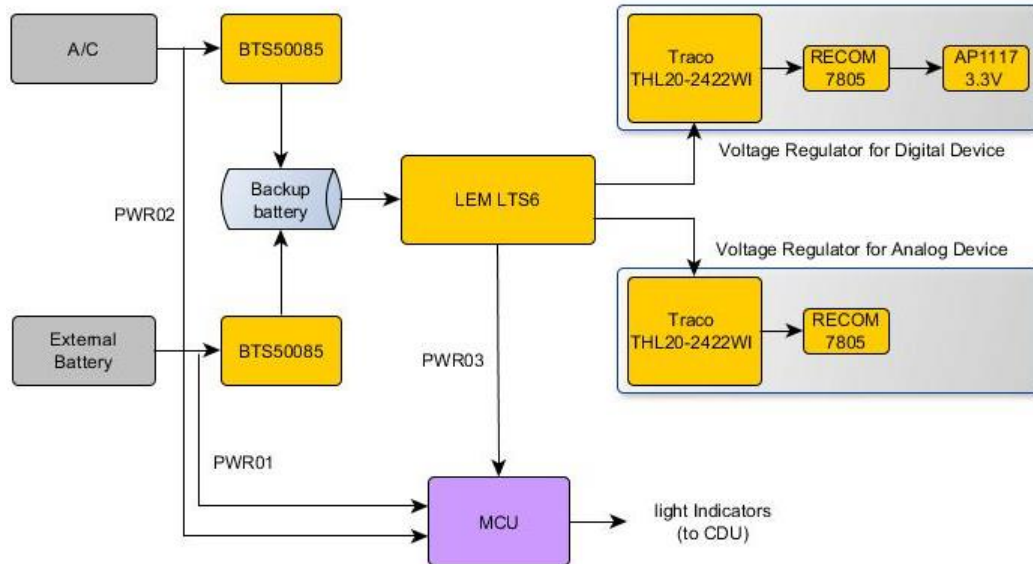


Figura 3.3 - Schema a blocchi della Power Supply Unit

Notiamo innanzitutto la presenza di due fonti di energia: A/C e External Battery, rispettivamente la presa di alimentazione dal velivolo e la presa di alimentazione da un eventuale batteria ausiliaria utile in fase di test post installazione dell'FTI.

Il circuito è stato concepito in modo da poter essere montato su svariati tipi di velivolo, infatti accetta come alimentazione tensioni che vanno da un minimo di 12 volt a un massimo di 38 volt sia per quanto riguarda la Vbus che la Vbatt anche se tipicamente quest'ultima sarà di 12 volt essendo un accumulatore di derivazione automobilistica.

Per permettere l'accensione e lo spegnimento di tutto il sistema, sono stati utilizzati due switch di potenza highside, U1 e U3 che essendo dei BTS50085 [6] hanno al loro interno protezioni per sovraccarico, per cortocircuito, per sovravoltaggio e per inversione delle polarità. Sono stati progettati per una varietà di impieghi sia industriali che automotive e quindi sono ideali per commutare qualunque tipo di carico resistivo, induttivo o capacitivo. PROtected Fet (PROFET) è il termine utilizzato per identificare la tecnologia che risalta le caratteristiche di robustezza che rendono questi dispositivi perfetti per essere impiegati in Mnemosine.

A valle di questi integrati ci sono due diodi schottky che impediscono un ritorno di corrente in caso di mancanza di una delle due alimentazioni.

È presente un circuito composto da un condensatore la cui corrente di carica passa attraverso una resistenza da 1.5Kohm da ½ Watt, mentre quella di scarica va ad alimentare direttamente tutto il circuito attraverso un diodo schottky come i precedenti. Questo semplice circuito ha il compito di intervenire in caso di mancanza di alimentazione e di alimentare tutto l'FTI per il tempo necessario ad andare in modalità protetta ed eseguire alcune procedure per salvare i file e spegnere i dispositivi correttamente.

Quando il sistema FTI è alimentato a 12 volt, il tempo per la ricarica completa del condensatore è di circa 13 secondi, carica che garantisce 20 millisecondi di alimentazione sufficienti al microcontrollore a rilevare la mancanza di alimentazione e, dopo essere andato in modalità recovery, a eseguire un procedura che garantisca l'integrità dei dati fino a quel momento registrati.

Per rilevare la presenza o meno delle due alimentazioni (Vbus e Vbatt) è presente un integrato (U8) che manda due segnali al microcontrollore per informarlo dell'eventualità che una delle due alimentazioni o entrambe, vengano meno. I due segnali sono PWR01 e PWR02.

L'integrato IC1 è un Lem-LTS6 [7] e funziona come trasduttore di corrente a effetto hall. L'uscita PWR03 di questo integrato è un segnale analogico che sarà dato in ingresso al microcontrollore che in questo modo avrà una stima del consumo istantaneo del sistema.

Sono presenti due connettori ( JP12 e JP13 ) per l'eventuale collegamento di due led che diano un indicazione visibile all'esterno riguardo la presenza o meno della relativa alimentazione.

A causa di disturbi presenti sulle linee di alimentazione dell'FTI si è deciso di implementare due differenti linee di regolatori di tensione, una che alimenta i device digitali e un'altra per i device analogici. Questo risolve il problema dei disturbi ma comporta il duplicamento dei regolatori di tensione necessari e un aumento quindi del costo del sistema.

Le due linee di regolatori di alimentazione sono praticamente identiche a meno di un regolatore 3.3V su quella digitale necessario per l'alimentazione del microcontrollore.

Le linee di regolatori di tensione sono poste a valle del trasduttore di corrente. Iniziano con due convertitori DC-DC (Traco THL20-2422WI [8] ) che accettano in ingresso tensioni fino a 50V e restituiscono in uscita una tensione duale di  $\pm 12V$ . Per avere i +5V sono stati utilizzati due circuiti 7805RS della Murata [9] che sono di tipo switching. Per alimentare il microcontrollore invece sono necessari i 3.3V e come detto prima, per ottenerli dalla linea di alimentazione digitale è stato utilizzato l'integrato AP1117-3.3V che è un regolatore lineare di tensione. In realtà è stata prevista anche la possibilità di montare un equivalente switching a 3.3V ma questo sarà una possibile futura modifica.

### 3.3 Scheda Olimex STM32-E407

La scheda Olimex STM32-E407 [10] è il cuore centrale del sistema FTI e si occupa per mezzo di un sistema Real-Time (che verrà esposto nella sezione software) di una serie di funzioni quali:

- l'acquisizione e la conversione di 8 segnali analogici che arrivano dalla slaveboard,
- la ricezione di 4 canali seriali,
- la gestione dei dati che transitano sul Can-Bus,
- il salvataggio dei dati sulla microSD,
- la gestione della porta Ethernet per comunicare con la Ground Station
- gestione di 2 porte USB per comunicare con eventuali periferiche come un uno schermo bluetooth da dare in dotazione all'FTE
- l'accensione/spengimento di led per comunicare determinati status del microcontrollore.

È basata su Core Cortex-M4 a 32bit con Reduced Instruction Set Code (RISC) che garantisce una grande capacità di memoria unita a una frequenza massima di funzionamento della CPU ad alta velocità, 168 Mhz (frequenza di clock), per applicazioni particolarmente complesse.

Il microcontrollore è un'evoluzione del Cortex M3 e infatti ha in aggiunta funzioni per l'elaborazione dei segnali digitali (Digital Signal Controller, DSC) eseguite in un singolo ciclo e istruzioni aritmetiche che sfruttano anche unità di calcolo in virgola mobile (Floating Point Unit, FPU) in modo da migliorare notevolmente le prestazioni. È inoltre presente una Memory Protection Unit (MPU) capace di migliorare la sicurezza del programma.

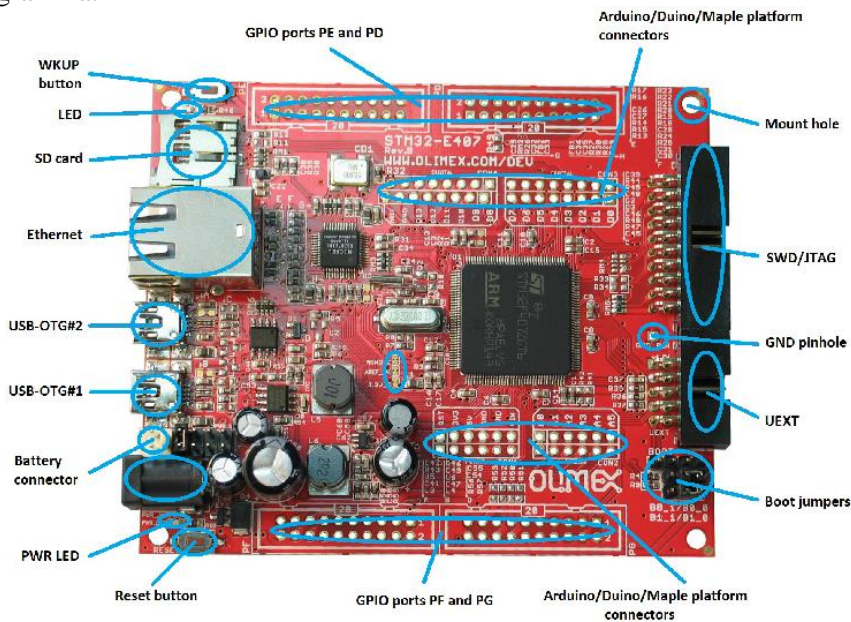


Figura 3.4 - Olimex STM32-E407 layout

La memoria del microcontrollore è composta da 1Mbyte di memoria flash e 192+4Kbyte di memoria SRAM inclusi i 64 Kbyte della Core Coupled Memory (CCM). Tutto questo garantisce un adeguato spazio di memoria per tutte le applicazioni per cui Mnemosine MK IV è stato progettato.

Con questo microcontrollore è possibile ottenere un massimo di tre convertitori analogico-digitali (ADC) a 12 bit a 2.4MSPS e due convertitori digitale-analogico (DAC) sempre a 12 bit.

È inoltre dotato di dodici timer a 16 bit e due timer a 32 bit ognuno configurabile per lavorare con acquisizioni, confronti output, e Pulse Width Modulation (PWM).

Sono presenti anche quindici differenti interfacce di comunicazione che includono la I<sup>2</sup>C, quattro USART, tre SPI, due interfacce CAN e una SDIO.

Queste ultime due sono molto importanti in quanto, il Can-Bus permette a Mnemosine di comunicare con gli altri nodi dell'FTI, e la SDIO permette il salvataggio diretto dei dati acquisiti su microSD senza l'ausilio di un modulo dedicato come era previsto in Mnemosine MK III.

La scheda Olimex è dotata di una Joint Test Action Group (JTAG) che è connettore Arm 2x10 pin per programmare e debuggare il software, presenta un connettore Ethernet che supporta una velocità fino a 100Mbit, ha due host USB On The Go (USB OTG), uno slot Secure Digital card (SD), e presenta onboard un circuito di alimentazione DC/DC. Per operare con la scheda, oltre al software il programmatore ha a disposizione due pulsanti: uno di Reset e uno di WaKeUP (WKUP) che rispettivamente hanno il compito di resettare il microcontrollore e di “risvegliarlo” dallo stato di standby.

Il microcontrollore è inoltre dotato di quattro slot da venti pin con un bus di memoria esterno per la ricezione/trasmisione di segnali elettrici con tutti i dispositivi con cui dovrà essere collegato.

### 3.4 Ingressi seriali

Nel modulo madre sono previste 4 porte di comunicazione, una interna per comunicare con il modulo GPS e 3 esterne per comunicare con device esterni tipo la piattaforma inerziale.

Per rendere il sistema compatibile con più protocolli di trasmissione, si è deciso di implementare un multimode serial interface che supporti diversi protocolli di comunicazione.

Lo modalità di comunicazione sarà selezionata via software mediante l'utilizzo di quattro differenti segnali:

- **232En** segnale attivo alto che abilita la comunicazione con protocollo Rs232;
- **485Tx** segnale attivo alto che abilita la comunicazione con protocollo Rs485 in trasmissione;
- **485Rx#** segnale attivo basso che abilita la comunicazione con protocollo Rs485 in ricezione;
- **422Rx#** segnale attivo basso che abilita la comunicazione con protocollo Rs422 in ricezione.

La tabella di seguito mostra come vanno settati i segnali di configurazione per abilitare i diversi protocolli.

modalità di comunicazione	Segnali di configurazione			
	232 En	485 Tx	485 Rx#	422 Rx#
	attivo alto	attivo alto	attivo basso	attivo basso
Porta spenta	off -> $\emptyset$	off -> $\emptyset$	off -> 1	off -> 1
Rs232	on -> 1	off -> $\emptyset$	off -> 1	off -> 1
Rs485 Tx	off -> $\emptyset$	on -> 1	off -> 1	off -> 1
Rs485 Rx	off -> $\emptyset$	off -> $\emptyset$	on -> $\emptyset$	off -> 1
Rs422 Rx	off -> $\emptyset$	off -> $\emptyset$	off -> 1	on -> $\emptyset$

Tabella 3.2 - Configurazione modalità di comunicazione

Verranno ora descritti gli standard di comunicazione supportati: RS232, RS485 e RS422

### 3.4.1 RS232 (EIA RS232)

L'RS232 [11] è lo standard EIA equivalente allo standard europeo CCITT V21/V24, è un'interfaccia a bassa velocità per lo scambio di informazione tra device digitali. Il metodo più semplice per descrivere un segnale EIA RS-232 è partire con un esempio.

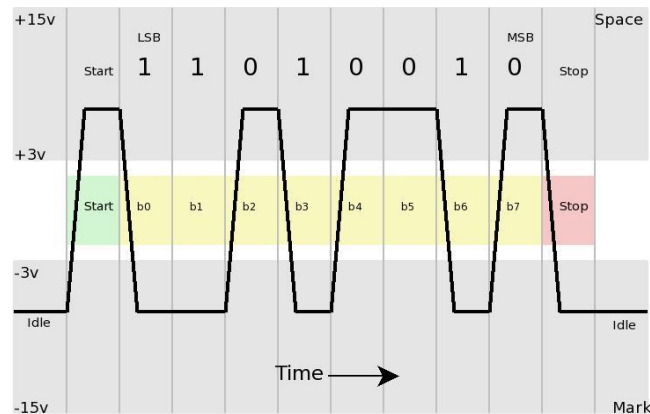


Figura 3.5 - Esempio del protocollo di trasmissione Rs232

Nella Figura 3.5 è visualizzato cosa appare collegando un oscilloscopio alla linea dati su cui transita un segnale EIA RS-232 a 9600 bps del tipo 8n1 (più avanti verrà spiegata questa sigla) corrispondente alla lettera "K" espressa in codice ASCII.

Tutte le transizioni appaiono in corrispondenza di multipli di 104  $\mu$ s (pari ad 1/9600 s, cioè ciascun bit dura esattamente l'inverso del bit rate).

La linea si trova inizialmente nello stato di riposo, a livello logico 1 (nessun dato in transito), quindi segnale di tensione negativo (p.e. -12V); la prima transizione da segnale negativo a positivo indica l'inizio della trasmissione (inizia il "bit di start", lungo esattamente 104  $\mu$ s), a cui segue il bit meno significativo (LSB), dopo altri 104  $\mu$ s, e così via, per otto volte, fino al bit più significativo (MSB). Da notare che il byte è trasmesso "al contrario", cioè va letto da destra verso sinistra. Segue infine un "bit di stop", cioè un periodo di riposo della linea di 104  $\mu$ s (ma può essere superiore qualora si convenga di usare un numero maggiore di bit di stop), e quindi, eventualmente, inizia un nuovo pacchetto di bit.

Un segnale seriale nello standard RS-232 può avere le seguenti varianti:

velocità di trasmissione, rispetto all'esempio di 9600 bps può essere più veloce o più lenta, cambiando in questo modo la distanza tra i fronti (p.e. a 1200 bps le transizioni avvengono a multipli di 1/1200 s, pari a 833  $\mu$ s)

numero di bit dati trasmessi, generalmente 8 bit (come nell'esempio), ma se ne possono trasmettere 5, 6, 7 o anche 9 (ma quest'ultima possibilità non è prevista dalle porte seriali dei normali PC)

possibilità di aggiungere un bit di parità alla fine del carattere

numero dei bit di stop, la linea rimane nello stato di riposo per almeno 1, 1,5 o 2 bit; ovviamente, nel caso in cui per un certo tempo non vi fosse più nulla da trasmettere, il "riposo" sarebbe molto più lungo.

A questo punto possiamo definire il significato della sigla 8n1 menzionata all'inizio dell'esempio. Tale sigla è usata in genere per definire il formato del pacchetto rice-trasmesso. Esempi possono essere 8n1 o 7e2:

il primo numero indica quanti bit di dati sono trasmessi (nei due esempi rispettivamente 8 e 7)

la lettera indica il tipo di parità (nei due esempi, rispettivamente, nessuna o even-parity, cioè parità pari)

il secondo numero indica il numero di bit di stop (rispettivamente 1 e 2)

Tenendo conto che esiste sempre un solo bit di start, una sequenza di bit trasmessa è quindi, per i due esempi citati, costituita rispettivamente da 10 (1+8+0+1) e 11 (1+7+1+2) bit. Da notare che di questi bit solo 8 e, rispettivamente, 7 sono effettivamente utili a trasportare l'informazione.

Lo standard originale prevede una velocità da 75 baud a 19200 baud. Uno standard successivo (RS-562) ha portato il limite a 64Kbps lasciando gli altri parametri elettrici praticamente invariati e rendendo quindi i due standard compatibili a bassa velocità. Nei normali PC le cosiddette interfacce seriali RS-232 arrivano in genere almeno a 115Kbps o anche più: pur essendo formalmente al di fuori di ogni standard ufficiale non si hanno particolari problemi di interconnessione.

Ovviamente sia trasmettitore che ricevitore devono accordarsi sul modo di trasmettere i dati prima di iniziare la trasmissione.

È importante garantire il rigoroso rispetto della durata dei singoli bit: infatti non è presente alcun segnale di clock comune a trasmettitore e ricevitore e l'unico elemento di sincronizzazione è dato dal fronte di discesa del bit di start. Come linea guida occorre considerare che il campionamento in ricezione è effettuato di norma al centro di ciascun bit: l'errore massimo ammesso è quindi, teoricamente, pari alla durata di mezzo bit (circa il 5% della frequenza di clock, considerando che anche il decimo bit deve essere correttamente sincronizzato). Naturalmente questo limite non tiene conto della possibile difficoltà di riconoscere con precisione il fronte del bit di start (soprattutto su grandi distanze ed in ambiente rumoroso) e della presenza di interferenze intersimboliche tra bit adiacenti: per questo spesso si consiglia caldamente di usare un clock con una precisione migliore dell'1% imponendo di fatto l'uso di oscillatori a quarzo. Si potrebbe anche ipotizzare un meccanismo che tenta di estrarre il clock dai fronti intermedi ma si tratta nel caso specifico di un lavoro poco utile, visto che la lunghezza del pacchetto è piuttosto breve.

### **3.4.2 RS485 (EIA RS485)**

L'EIA RS-485 [12], equivalente allo standard Europeo CCITT V11, è una specifica Modello OSI a livello fisico di una connessione seriale a due fili, half-duplex e multipunto. Lo standard specifica un sistema di gestione del segnale in forma differenziale: la differenza tra la tensione presente sui due fili costituisce il dato in transito. Una polarità indica un livello logico 1, quella inversa indica il livello logico 0. La differenza di potenziale deve essere di almeno 0,2 V per un'operazione valida, ma qualsiasi tensione compresa tra +12 V e -7 V permette il corretto funzionamento del ricevitore.

La EIA RS-485 specifica soltanto le caratteristiche elettriche del trasmettitore e del ricevitore. Non indica né raccomanda alcun protocollo per la trasmissione dei dati. EIA RS-485 permette la configurazione di reti locali a basso costo e comunicazioni multipunto. Permette una velocità di trasmissione molto elevata (35 Mbit/s fino a 10 m e 100 kbit/s a 1.200 m). Dal momento che utilizza un sistema di segnalazione con una tensione non trascurabile, con una linea bilanciata tramite l'impiego di un doppino (come avviene nella EIA RS-422), si possono raggiungere distanze relativamente notevoli (fino a poco più di 1.200 m).

Rispetto alla EIA RS-422, che ha un singolo circuito di pilotaggio, che non può essere spento, il trasmettitore per la EIA-485 viene messo in modalità di trasmissione in modo esplicito, applicando un segnale (CS).

EIA RS-485, come l'EIA RS-422 può essere resa full-duplex utilizzando quattro fili (due doppini), ma dal momento che EIA-485 è una specifica di tipo multipunto, ciò in molti casi non è necessario. EIA-485 e EIA-422 possono risultare compatibili entro certi limiti: in caso di connessioni multipunto in cui viene usato, ad esempio, un protocollo di tipo polling/selecting il trasmettitore master e il suo relativo ricevitore possono essere di tipo EIA RS422 mentre tutte le periferiche slave devono essere con un'interfaccia di tipo EIA RS-485.

EIA RS-485 è usata spesso con alcuni Universal Asynchronous Receiver/Transmitter (UART) piuttosto comuni per implementare dei sistemi di comunicazione dati nelle cabine degli aerei commerciali. È anche utilizzata con i controllori logici programmabili (PLC), ad esempio nelle aziende e nelle fabbriche per protocolli di comunicazione di tipo proprietario. Dal momento che è differenziale, resiste alle interferenze di natura elettromagnetica provenienti dai motori e dalle stazioni di saldatura.

### **3.4.3 RS422 (EIA RS422)**

L'EIA RS-422 [13] è uno standard EIA o V11 nella normativa europea CCITT. Si tratta di un protocollo per la comunicazione dati seriale che prevede l'impiego di due fili con linea differenziale e multi punto (differenziale bilanciata). Prevede, per ogni coppia di fili, la trasmissione di dati unidirezionale e non reversibile, su linee di trasmissione dati terminate o non terminate. Nel caso di due coppie di fili e, ovviamente con due circuiti simili, si ottiene la connessione full duplex.

A differenza dell'EIA RS-485, a cui differisce per la sola capacità di porsi sulla linea in alta impedenza se non selezionato, l'EIA-422 non permette trasmettitori multipli ma solo ricevitori multipli.

A differenza dello standard EIA RS-232 è stato progettato per connettere direttamente due apparecchiature (siano esse DTE o DCE) con alta immunità ai disturbi anche a distanze considerevoli (tipicamente fino a 1550 m) e a velocità considerevoli (anche oltre i 20000 bit/secondo). È uno standard molto diffuso, soprattutto in ambito industriale.

Poiché la variazione di stato del dato è determinato dalla differenza delle tensioni sui due fili in modo bilanciato (da 0 a +5 V e -5 V sui due conduttori rispettivamente) e dal momento che entrambi i fili arrotolati fra di loro con un passo ben preciso devono seguire lo stesso percorso, un rumore elettrico o disturbo, ripercuotendosi su entrambi i



conduttori, non altera la tensione relativa fra questi ottenendo così una alta immunità ai disturbi.

### 3.4.4 Circuito per le porte seriali

Il canale seriale interno dedicato al GPS sfrutta il protocollo proprio del ricevitore che attualmente è un U-Blox Lea4t [14] ma che potrà essere sostituito con uno più performante e di cui parleremo in seguito.



Figura 3.6 - Ricevitore GPS U-Blox Lea4T in tecnologia SMD

Il ricevitore GPS della U-Blox comunica col microcontrollore tramite il protocollo. Per dar luogo alla comunicazione col microcontrollore è necessario connettere i due componenti con due soli pin:

- **Tx Data** è il pin con cui il GPS trasmette i dati al microcontrollore;
- **Rx Data** è il pin con cui il GPS riceve i dati dal microcontrollore;

In realtà in Mnemosine MK IV esiste un terzo pin connesso. Questo è il **Time Pulse** che è generato dal GPS e è condiviso anche a tutti gli altri nodi grazie all'interfaccia Can-Bus e permette di allineare temporalmente tutta l'elettronica dell'FTI.

La porta di comunicazione del microcontrollore dedicata al ricevitore GPS è la Usart1. Nella Tabella 3.3 sono definiti i pin del microcontrollore dedicati alla comunicazione col ricevitore GPS.

Pin del ricevitore GPS	Segnale	Pin del microcontrollore
Pin3 - Tx	GPS data	PB07 – D0- Usart1 Rx
Pin4 - Rx	GPS data	PB06- D1 - Usart1 Tx
Pin28 - Time Pulse	GPS interrupt	PD07

Tabella 3.3 - Collegamenti ricevitore GPS

Per quanto riguarda invece le porte seriali esterne, uno schema per capire come funziona intuitivamente è il seguente:

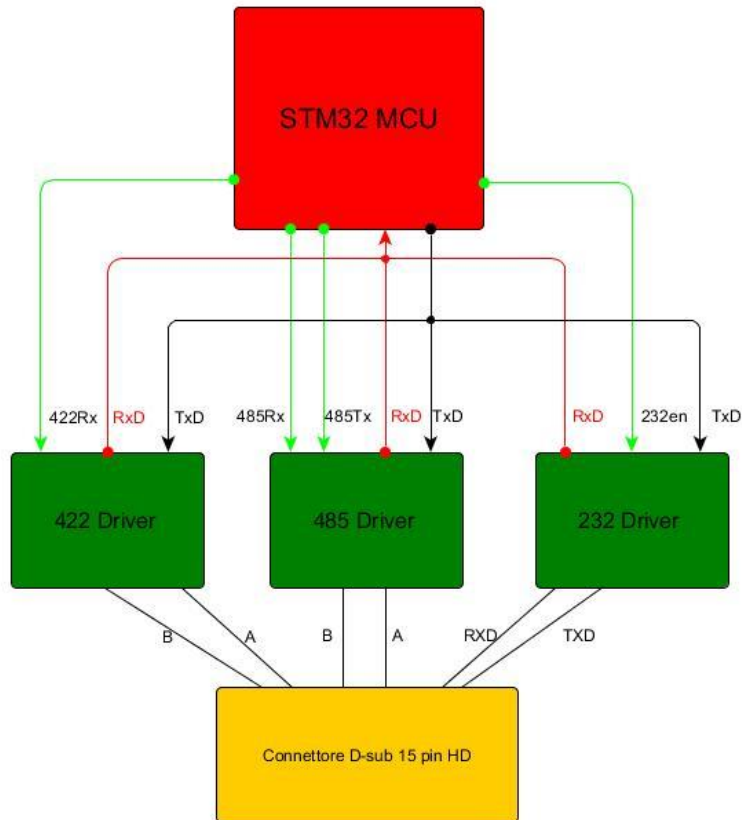


Figura 3.7 - Schema concettuale per il multimode serial port

In pratica i canali Data TX e Data RX saranno comuni a tutti i driver. Via software potremo decidere quale dei driver deve lavorare scegliendo così la modalità di comunicazione abilitando e disabilitando uno dei quattro segnali di configurazione:

- **232en** abilita la porta seriale a comunicare con protocollo 232
- **485rx** abilita la porta seriale a comunicare con protocollo 485 in ricezione
- **485tx** abilita la porta seriale a comunicare con protocollo 485 in trasmissione
- **422rx** abilita la porta seriale a comunicare con protocollo 422 in ricezione

Questi segnali saranno presenti per tutte e tre le porte seriali esterne che chiameremo Port A, Port B e Port C.

I dati contenuti nelle Tabella 3.4, Tabella 3.5 e Tabella 3.6 mostrano quali pin del microcontrollore sono associati ai vari segnali di controllo e di comunicazione delle porte seriali.

Porta seriale	Tipo di segnali	Segnale	Pin	I/O/Conf.
<b>Port A</b>	<b>USART 2</b>	Tx data	PD05	Output
		Rx data	PD06	Input
		Cts	PD03	Input
		Rts	PD04	Output
	Segnali ausiliari che definiscono il protocollo	422 Rx	PE13	Conf. (attivo basso)
		485 Rx	PE15	Conf. (attivo basso)
		485 Tx	PE14	Conf. (attivo alto)
		232 En	PE01	Conf. (attivo alto)
	led per indicazioni visive	Led verde	PE10	Output ottico
		Led giallo	PE11	Output ottico
		Led rosso	PE12	Output ottico

Tabella 3.4 - Pin del microcontrollore dedicati alla multimode serial interface Port A

Porta seriale	Tipo di segnali	Segnale	Pin	I/O/Conf.
<b>Port B</b>	<b>USART 3</b>	Tx data	PD08	Output
		Rx data	PD09	Input
		Cts	PD11	Input
		Rts	PD12	Output
	Segnali ausiliari che definiscono il protocollo	422 Rx	PG00	Conf. (attivo basso)
		485 Rx	PG02	Conf. (attivo basso)
		485 Tx	PG01	Conf. (attivo alto)
		232 En	PG03	Conf. (attivo alto)
	led per indicazioni visive	Led verde	PG04	Output ottico
		Led giallo	PG05	Output ottico
		Led rosso	PG06	Output ottico

Tabella 3.5 - Pin del microcontrollore dedicati alla multimode serial interface Port B

Porta seriale	Tipo di segnali	Segnale	Pin	I/O/Conf.
<b>Port C</b>	<b>USART 6</b>	Tx data	PC6 (Uext)	Output
		Rx data	PC7 (Uext)	Input
		Cts	PG15	Input
		Rts	PG12	Output
	Segnali ausiliari che definiscono il protocollo	422 Rx	PE02	Conf. (attivo basso)
		485 Rx	PE04	Conf. (attivo basso)
		485 Tx	PE03	Conf. (attivo alto)
		232 En	PE05	Conf. (attivo alto)
	Led per indicazioni visive	Led verde	PE08	Output ottico
		Led giallo	PE07	Output ottico
		Led rosso	PE06	Output ottico

Tabella 3.6 - Pin del microcontrollore dedicati alla multimode serial interface Port C

Le porte seriali esterne saranno disponibili su dei connettori D-sub HD a 15 pin con il seguente pin-out:

Pin	Segnale
1	+12 Vc
2	+5 Vc
3	Not connected
4	Not connected
5	GNDc
6	485 A - 422Tx A
7	422Rx A
8	Not connected
9	485 B - 422Tx B
10	422Rx B
11	232Tx
12	232Rx
13	232Cts
14	232Rts
15	Not connected
Isolamento	Gnd Frame

Tabella 3.7 - Pinout delle porte seriali

Lo schema del circuito interno al modulo madre utilizzato per interfacciare il microcontrollore al connettore della porta seriale sarà il seguente:

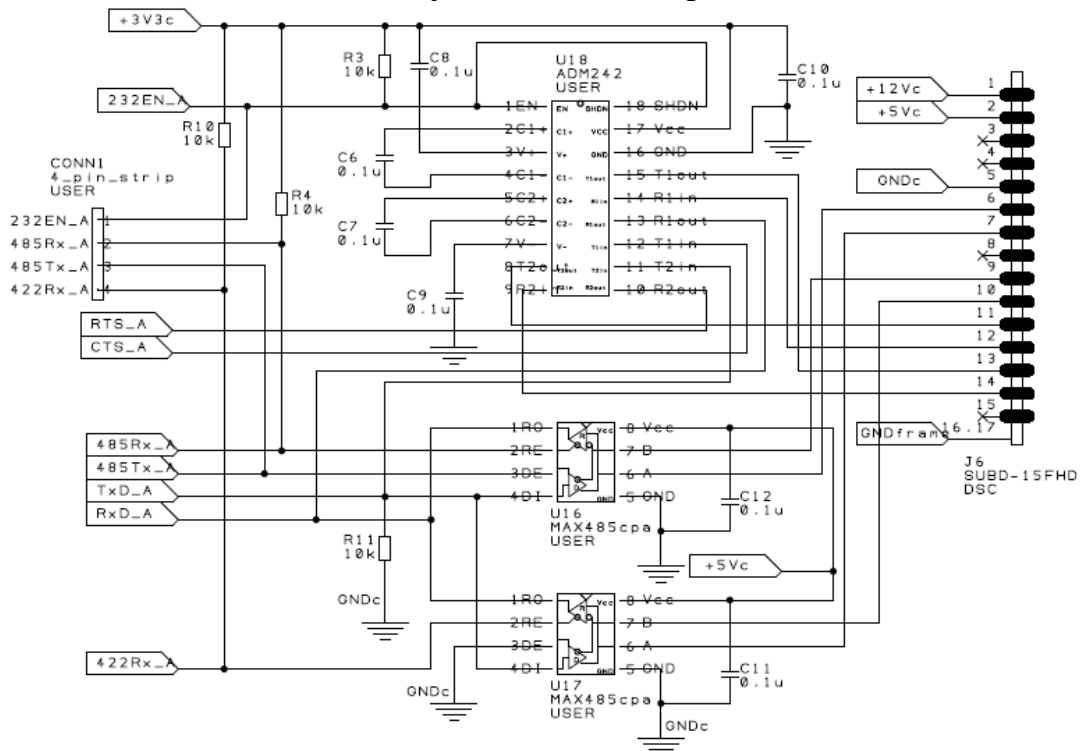


Figura 3.8 - Schematico multimode serial port A

## 3.5 Ingressi analogici

Il prototipo di Mnemosine MK IV prevede la possibilità di acquisire 8 canali analogici. Questa caratteristica potrà essere in futuro evoluta attraverso due vie: l'implementazione di ulteriori ingressi analogici onboard (sempre sul modulo madre) o attraverso l'adozione di un modulo aggiuntivo che comunica con l'FTI tramite il Can-Bus. Per ognuno di questi canali sono previsti:

- una porta di comunicazione;
- un circuito di condizionamento del segnale composto da
  - un filtro attivo;
  - un circuito di amplificazione;
  - un disaccoppiatore ottico;
- un pin del microcontrollore dedicato.

### 3.5.1 Porta di comunicazione per gli ingressi analogici

La porta di comunicazione degli ingressi analogici è fisicamente costituita da un connettore al quale possono essere collegati i sensori di misura. Per acquisire una variabile fisica, questa dovrà essere convertita dalla sua natura fisica, a una corrispondente misura elettrica acquisibile dal microcontrollore. Questa conversione viene effettuata dal sensore. Nel nostro caso i sensori utilizzati saranno potenziometri o celle di carico. Questo tipo di sensori è passivo e necessita quindi di una alimentazione. Per questo ad ogni canale analogico acquisito sono dedicati 4 pin:

- Ground frame: massa del telaio del velivolo;
- Ground: polo negativo dell'alimentazione;
- +5V: polo positivo dell'alimentazione
- Signal-in: segnale analogico da acquisire compreso tra 0 e 5 Volt

Per soddisfare questa necessità senza incorrere in un eccessivo ingombro del layout del frontale, sono stati previsti quattro connettori a otto pin ognuno dei quali trasporta due canali analogici. Il layout sul frontale sarà il seguente:

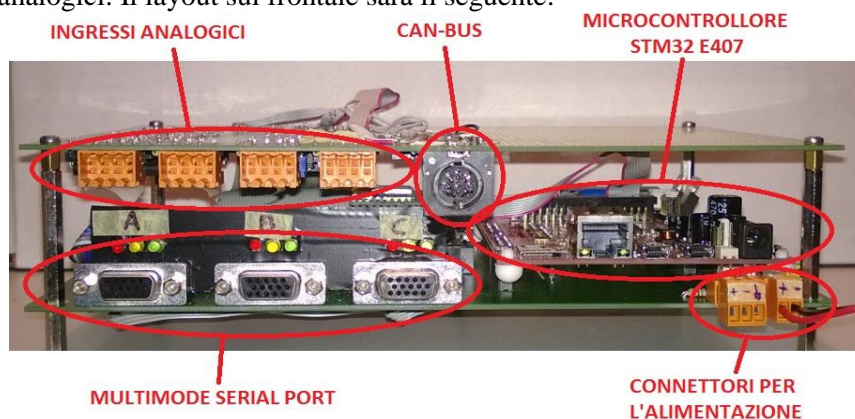


Figura 3.9 - Layout del frontale

### 3.5.2 Il circuito di condizionamento del segnale

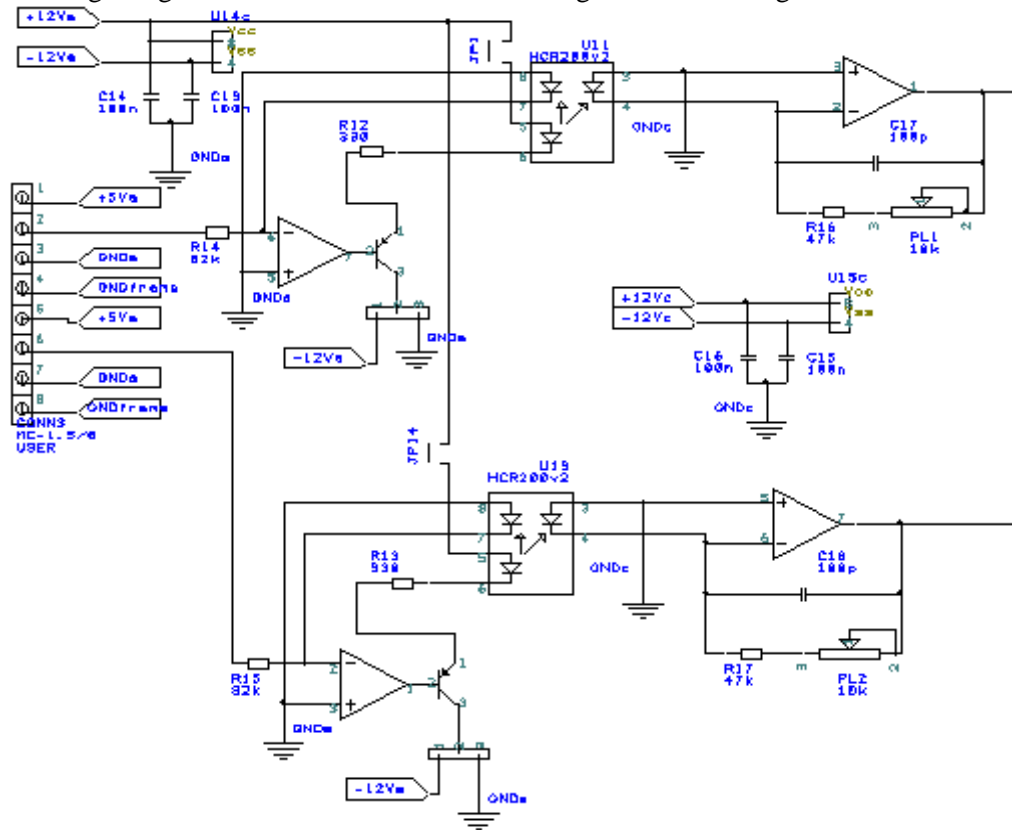
Il circuito di condizionamento del segnale è costituito da tre elementi: un disaccoppiatore ottico, un filtro e un amplificatore.

Questi 3 elementi in fase di progettazione sono stati integrati in un unico circuito a operazionali.

I segnali in ingresso hanno un range tra 0 e 5 Volt, mentre il microcontrollore accetta nell'ADC tensioni tra 0 e 3,3 volt.

Il filtro serve per eliminare possibili disturbi presenti al di fuori delle frequenze di interesse. La frequenza di taglio del filtro implementato è stata scelta di 150Hz. In realtà regolando il trimmer questa varia fino a un massimo di 178Hz.

Si è quindi progettato un filtro attivo che è un circuito a operazionali in grado di effettuare entrambe queste operazioni. È dotato di un trimmer 20 giri per la regolazione fine del guadagno e arrivare a sfruttare tutto il range accettato in ingresso dell'ADC.



### 3.5.3 Disaccoppiatore ottico

Una importante caratteristica di Mnemosine MK IV, è quella di avere tutti i segnali analogici disaccoppiati otticamente. Questo garantisce una protezione a tutto l'impianto ma soprattutto al microcontrollore che in caso di malfunzionamento non subirà danni.

L'unico componente esposto a rischi sarà l'HCR 201 v2 [15] che però avendo un costo esiguo non comporta particolari problemi.

La criticità di progettare un disaccoppiatore ottico per un segnale analogico consiste nella non linearità intrinseca dei componenti optoelettronici. L'integrato hcr200 v2 è stato progettato anche per essere utilizzato in un circuito che possa risolvere questa problematica.

L'integrato hcr200 v2 è dotato di un illuminatore e due fototransistor. L'illuminatore è alimentato dall'uscita di un operazionale, mentre i due foto transistor sono utilizzati uno per generare il segnale da mandare al microcontrollore mentre l'altro per realizzare la retroazione che linearizza l'uscita.

La verifica della linearità è avvenuta collegando due voltmetri per misurare rispettivamente la tensione in ingresso al canale analogico e la tensione in ingresso al microcontrollore.

Sono stati poi acquisiti gli estremi di taratura raggiungibili portando a fondo scala positivo e negativo i trimmer di taratura.

Fatto ciò si è regolato il trimmer di ogni canale in modo da ottenere una tensione di 3.3Volt al microcontrollore quando i pin di signal in e +5V dei singoli canali sono cortocircuitati.

Ovviamente questo lavoro è stato effettuato su tutti gli otto canali.

# porta	RANGE (5v)			TARATURA	
	Vin (Volt)	Vout <sub>min</sub> (Volt)	Vout <sub>max</sub> (Volt)	Vin (Volt)	Vadc (Volt)
1	4.9995	2.8061	3.4130	4.9995	3.3000
2	4.9992	2.8475	3.4526	4.9992	3.3000
3	4.9992	2.7869	3.3513	4.9992	3.3000
4	4.9994	2.7715	3.3666	4.9994	3.3000
5	4.9992	2.7722	3.3642	4.9992	3.3000
6	4.9992	2.7679	3.3596	4.9992	3.3000
7	5.0002	2.8436	3.4359	5.0002	3.3000
8	5.0003	3.3507	2.7874	5.0003	3.3000

Tabella 3.8 - Taratura ingressi analogici

Una volta tarati i canali, sono state effettuate le misure necessarie per verificare che il legame tra ingressi e uscite fosse lineare.

Le tensioni di prova sono state scelte tra 0 e 5 volt con step di 0.5 Volt.

Le tensioni che ci aspetteremmo di misurare in uscita vanno da 0 a 3.3 Volt linearmente con l'ingresso e sono quelle riportate nella Tabella 3.9.

	Tensioni di prova (Volt)										
	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
Tensioni previste	0	0.33	0.66	0.99	1.32	1.65	1.98	2.31	2.64	2.97	3.3

Tabella 3.9 - Tensioni previste ipotizzando la linearità del circuito.

La Tabella 3.10 riporta le tensioni misurate in ingresso al microcontrollore per le relative tensioni di prova.

# porta	Tensioni di prova (Volt)										
	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
1	-0.002	0.327	0.657	0.988	1.317	1.648	1.978	2.309	2.639	2.969	3.299
2	-0.003	0.327	0.660	0.988	1.318	1.648	1.978	2.309	2.639	2.969	3.300
3	-0.001	0.329	0.659	0.989	1.318	1.650	1.979	2.309	2.640	2.969	3.299
4	0.000	0.330	0.660	0.990	1.319	1.649	1.979	2.309	2.639	2.969	3.299
5	0.000	0.329	0.659	0.988	1.319	1.648	1.978	2.308	2.638	2.968	3.298
6	-0.001	0.330	0.660	0.990	1.320	1.650	1.980	2.310	2.640	2.970	3.300
7	-0.001	0.329	0.660	0.989	1.319	1.649	1.978	2.308	2.639	2.968	3.299
8	-0.001	0.330	0.660	0.990	1.319	1.650	1.980	2.310	2.640	2.969	3.299

**Tabella 3.10 - Test di linearità degli ingressi analogici**

In Tabella 3.11 possiamo valutare l'entità degli errori tra i dati misurati e i dati che ci aspetteremmo dalle rette di calibrazione. Questi si discostano dalle rette di calibrazioni con un errore massimo dello 0.067% calcolato in percentuale di fondoscala.

# porta	Errori percentuali di linearità										
	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
1	-0.043%	-0.052%	-0.052%	-0.050%	-0.060%	-0.040%	-0.040%	-0.020%	-0.020%	-0.020%	-0.020%
2	-0.067%	-0.064%	-0.002%	-0.040%	-0.040%	-0.040%	-0.040%	-0.020%	-0.020%	-0.020%	0.000%
3	-0.013%	-0.024%	-0.022%	-0.026%	-0.040%	0.002%	-0.020%	-0.020%	0.002%	-0.020%	-0.020%
4	0.000%	-0.004%	-0.008%	-0.004%	-0.020%	-0.020%	-0.020%	-0.020%	-0.016%	-0.016%	-0.018%
5	-0.008%	-0.016%	-0.020%	-0.032%	-0.020%	-0.040%	-0.040%	-0.040%	-0.040%	-0.040%	-0.040%
6	-0.018%	-0.005%	-0.002%	0.004%	0.008%	0.006%	0.000%	0.002%	0.006%	0.000%	0.002%
7	-0.023%	-0.022%	-0.008%	-0.022%	-0.020%	-0.020%	-0.040%	-0.040%	-0.020%	-0.040%	-0.020%
8	-0.012%	-0.008%	-0.008%	-0.009%	-0.018%	-0.008%	-0.006%	-0.006%	-0.010%	-0.014%	-0.016%

**Tabella 3.11 - Errori di linearità dei canali analogici**

Per verificare la linearità dei circuiti di condizionamento, dobbiamo però verificare che l'andamento degli errori non abbia tendenze lineari o quadratiche o di grado superiore. Per fare ciò costruiamo il grafico con in ascissa le tensioni in ingresso, e in ordinata il valore dell'errore rilevato espresso in percentuale del fondoscala.



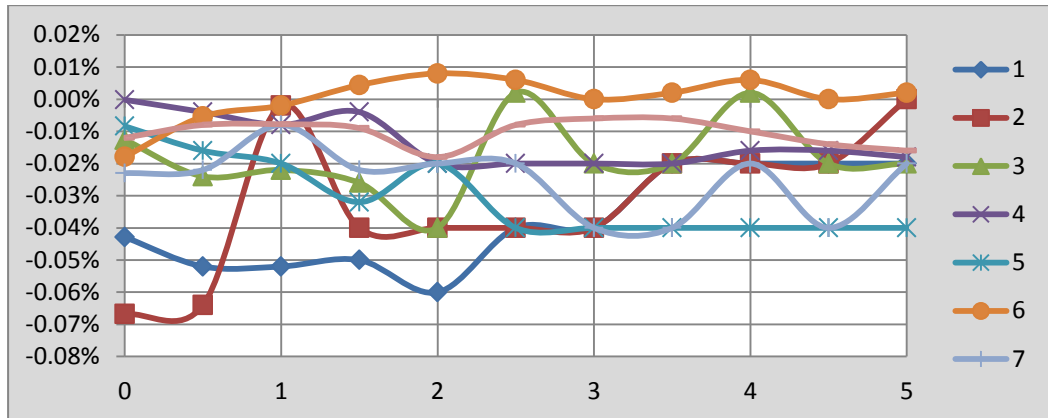


Figura 3.10 - Andamento degli errori dei canali analogici

Come possiamo vedere dal grafico in Figura 3.10 non esiste una tendenza lineare o quadratica degli errori, quindi possiamo dire che il circuito è un disaccoppiatore ottico lineare.

I segnali seriali così condizionati saranno collegati ai relativi pin del microcontrollore che li acquisirà grazie al convertitore ADC al suo interno.

I pin del microcontrollore dedicati ai segnali analogici sono i pin dal 6 al 13 della porta PF.

### 3.6 Interfaccia Can-Bus

Come descriveremo in seguito, questo circuito è stato implementato successivamente alla realizzazione delle schede PCB. Infatti è stato realizzato manualmente in via sperimentale sulla parte millefori della Slaveboard.

Come già spiegato in precedenza, il modulo madre comunica con alcuni altri moduli attraverso un Can-Bus.

Lo standard Can-Bus [16] è stato introdotto dalla Robert Bosch GmbH nel 1986 per interconnettere diverse Unità Elettroniche di Controllo (ECU). Per come è stato progettato il Can è un efficace mezzo di comunicazione in ambienti fortemente disturbati dalla presenza di onde elettromagnetiche. Il collegamento elettrico è simile a quello previsto per la RS485 infatti utilizza una linea a differenza di potenziale bilanciata. Con un cavo di tipo Twisted pair, l'immunità ai disturbi aumenta ulteriormente con costi ragionevoli.

Una rete di tipo Can è costituita da un unico cablaggio che interconnette uno svariato numero di moduli, teoricamente infiniti. Nella versione originale, la comunicazione avviene con tempo non deterministico, ma per le nostre necessità, lo standard è stato evoluto e presenta un segnale di sincronia che permette la ricostruzione temporale della sequenza di tutti i messaggi inviati dai vari nodi.

In una rete di tipo Can, tutti i device sono sempre pronti a ricevere le informazioni che transitano sul bus che prendono il nome di messaggi multicast. Per quanto riguarda invece la trasmissione di informazioni, ogni modulo è in grado di comprendere quando il Can non è impegnato da un altro device in modo da sapere di poter iniziare la trasmissione del proprio pacchetto di informazioni. Nel caso due periferiche iniziassero

la trasmissione nello stesso istante, i device riconoscerebbero un conflitto e la priorità sarebbe stabilita secondo specifiche definite Collision Detection.

Per realizzare questo tipo di comunicazione sono necessari diversi segnali elettrici. Per collegare il nostro sistema FTI al Can-Bus, è stato scelto un connettore a 8 contatti con il pin-out definito in tabella.

Pin	Segnale
1	GNDc
2	Clk B
3	GNDc
4	CAN H
5	Clk A
6	+12 Vc
7	-12 Vc
8	CAN L

Tabella 3.12 - Pinout del connettore del Can-Bus

I segnali CanH e CanL sono i segnali previsti dallo standard Can-Bus, +12, -12 e le due ground sono le alimentazioni e le tensioni di riferimento, mentre i segnali A e B sono segnali necessari per allineare temporalmente i moduli dislocati dell'FTI.

Come driver dei segnali CanH e CanL è stato utilizzato l'integrato MCP 2562, mentre per i segnali A e B è stato usato un comune Max485. In figura possiamo vedere lo schematico dei collegamenti che sarà praticamente identico su ogni modulo di Mnemosine MK4.

Nelle Figura 3.11 e Figura 3.12 vediamo lo schema elettrico dello stampato e la sua realizzazione sulla parte millefori della Slaveboard.

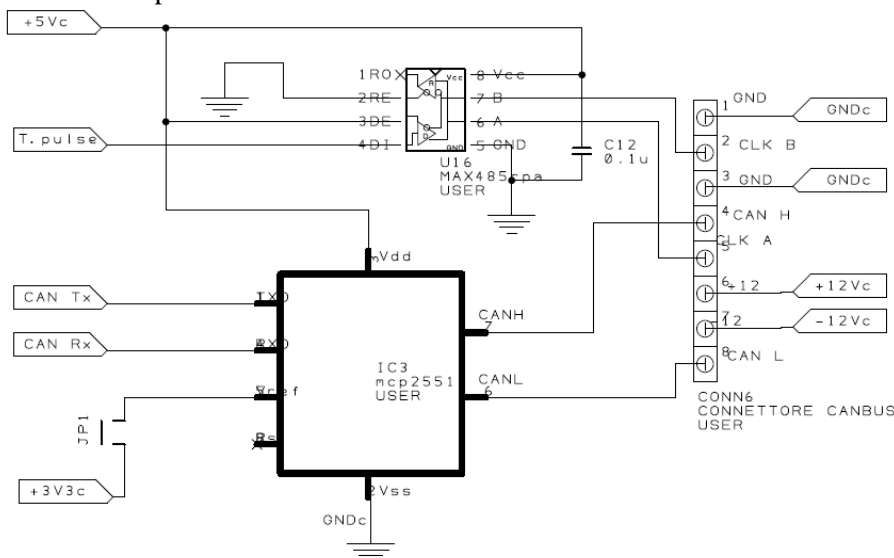


Figura 3.11 - Schematico Driver Can-Bus

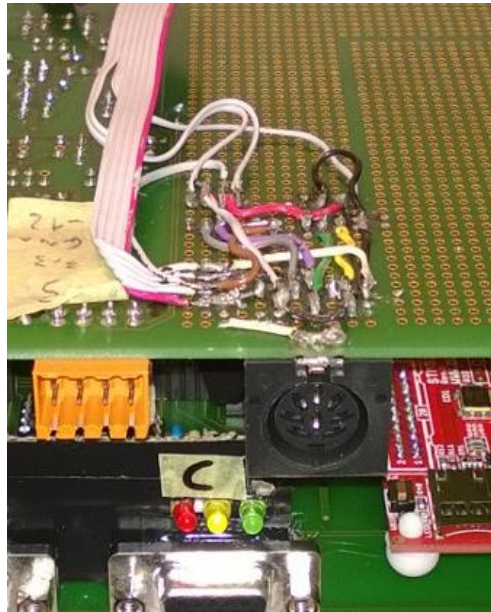


Figura 3.12 - Realizzazione sperimentale del circuito Driver Can-Bus

### 3.7 Modifiche e correzioni

Come succede durante lo sviluppo di ogni device in fase prototipale, anche durante la progettazione/realizzazione di Mnemosine MK4 ci si è trovati di fronte a bug e a variazioni di progetto che saranno riportate in questo capitolo.

#### 3.7.1 Implementazione del circuito Driver per il Can-Bus

Come già detto in precedenza il modulo Can-Bus implementato in questo prototipo, non era presente sui PCB prodotti all'inizio di luglio. Questo era solo un device presente sulla carta che è stato successivamente realizzato sperimentalmente. L'unico motivo per cui il connettore del Can-Bus è stato posizionato sulla slaveboard, è stato esclusivamente una questione di ingombro sul Pannello frontale.

Inizialmente lo schema concettuale dei segnali per il Can-Bus doveva essere quello riportato in Figura 3.13.

Si nota la presenza di 2 connettori Can-Bus, e 4 segnali provenienti dal Microcontrollore.

I 4 segnali che interessano il circuito Driver Can-Bus sono:

- **Can Tx Data:** segnale elettrico che porta le informazioni che il modulo madre vuole trasmettere sul Can-Bus
- **Can Rx Data:** segnale elettrico che porta le informazioni trasmesse dalle altre periferiche connesse al Can-Bus, dal circuito driver al microcontrollore STM32-E407

- **DI**: segnale TimePulse proveniente dal GPS e inviato a tutti i device collegati al modulo madre.
- **RO**: segnale TimePulse di ritorno dal driver 485 del Can-Bus affetto dal ritardo intrinseco del driver e destinato al microcontrollore

Per come è configurato il connettore 40 pin che interconnette le due schede, tutte le alimentazioni sono già presenti sulla slaveboard. L'implementazione di questo circuito però necessitava di 4 canali di comunicazione tra i driver e il microcontrollore STM32-E407. Il flat cable 40 pin aveva inutilizzati solo 3 pin, in quanto 20 erano già impegnati dall'isolamento realizzato con la Gnd-Frame, 8 dai segnali analogici degli altrettanti canali analogici acquisiti, 4 dalle alimentazioni analogiche e 5 dalle alimentazioni digitali. Da qui la necessità di modificare il circuito driver Can-Bus per rendere sufficienti 3 fili di comunicazione tra la slaveboard e la masterboard.

Questo risultato è stato ottenuto eliminando il segnale Time-Pulse di ritorno dal pin RO del driver Max485. In pratica il segnale TimePulse che viene usato da tutti i componenti dell'FTI, per ottenere una perfetta sincronizzazione, veniva fatto passare dal driver 485 del Can-Bus anche nel modulo madre in modo che il ritardo dovuto alla presenza di questo componente che affligge tutti i moduli periferici che lo ricevono, interessasse anche il modulo principale. Accettando questo errore, dell'ordine di qualche nanosecondo, il segnale viene inviato dal ricevitore GPS direttamente sia al microcontrollore che al driver Can-Bus.

Negli schemi di Figura 3.13 e Figura 3.14, si nota come il circuito era stato progettato e come è stato poi modificato a causa della necessità di usare solo 3 pin di collegamento tra Masterboard e Slaveboard.

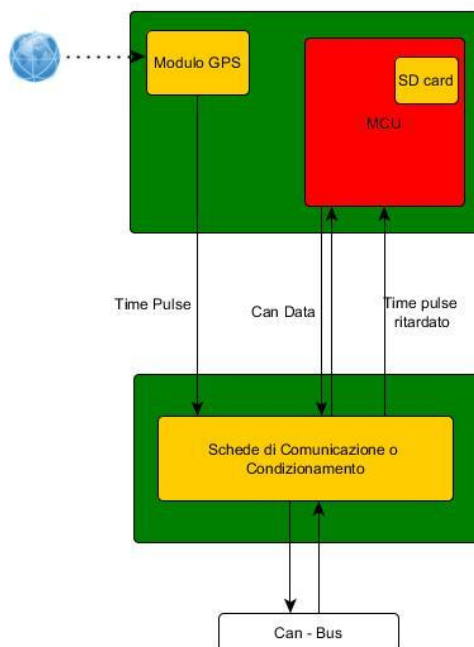
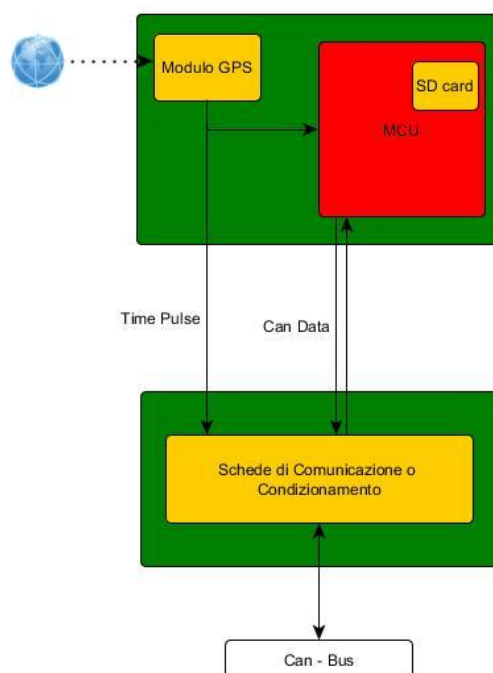


Figura 3.13 - Schema del circuito Driver Can-Bus progettato inizialmente



**Figura 3.14 - Schema del circuito driver Can-Bus realizzato**

In Tabella 3.13 è riportato il nuovo pinout del connettore 40 pin che collega le due schede.

Pinout connettore Masterboard/Slaveboard			
pin	segnale	pin	Segnale
1	Gnd-Frame	2	12Va
3	Gnd-Frame	4	5Va
5	Gnd-Frame	6	Gnda
7	Gnd-Frame	8	12Va
9	Gnd-Frame	10	12Vc
11	Gnd-Frame	12	5Vc
13	Gnd-Frame	14	3,3Vc
15	Gnd-Frame	16	Gndc
17	Gnd-Frame	18	12Vc
19	Gnd-Frame	20	CH1
21	Gnd-Frame	22	CH2
23	Gnd-Frame	24	CH3
25	Gnd-Frame	26	CH4
27	Gnd-Frame	28	CH5
29	Gnd-Frame	30	CH6
31	Gnd-Frame	32	CH7
33	Gnd-Frame	34	CH8
35	Gnd-Frame	36	Time-pulse
37	Gnd-Frame	38	Can TxD
39	Gnd-Frame	40	Can RxD

**Tabella 3.13 - Pinout del connettore 40 pin.**

Nella Figura 3.15 vediamo le due schede connesse dal Flat-Cable 40 pin

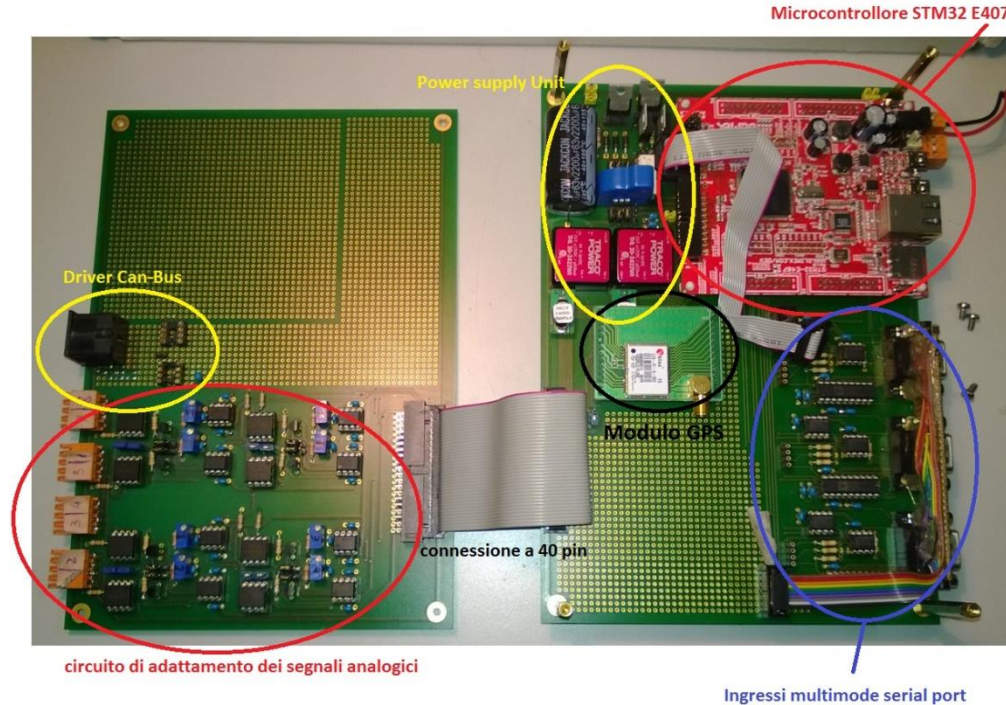


Figura 3.15 - Schede interconnesse con il flat-cable 40 pin.

### 3.7.2 Correzioni e completamento degli stampati

Lo stampato mandato in produzione era affetto da alcuni errori:

- **Alimentazione driver RS485 e RS232**

I circuiti integrati dei driver dei multimode serial port implementati, non erano collegati all'alimentazione. Per rimediare a tale errore sono stati collegati ai +5Vd in fase di saldatura. Questa modifica è stata evidenziata in rosso in Figura 3.16.

- **TxData e RxData per il modulo GPS**

Un altro errore è dipeso da un'imprecisione nella compilazione della tabella della piedinatura relativa all'STM32-E407. In questo file infatti risultava che i piedini D0 e D1 della porta Arduino corrispondevano ai segnali Usart1 Tx e Usart1 Rx. Secondo questa logica quindi lo stampato prevedeva una connessione tra D0 e il piedino RxD e tra D1 e il piedino TxD del ricevitore GPS. In realtà essendo state erroneamente invertite le diciture Tx e Rx, il collegamento dei due segnali andava invertito. La cosa è stata eseguita in fase di saldatura mediante 2 fili saldati direttamente sulle piste.

- **Dimensione dei componenti BTS50085**

Per lo sviluppo di questo progetto è stata costruita una libreria di componenti utilizzata per il programma DesignSpark della RS component. Il problema deriva da un

dimensionamento degli ingombri del componente BTS50085 non corretto. Per questo motivo nel prototipo realizzato e montato, questi componenti si trovano leggermente inclinati rispetto alla posizione corretta di montaggio.

- **Piedinatura integrato AP1117-3.3V**

Per quanto riguarda l'integrato AP1117-3.3V, la piedinatura realizzata sullo stampato non corrispondeva alla realtà per un errore durante la lettura del datasheet, quindi sono state apportate le relative modifiche. Queste ultime nella Figura 3.16 della pagina seguente sono evidenziate in giallo.

Oltre a queste modifiche, sul prototipo di Mnemosine MKIV sono state realizzate delle connessioni post produzione dei PCB perché la scelta dei pin da dedicare alle porte seriali e ai relativi led non era stata completata in tempo.

Queste connessioni, visibili nella Figura 3.16, sono realizzate con dei comuni flat multifilari di colore grigio chiaro.

A seguito di tutti questi controlli e relative correzioni, si è riscontrato un corretto funzionamento di tutto l'hardware.

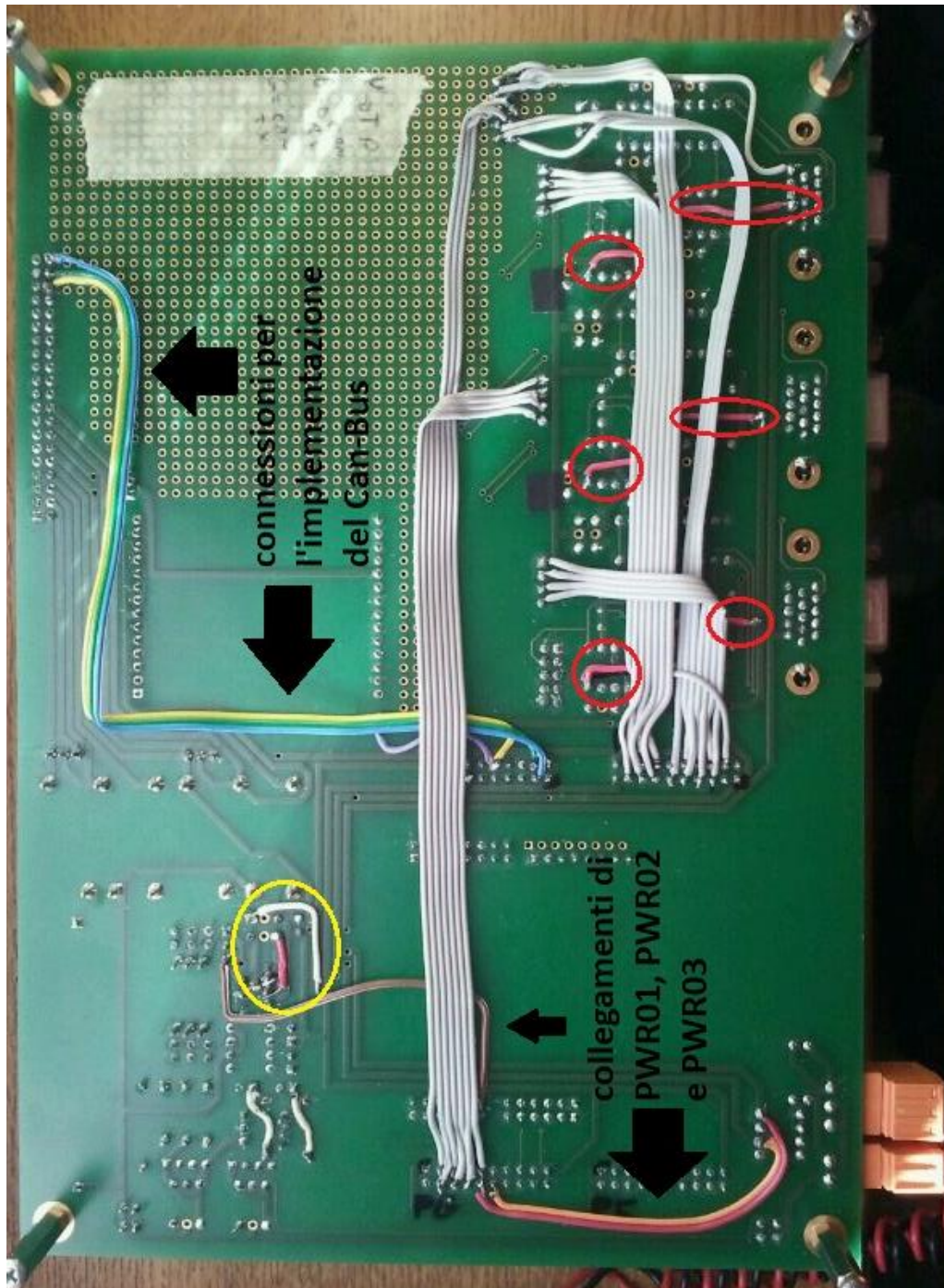


Figura 3.16 - Vista della parte inferiore della Masterboard con modifiche



## 4 Progetto del nodo dati aria

Il nodo dati aria permette al nostro sistema FTI di registrare i valori di pressione e temperatura del flusso d'aria che investe il nostro velivolo. Permette inoltre di acquisire informazioni riguardo la direzione del vento relativo.

Verranno quindi progettati gli schemi elettrici che permettono di interfacciare il microcontrollore del modulo ai sensori che convertono le variabili fisiche in segnali elettrici.

Il progetto definitivo è ancora prematuro ma si potrà comunque fare una stima preliminare riguardo gli ingombri e l'organizzazione dei vari componenti sulla scheda.

Il nodo dati aria, per come è stato concepito, dovrà essere necessariamente collocato in prossimità del Pitot-Boom, quindi all'esterno del cockpit vicino al collegamento tra controventatura e ala. Questa posizione comporta che il modulo verrà investito da un flusso d'aria. Il progetto del case e quindi anche quello degli stampati al suo interno dovranno tenere conto anche di un fattore aerodinamico.

### 4.1 Architettura del nodo dati aria

Come detto in precedenza, i nodi periferici di Mnemosine MK IV dovranno essere così composti:

- eventuale circuito per la comunicazione con sensori con comunicazione seriale;
- eventuale circuito per il condizionamento e l'acquisizione di segnali analogici;
- un microcontrollore per l'elaborazione digitale dei dati;
- un interfaccia Can-Bus per comunicare con gli altri moduli dell'FTI.

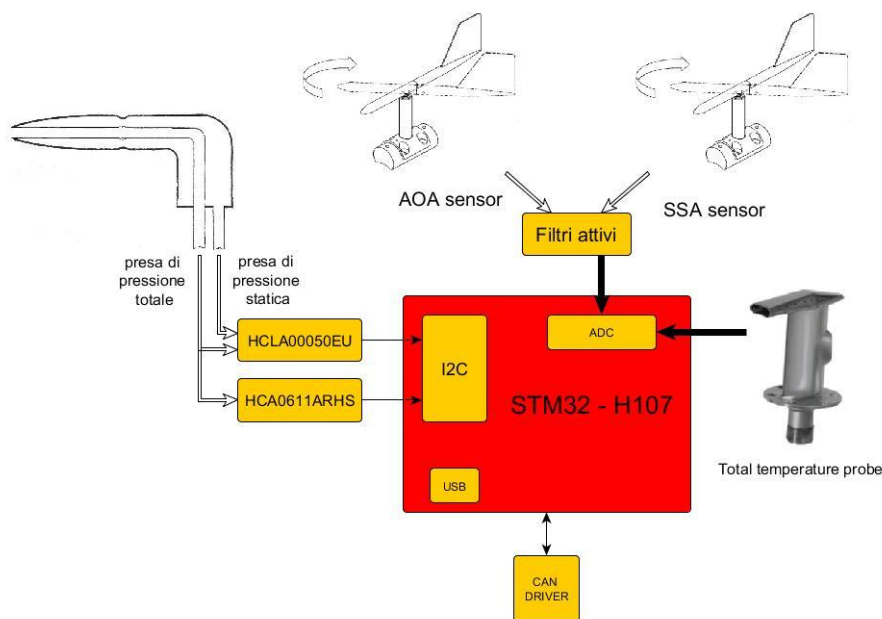
Il nodo dati aria rappresenta uno dei moduli periferici più importanti presenti nel nostro sistema FTI. Esso infatti è previsto in fase di prima progettazione e non è un modulo di espansione del sistema FTI. Questo è indispensabile per permette di analizzare alcune delle prestazioni fondamentali per la certificazione di un velivolo.

Questo modulo è collegato a un dispositivo chiamato Pitot-Boom che è la struttura su cui sono presenti le prese di pressione, il sensore di temperatura e i potenziometri che ci permetteranno di effettuare varie misurazioni.

I parametri acquisiti dai sensori sono:

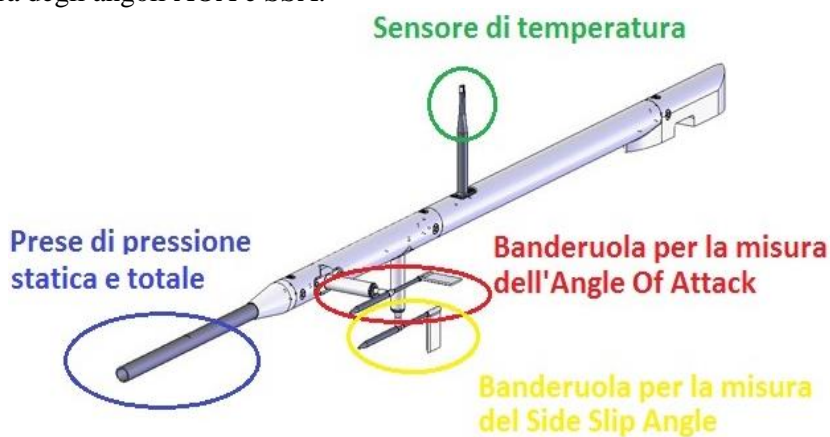
- pressione statica;
- pressione dinamica;
- temperatura totale aria;
- Angle Of Attack (AOA);
- Side Slip Angle (SSA).

Tutte queste misurazioni sono effettuate su un dispositivo chiamato Pitot-Boom che è espressamente nato per la misurazione dei dati aria durante le i voli sperimentali di velivoli in fase prototipale di sviluppo.



**Figura 4.1 - Schema a blocchi del nodo dati aria**

Nella Figura 4.2 vediamo la struttura di un Pitot-Boom con evidenziate le posizioni delle prese di pressione statica e dinamica, il sensore di temperatura, e le banderuole per la misura degli angoli AOA e SSA.



**Figura 4.2 - Struttura di un Pitot-Boom**

Per quanto riguarda le prese di pressione, queste sono disposte come su un normale tubo di pitot. La presenza di questa struttura snella che porta le prese di pressione statica e totale in posizione così avanzata è necessaria per ottenere delle misurazioni migliori, non disturbate dalle dinamiche locali dell'aerodinamica del velivolo.

Sul Pitot-Boom non sono presenti sensori di pressione, ma solo le prese di pressione. Il valore di pressione delle due prese sarà portato ai veri e propri sensori attraverso dei tubi flessibili ermetici. I sensori di pressione veri e propri si trovano infatti all'interno del nodo dati aria montati direttamente sullo stampato. Per evitare una lunghezza dei tubi troppo elevata e per non rischiare di avere quindi degli errori nella misura delle

pressioni dovute a ritardi di propagazione o a eccessiva deformabilità dei tubi, il modulo sarà posizionato in prossimità del Pitot-Boom.

Il dispositivo che contiene il sensore di temperatura, è in grado di misurare la temperatura totale del flusso d'aria. Prende il nome di Total Temperature Probe e alcuni esempi di tipologie per questi dispositivi sono visibili in Figura 4.3.



**Figura 4.3 - Total Temperature Probe**

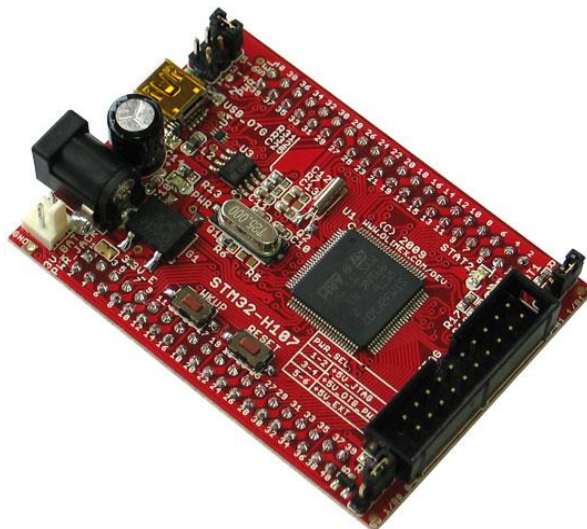
Gli angoli di AOA e di SSA sono misurati tramite due banderuole che si allineano al vento relativo, ognuna delle quali è collegata ad un potenziometro.

Tale potenziometro verrà connesso al nostro nodo dati aria e il segnale elettrico andrà, tramite un circuito di condizionamento, all'ADC del microcontrollore per essere digitalizzato.

#### **4.1.1 Il microcontrollore del nodo dati aria: Olimex STM32-H107**

Il microcontrollore Olimex STM32-H107 prodotto dalla ST Microelectronics è stato scelto per il nodo dati aria perché è dotato di una grandissima capacità di calcolo ma al tempo stesso è montato su una scheda semplice e poco ingombrante, fattore molto importante per il nodo in questione.

Come si vede nella Figura 4.4, la scheda ha ingombri ridottissimi (70x50mm) e l'unica porta implementata onboard è un USB On The Go oltre al connettore JTAG necessario per la programmazione e il debugging.



**Figura 4.4 - Scheda del microcontrollore STM32-H107**

Il microcontrollore STM32-H107 [17] è basato su un Core ARM Cortex-M3 a 32 bit che può lavorare a una frequenza massima di 72 MHz. È dotato di 64Kbytes di memoria SRAM e di 256Kbytes di memoria Flash. Dispone inoltre di un'ampia gamma di collegamenti Input/Output e di due connessioni periferiche a due bus APB. Offre due ADC a 12 bit, quattro timer a 16 bit per applicazioni generali, un timer PWM e è in grado di interfacciarsi con altri dispositivi mediante diversi tipi di comunicazione: I<sup>2</sup>C, SPI, I<sup>2</sup>S, USART e CAN.

#### 4.1.2 I sensori di pressione

I sensori di pressione utilizzati sono realizzati dalla Sensor Technics e rispondono alle seguenti sigle: HCLA0050EU [18] e HCA0611ARHS [19]. Questi due integrati in versione SMD (Surface Mount Device), contengono al loro interno oltre al sensore di pressione vero e proprio, anche un modulo di comunicazione seriale che converte il dato di pressione e lo trasmette come dato seriale con un protocollo I<sup>2</sup>C. Gli stessi componenti in versione Plating Through Holes (PTH) non integrano questa caratteristica.

Il protocollo I<sup>2</sup>C è un protocollo hardware per realizzare una trasmissione seriale sincrona. Essendo di tipo sincrono, questo tipo di comunicazione necessita di 2 linee di comunicazione:

- SDA che è la linea dove vengono trasmesse le informazioni,
- SCL dove viene trasmesso il clock di trasmissione.

Utilizzando la tecnologia PTH per la realizzazione del prototipo del modulo dati aria, sono stati utilizzati i sensori in versione SMD montati manualmente su due zoccoli per integrati in tecnologia PTH in modo da poterli installare senza problemi su una scheda realizzata similmente a quella del modulo principale.

I due integrati, HCLA e HCA, sono però concettualmente diversi:

- HCLA0050EU è un dispositivo integrato dotato di due distinti sensori di pressione che esegue internamente la differenza tra i due valori acquisiti. Questo ci permette quindi di misurare la pressione dinamica che è una misura di tipo indiretto infatti deriva dalla differenza tra pressione totale e pressione statica. Nella Figura 4.5 si notano i due attacchi per il collegamento dei tubi relativi alle pressioni totale e statica che arrivano dal Pitot-Boom.



Figura 4.5 - Sensore HCLA005EU

- HCA0611ARHS, visibile in Figura 4.6, è invece un sensore di pressione assoluta e misura quindi una sola pressione. Nel nostro caso la totale.



Figura 4.6 - Sensore HCA0611ARHS

Grazie alla presenza dell'interfaccia seriale integrata sui due componenti, questi ultimi dovranno semplicemente essere alimentati dal modulo, collegati alle prese di pressione tramite dei tubi flessibili e connessi al microcontrollore attraverso le sue porte seriali USART con una semplice resistenza di pull-up. Questo semplifica di molto l'elettronica relativa al trattamento dei segnali di pressione permettendoci di evitare la realizzazione di un circuito per il condizionamento dei segnali analogici di pressione e un circuito a operazionali in configurazione sottrattore per il calcolo della pressione dinamica dalle misure di pressione statica e totale.

Dalla Figura 4.7 si nota l'estrema semplicità del circuito necessario all'acquisizione dei dati di pressione.

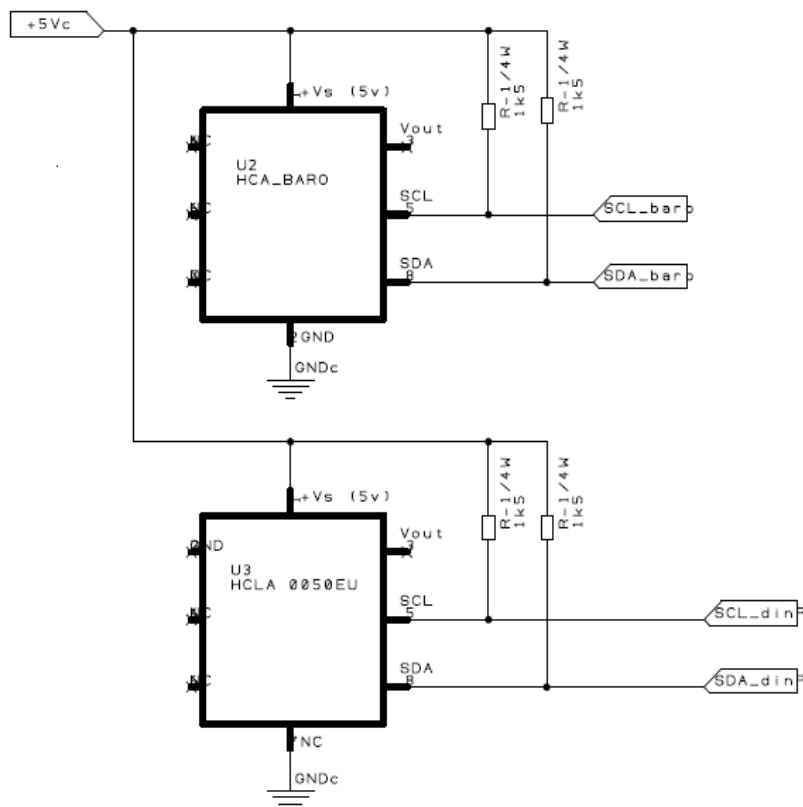


Figura 4.7 - Schematico del collegamento dei sensori di pressione

### 4.1.3 Sensori per AOA e SSA

Per la misura di questi due angoli d'assetto, come si è già detto in precedenza sono presenti due banderuole montate su due potenziometri che quindi danno un segnale elettrico analogico proporzionale alla direzione assunta dalla banderuola. Quest'ultima nelle fasi di volo sarà allineata al vento relativo.

Per il collegamento con il modulo è stato deciso di utilizzare lo stesso connettore degli ingressi analogici del modulo principale in quanto anche queste misurazioni saranno di tipo analogico.

All'interno del nodo dati aria sarà poi presente il circuito di condizionamento del segnale schematizzato nella Figura 4.8.

Questo circuito è costituito da due filtri attivi passa basso del quarto ordine, uno per AOA e uno per SSA, con frequenza di taglio a 100Hz e guadagno pari a 1.

La scelta della frequenza di taglio è stata calcolata considerando una velocità di rotazione massima attorno agli assi di spin e di imbardata dell'ordine dei 10°/s.

Ogni singolo circuito di condizionamento è costituito da due filtri attivi passa basso invertenti del secondo ordine collegati in serie.

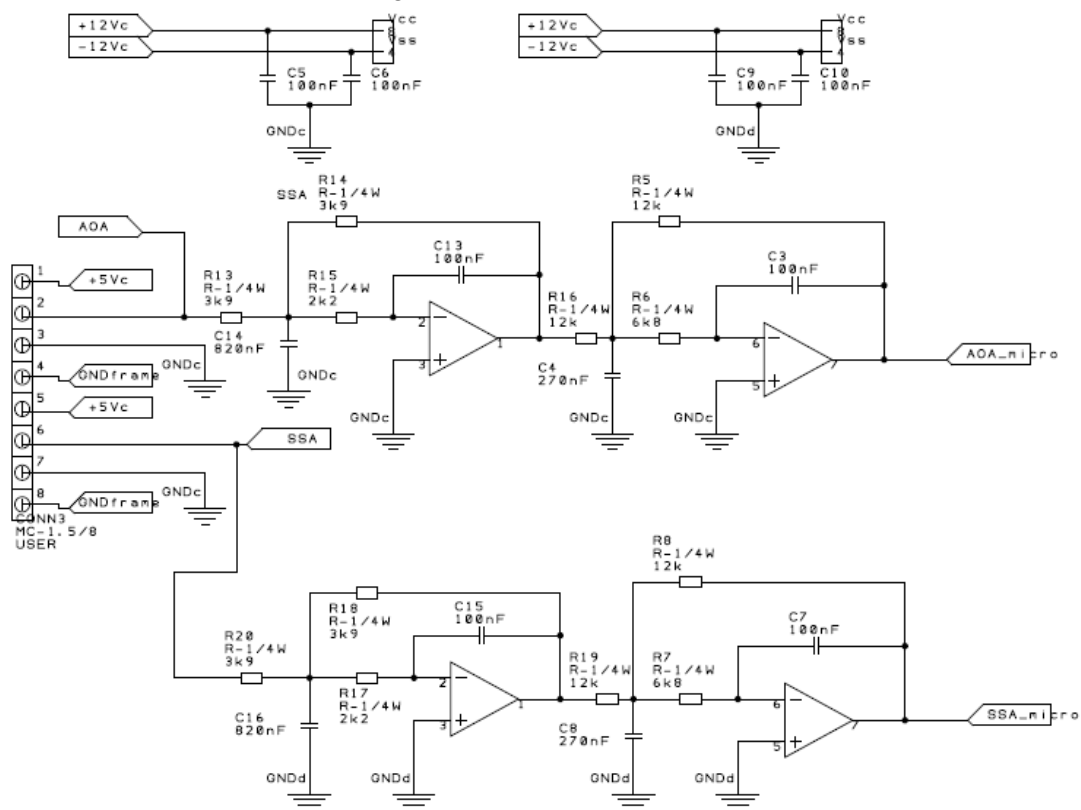


Figura 4.8 - Circuito di condizionamento del segnale per AOA e SSA

Nella Figura 4.8 si nota anche la presenza del connettore otto pin come quello degli ingressi analogici del modulo madre.

I pin dedicati all'acquisizione dei due segnali analogici di AOA e SSA non sono ancora stati assegnati.

#### 4.1.4 Sensore di temperatura totale

Il sensore di temperatura totale è montato sul Pitot-Boom e è costituito da una total temperature probe in grado di misurare direttamente la temperatura totale del flusso d'aria che la investe.

Questo sensore è costituito internamente di un canale aria in grado di rallentare il flusso d'aria fino a fermarlo senza modificarne il valore di temperatura totale mediante diversi accorgimenti visibili in Figura 4.9.

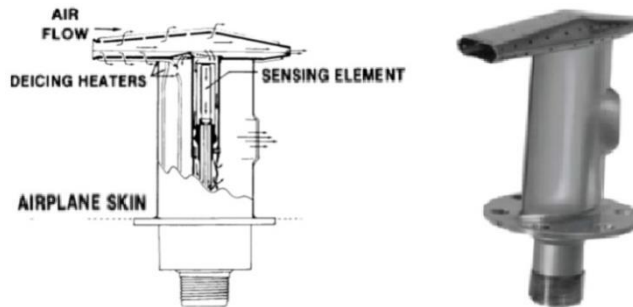


Figura 4.9 - Total temperatur probe

#### 4.1.5 Interfaccia Can-Bus del modulo dati aria

L'interfaccia Can-Bus del nodo dati aria è praticamente identica a quella del modulo principale. Per questo modulo, essendo stato progettato dopo la scelta definitiva dei driver Can-Bus, è prevista l'implementazione del circuito driver Can-Bus direttamente in fase di stampaggio del PCB. Si nota la presenza di due connettori Can-Bus, caratteristica utile nel caso il device sia in un punto intermedio della rete Can.

Il pinout sarà identico a quello del Circuito driver Can-Bus del modulo madre. Nella Figura 4.10 si vede lo schematico di questo circuito.

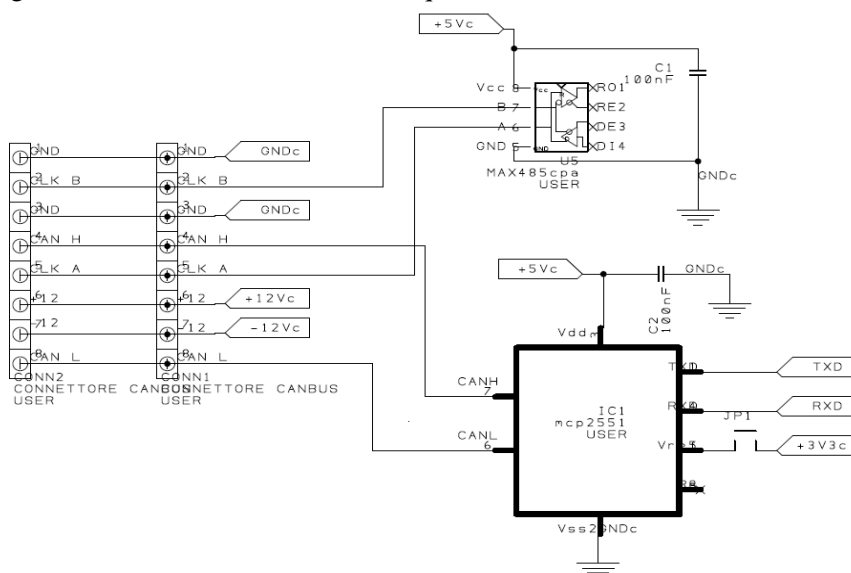
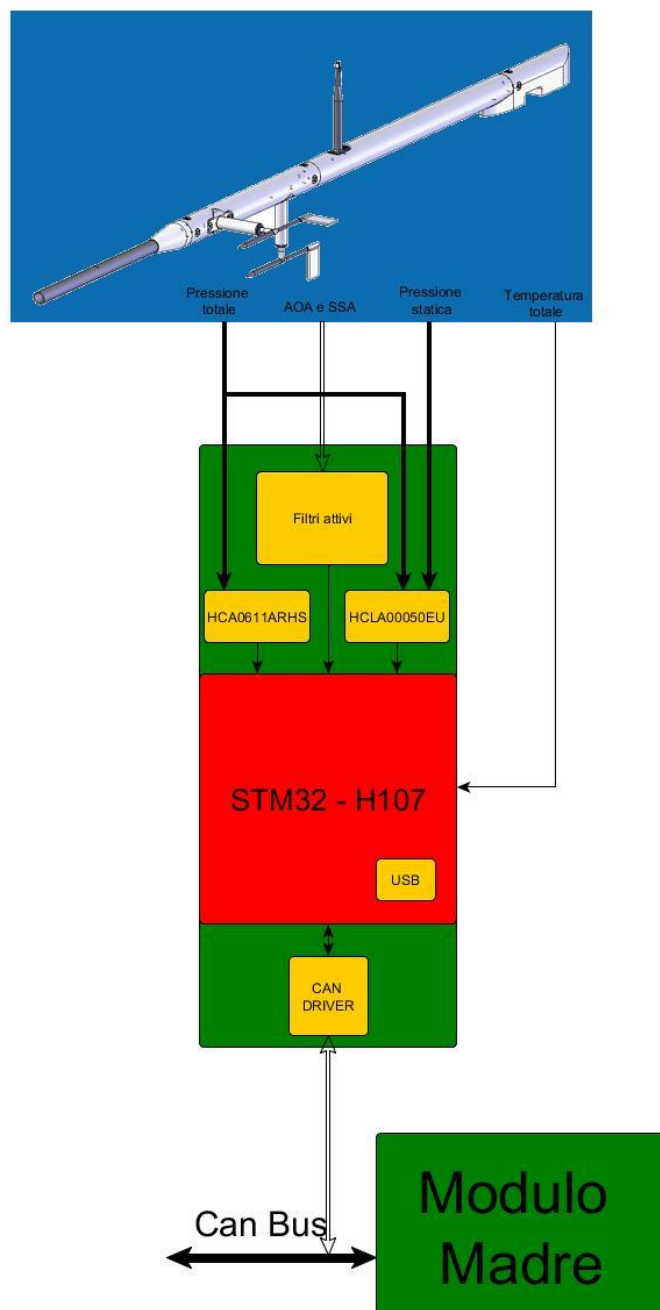


Figura 4.10 - Driver Can-Bus nodo dati aria

## 4.2 Definizione dell'architettura interna e degli ingombri

L'organizzazione dei componenti sulla scheda del modulo dati aria è rappresentata nella Figura 4.11.



**Figura 4.11 - Architettura modulo dati aria**

Il miglior posizionamento del modulo dati aria è stato immaginato dietro al Pitot-Boom, nella sua scia in modo da non disturbare troppo l'aerodinamica del velivolo. Si potrebbe



concepire un case cilindrico montato come un prolungamento del tubo di supporto del Pitot-Boom.

Con una semplice analisi degli ingombri, è stata sviluppata la che evidenzia le seguenti caratteristiche di progetto:

- Per la realizzazione dello stampato si tenga conto che la dimensione trasversale massima dovrà essere al più uguale a quella della scheda del microcontrollore per poter montare le schede all'interno di un case cilindrico posizionato come prolungamento del Pitot-Boom.
- L'organizzazione interna al case dovrà prestare particolare attenzione agli ingombri delle schede stampate, dei componenti elettronici ma anche dei tubi che portano l'informazione di pressione ai sensori montati sugli stampati.
- La sezione trasversale del connettore cilindrico dovrà essere ben sfruttata per permettere il posizionamento di tutti i connettori previsti.

Per realizzare ciò si deve tener conto degli ingombri trasversali del modulo e fare in modo che questi non superino eccessivamente la misura del diametro del Pitot-Boom cioè 50mm.

La dimensione trasversale del microcontrollore è di 50mm. Sapendo che questo andrà montato su un'altra scheda PCB mediante StripLine, si può pensare di realizzare uno stampato con le stesse dimensioni trasversali del microcontrollore e al limite più lungo in modo da ottenere due schede sovrapposte della stessa larghezza da inserire nel case del modulo dati aria. Quest'ultimo sarà come un tubo che prolungherà il supporto del Pitot-Boom. Le Figura 4.12 e Figura 4.13 mostrano idealmente il risultato di questa disposizione.

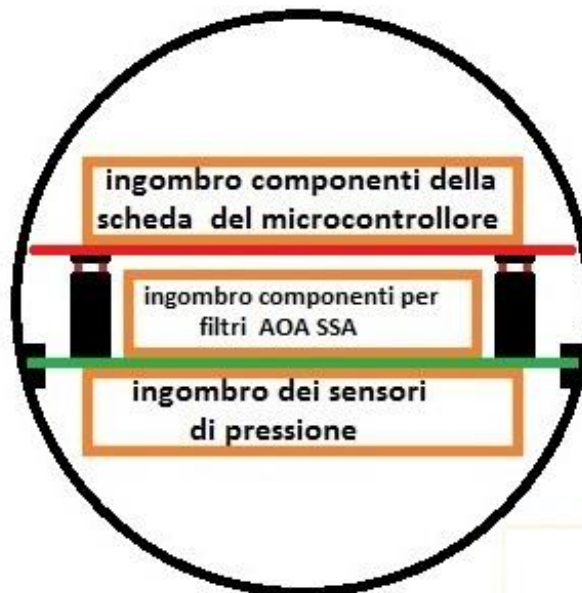


Figura 4.12 - Organizzazione dei componenti del nodo dati aria all'interno del case

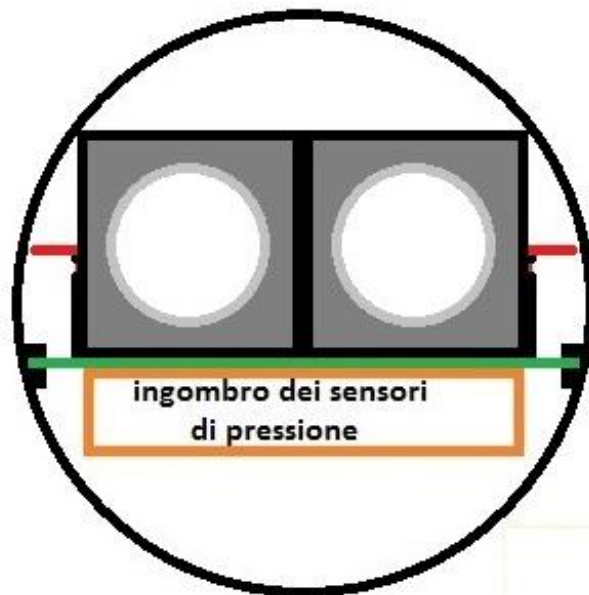


Figura 4.13 - Vista posteriore del nodo dati aria con i due connettori CAN-BUS

Nelle Figura 4.14 Figura 4.15 vediamo due viste 3D rispettivamente della parte superiore e della parte inferiore del nodo dati aria realizzate grazie al programma Design Spark [20].

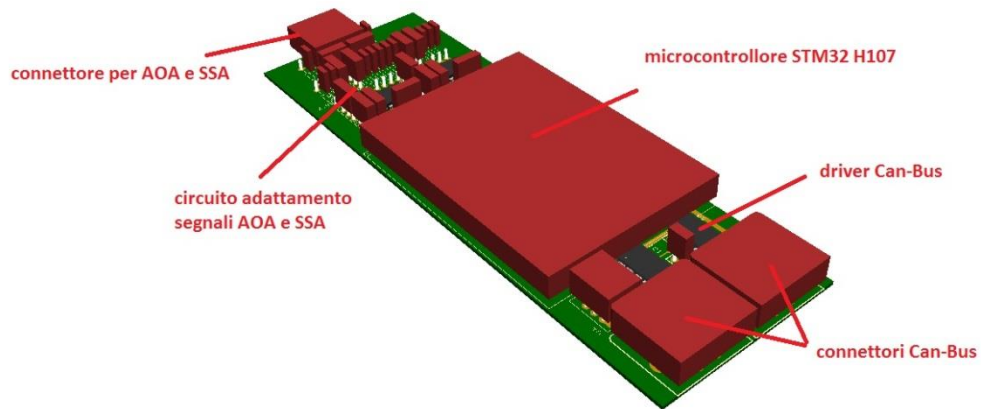
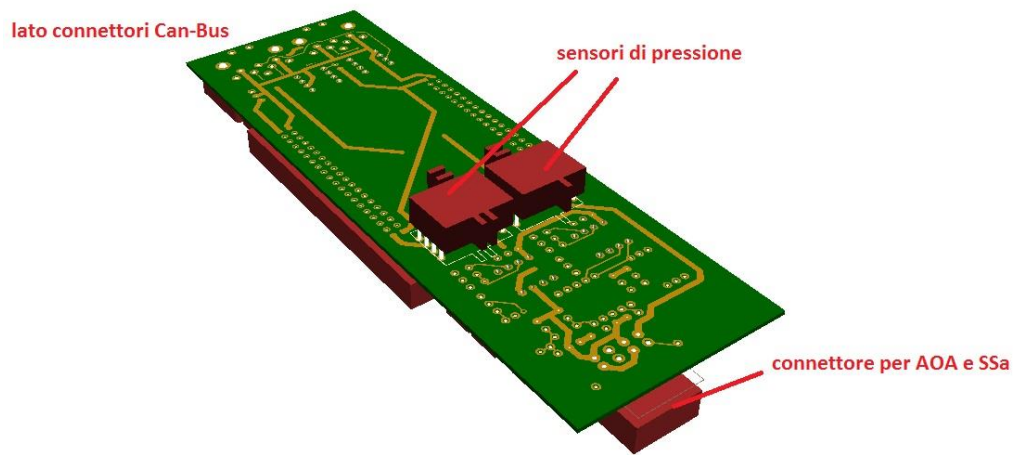


Figura 4.14 - Vista superiore del PCB del nodo dati aria



**Figura 4.15 - Vista inferiore del PCB del nodo dati aria**

È stato deciso di posizionare i sensori di pressione sul lato inferiore del PCB in modo da esser sicuri i componenti elettronici e i tubi per la misura delle pressioni non creino interferenze.

## 5 Sistema Operativo Real-Time (RTOS)

Dopo aver definito l'architettura del sistema FTI, si passa alla realizzazione del software che controllerà l'hardware in modo da acquisire, elaborare e salvare i dati.

La terminologia Real-Time sta a indicare un processo che risponde in tempo reale a una sollecitazione. In ambito informatico sta a indicare quei software il cui corretto funzionamento dipende dal tempo di risposta. Questo significa che questi programmi devono rispondere a determinati input entro tempistiche prestabilite.

Prima di continuare a parlare di sistemi operativi, verranno spiegati i significati di alcune terminologie fondamentali per comprendere l'argomento:

- **Shared Resource:** sono risorse usate da molti task. Ognuno dei quali ha un accesso esclusivo per prevenire dei conflitti durante letture e scritture. Questo modo di operare è chiamato mutual exclusion e è una caratteristica che deve essere presente nel sistema operativo real time scelto.
- **Multitasking:** è la metodologia con cui la Central Processing Unit (CPU) schedula e si organizza l'elaborazione dei vari processi in background; il Multitasking dà appunto la possibilità al sistema di avere più processi in background. Questa caratteristica facilita la programmazione perché permette di definire diversi processi eseguiti da una singola periferica. La gestione di un Sistema real-time multitasking è più semplice di un equivalente monotasking. Attualmente esistono diversi tipi di sistemi operativi ognuno con le proprie particolarità e peculiarità che verranno espone in seguito e che ci porteranno a scegliere una determinata tipologia di RTOS piuttosto che un'altra.
- **Task:** chiamati anche Thread, sono semplici programmi che sono scritti come se venissero eseguiti singolarmente dalla CPU. La costruzione del processo real-time vero e proprio subentra quando il lavoro suddiviso in vari Task indipendenti, viene eseguito da un'unica CPU. Quest'ultima non fa altro che assegnare a ogni task una porzione di memoria e una stack area e eseguire tutti i processi secondo delle priorità assegnate.
- **Mutexes:** con questo termine si intende un processo di mutua esclusione che impedisce a più task di accedere a aree di memoria o ad altre risorse soggette a race condition (corsa critica).
- **Kernel:** è la parte del sistema operativo multitasking responsabile della gestione dei singoli task. Si occupa ad esempio dell'intercomunicazione tra i vari processi, dei tempi dedicati a ognuno di essi e delle priorità che li regolano. Il kernel aggiunge un carico di lavoro al sistema non strettamente legato ai task in quanto esso stesso richiede un tempo di esecuzione. L'ammontare del carico di lavoro aggiuntivo, dipende da quante volte questo servizio viene richiamato e ovviamente da come il kernel stesso sia fatto. Uno dei più importanti indici di performance del kernel, è il Context Switch. Il context è l'insieme dei dati che la CPU utilizza per l'esecuzione di un singolo task. Quando il kernel multitasking decide di cambiare il task in esecuzione, esso salva il context della CPU associato al processo in esecuzione in un'area di memoria denominata

context storage area. Successivamente prende il context del nuovo task che deve elaborare, lo carica, e riparte con l'esecuzione dei processi.

- **Preemptive kernel:** con questa terminologia si indica che il task con la massima priorità può prendere in qualsiasi momento il controllo della CPU. Questo torna utile quando per un task la velocità di risposta del sistema ha la massima importanza come nel caso di una criticità (es: mancanza di alimentazione esterna). Quando durante l'elaborazione, un Task a elevata priorità richiede la propria esecuzione, il task corrente viene messo in sospenso e il processo ad elevate priorità viene inviato alla CPU e elaborato.

Come nei normali sistemi operativi, anche in quelli Real-time esistono quindi dei processi denominati Task che eseguono determinate funzioni. Il tempo di risposta di questi Task è proprio il fattore critico per i sistemi operativi con cui andremo ad operare.

Per ottenere un RTOS, esistono due tipi di soluzioni così denominate:

- **Soft Real-Time** dove i task vengono eseguiti correttamente dal Sistema il più velocemente possibile in modo da avere un tempo di risposta basso, ma non esiste alcun vincolo sul massimo tempo necessario ad eseguire completamente il task.
- **Hard Real-Time** dove invece i processi non solo devono essere eseguiti correttamente ma devono anche essere terminati entro un tempo prefissato.

In realtà la separazione tra queste due tipologie non è proprio così netta, infatti Mnemosine MK4, come la maggior parte dei processi Real-Time oggi utilizzati, è dotato di un RTOS di tipo *foreground/background* denominato anche *super-loops*.

Questo tipo di sistemi operativi è caratterizzato da un continuo ciclico richiamo dei task che il sistema deve eseguire sempre e che viene chiamato *background* o *task level*. Per le routine asincrone, esse verranno eseguite secondo delle Interrupt Service Routine (ISR) che elaborano questi processi a un livello chiamato *interrupt level*.

I processi critici verranno fatti eseguire dalle ISRs in modo da essere sicuri che la loro esecuzione rispetti i requisiti in termini di tempistiche di esecuzione. Ed è per questo che tali processi hanno la tendenza a durare più del tempo che dovrebbero richiedere. Inoltre le ISR non sono elaborate fino a che i processi in Task level non liberano il microcontrollore.

Tornando al nostro RTOS, Mnemosine deve garantire il determinismo degli eventi, quindi ogni Task ha uno specifico tempo massimo per l'esecuzione.

## 5.1 Scelta dell'RTOS

Dopo una prima analisi del mondo degli RTOS e dopo aver escluso tutti quelli dotati di una licenza proprietaria, sono rimaste due sole alternative: ChibiOS/RT [21] e FreeRTOS [22].

Entrambi sono distribuiti sotto licenza General public Licence V3 (GL3) che significa sostanzialmente che è possibile utilizzare il codice o parte di esso per scopi commerciali.

La discriminante che ha portato alla scelta di ChibiOS/RT è che questo sistema operativo è progettato per applicazioni in tempo reale dove integrabilità, efficienza di esecuzione e codice compatto sono requisiti fondamentali. Questo RTOS è caratterizzato da elevata portabilità, struttura compatta e grazie a un'architettura ottimizzata, un efficientissimo context switching. Parlando di numeri, con il processore utilizzato in Mnemosine MK IV, con una dimensione del kernel di 6172byte, il context switch time è di soli 0.4µs con tutti i sottosistemi abilitati escludendo solo quelli di debug.

La memoria è allocata staticamente in fase di compilazione. Tuttavia ne è previsto anche un utilizzo dinamico supportato da uno strato opzionale su uno strato superiore del core statico. C'è un completo set di funzioni primarie che definiscono Treads, virtual timers, semaphores, mutexes, variabili condizionali, messaggi, indirizzi fisici, segnalatori di eventi, code di esecuzione. Chibios/RT supporta inoltre algoritmi di priorità sul passaggio dei mutexes.

Per la gestione dei principali device presenti sulla scheda del microcontrollore, come ad esempio lo slot µSD, o la porta Ethernet, è presente un Hardware Abstraction Layer (HAL). L'HAL può essere immaginato come l'interfaccia tra ChibiOS/RT e l'hardware su cui viene caricato. È semplicemente una raccolta di driver che permette una facile gestione dei componenti esterni come il microIP (uIP), light wheight IP (lwIP) e File Allocation Table File Sistem (FatFS) essenziali per il corretto funzionamento dello slot µSD.

## 5.2 Descrizione dei vari Threads

Il software sviluppato come detto prima è suddiviso in vari threads ognuno con uno specifico compito.

Il software completo in fine dovrà soddisfare le seguenti necessità:

- **Acquisizione dei seguenti parametri:**
  - Dati Aria
    - Pressione Statica,
    - Pressione Dinamica,
    - Temperatura totale
    - Angle Of Attack
    - Side Slip Angle
  - Dati GPS:
    - Posizione 3D in coordinate ECEF
    - Velocità in coordinate ECEF
  - Posizione delle superfici di controllo:
    - Alettoni;
    - Timone;
    - Elevatore;
    - Flap
  - Sforzi di barra;
  - Dati motore:
    - Revolution Per Minute (RPM)

- Exhaust Gas Temperature (EGT)
  - Fuel Flow
- **Comunicazione attraverso vari standard:**
  - porta Ethernet;
  - protocollo CAN
  - I<sup>2</sup>C
- **Manutenzione del software tramite porta USART**
- **Assicurare la sincronia dei dati acquisiti.**

Verranno ora descritti i Thread che compongono l'intero codice.

### 5.2.1 Main

Il Main [21] è quella parte di programma che ha il compito di inizializzare il sistema operativo secondo i parametri operativi della scheda OLIMEX STM32E407. Abilita la creazione di tutti i thread e delle rispettivi indirizzi di memoria. Sostanzialmente il lavoro del main finisce con la fase di inizializzazione quando il Time scheduler grazie al sistema operativo prende il controllo dell'intero software.

### 5.2.2 Time Scheduler

Il Time scheduler [21] non è solo un semplice thread ma è un insieme di funzioni gestite indipendentemente tramite l'uso dell'hardware e degli interrupt software. Esso gestisce il segnale Time pulse in arrivo dal Ricevitore GPS e in base a questo e alle frequenze di campionamento richieste delle varie misurazioni richieste, fa partire un timer secondo il quale andrà ad abilitare l'esecuzione dei vari task.

L'ordine di priorità dei vari task, è definito nel Main secondo il tempo necessario all'esecuzione stessa del task. L'assegnazione delle priorità non è una cosa da sottovalutare a causa della natura complessa dei sistemi real time.

Una interessante tecnica per definire queste priorità è la Rate Monotonic Scheduling (RMS) che assegna la priorità in base a quanto è frequente l'esecuzione del task: un task eseguito più di frequente avrà una maggiore priorità.

Supponendo di avere un set di n task con un assegnata priorità RMS, il teorema RMS base stabilisce l'arresto forzato di ogni task è stabilito dal non soddisfacimento della seguente disuguaglianza:

$$\sum_i \frac{E_i}{T_i} < n(2^{1/n} - 1)$$

Dove  $E_i$  è il tempo massimo di esecuzione del task;  $T_i$  è il tempo di esecuzione del task.

In altre parole  $\frac{E_i}{T_i}$  corrisponde alla frazione di tempo richiesta all'esecuzione dell'iesimo task rispetto al tempo che abbiamo deciso di dedicargli. È probabile che Mnemosine MK IV non possa essere gestito interamente secondo questa filosofia, ma la tecnica RMS è un buon punto di partenza.

### 5.2.3 SD Thread

Il thread SD [21] lavora grazie all'HAL e al modulo SDCard di ChibiOS. I driver SDC lavorano secondo una logica a stati. Questo comporta che non tutte le funzionalità possono essere usate in ogni momento. Infatti ogni comando non espressamente espresso nel diagramma in Figura 5.1 è da considerarsi un errore.

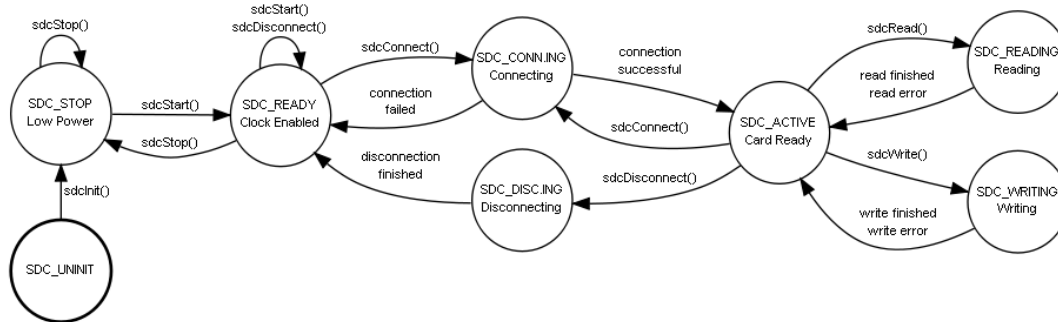


Figura 5.1 - Diagramma agli stati del driver SDC

Questo diagramma va letto da sinistra a destra:

Dopo la fase di inizializzazione, il thread rimane in attesa fino al segnale del time scheduler.

Il software deve garantire:

- Una cartella sulla SD dedicata a ogni missione
- Il salvataggio delle misure acquisite in un determinato intervallo di tempo
- L'inizio e la fine della registrazione
- La comunicazione del suo stato e di ogni eventuale errore
- Tenere traccia dello spazio di memoria occupato
- Tenere traccia dello spazio di memoria libero
- Il montaggio e lo smontaggio della memoria SD
- Una procedura di Recovery che permetta il salvataggio dei dati senza perdita di informazioni.

In accordo con la gestione delle memorie SD, è necessario un file system. ChibiOS usa una formattazione FatFs. Fat Fs è una formattazione standard per supporti embedded di piccole dimensioni.

Questa formattazione, indipendentemente dall'hardware su cui lavora, ha le seguenti caratteristiche:

- Compatibilità con il file system FAT di Windows
- È una piattaforma indipendente e facilmente implementabile su differenti macchine



- Il file di sistema per la definizione della formattazione occupa una porzione di memoria molto piccola
- Permette di essere vista in differenti volumi mediante partizioni
- Supporta nomenclature dei file lunghe in ANSI/OEM o Unicode
- Supporto di RTOS
- Supporto il multiple sector size
- Supporta la modalità in sola lettura
- Ha un'Application Program Interface minimizzata
- Ha un buffer Input/Output.

### 5.2.4 Ethernet Thread

Il thread che gestisce la porta Ethernet [21] utilizza un light-weight Internet Protocol (lwIP) che è un implementazione ridotta indipendente della suite Trasmissione Control Protocol / Internet Protocol (TCP/IP).

Il punto cardine dell'utilizzo della lwIP è di ridurre l'utilizzo di risorse rispetto alla TCP/IP. Questo rende l'lwIP adatta ai sistemi embedded con decine di kilobytes di memoria RAM e pochi kilobytes di Read Only Memory (ROM).

### 5.2.5 Can Thread

Il thread CAN [21] lavora grazie all'HAL e la modulo CAN di ChibiOS.

Il CAN Driver lavora secondo una logica a stati, quindi non tutte le funzionalità sono eseguibili in ogni momento. Ogni comando non espressamente espresso nel diagramma in Figura 5.2 è da considerarsi un errore.

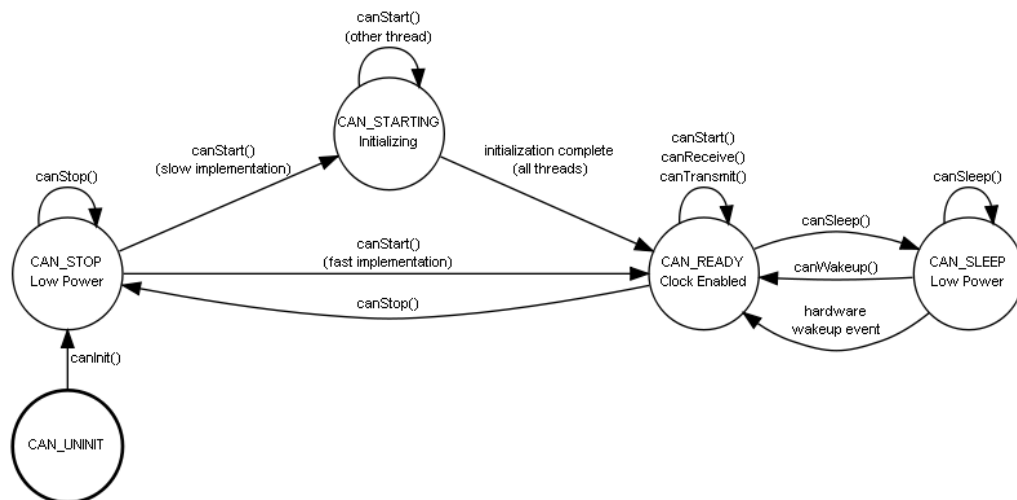


Figura 5.2 - Diagramma agli stati del Driver CAN

## 5.2.6 GPS Thread

Il Thread dedicato ai dati GPS [21], non è altro che un Thread per Gestire la comunicazione seriale con il ricevitore GPS montato sull'FTI.

La comunicazione avviene in modalità I<sup>2</sup>C.

Il Thread comincia inizializzando la comunicazione col ricevitore dopo di che rimane in attesa che il time scheduler richieda i dati GPS per instaurare la comunicazione.

Il time scheduler interrogherà il driver GPS ogni 200ms (5Hz).

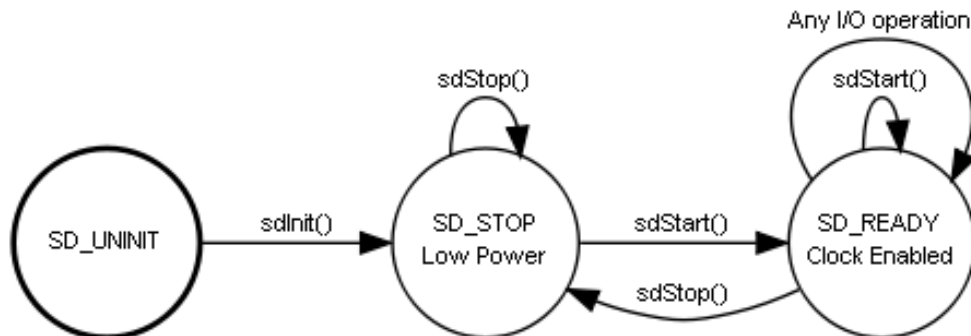


Figura 5.3 - Diagramma a blocchi per il Driver di comunicazione seriale

## 5.2.7 UART Threads: AHRS, CDU e Stick Force

Ci sono vari dispositivi che comunicano tramite UART con il sistema FTI.

In primo luogo l'Attitude and Heading Reference System (AHRS) ma anche LA Command Data Unit e il modulo Stick Force.

Tutti questi dispositivi comunicano con l'FTI tramite Uart, e sebbene le priorità di questi thread siano differenti, essi vengono descritti dal medesimo schema a blocchi.

La massima priorità è dell'AHRS che viene interrogata ogni 20ms (50Hz).

Il dispositivo per lo StickForce sarà interrogato ogni 100ms (10Hz) mentre per quanto riguarda la CDU, essendo un dispositivo ancora ipotetico, non sono ancora stati definite tempistiche di interrogazione.

Il thread UART [21] comincia con l'inizializzazione della comunicazione. Segue poi un test per verificare il corretto funzionamento della comunicazione. In caso di errore, questo sarà comunicato tramite i Led dedicati ad ogni canale UART. Il Thread continua poi con l'invio di un messaggio di richiesta di dati, seguito da un ciclo di attesa. Nelle Figura 5.4 Figura 5.5 e Figura 5.6 vengono mostrati i diagrammi a blocchi che spiegano il funzionamento rispettivamente de: l'inizializzazione della comunicazione UART, la procedura per la trasmissione di un messaggio UART, e la procedura per la ricezione di un messaggio UART

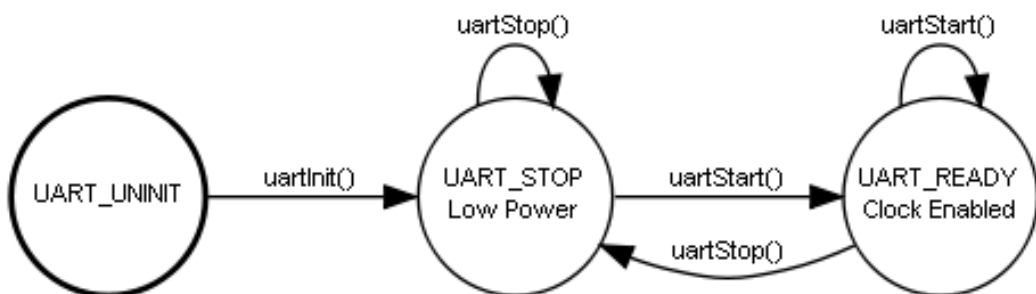


Figura 5.4 - Diagramma a blocchi per comunicazione UART

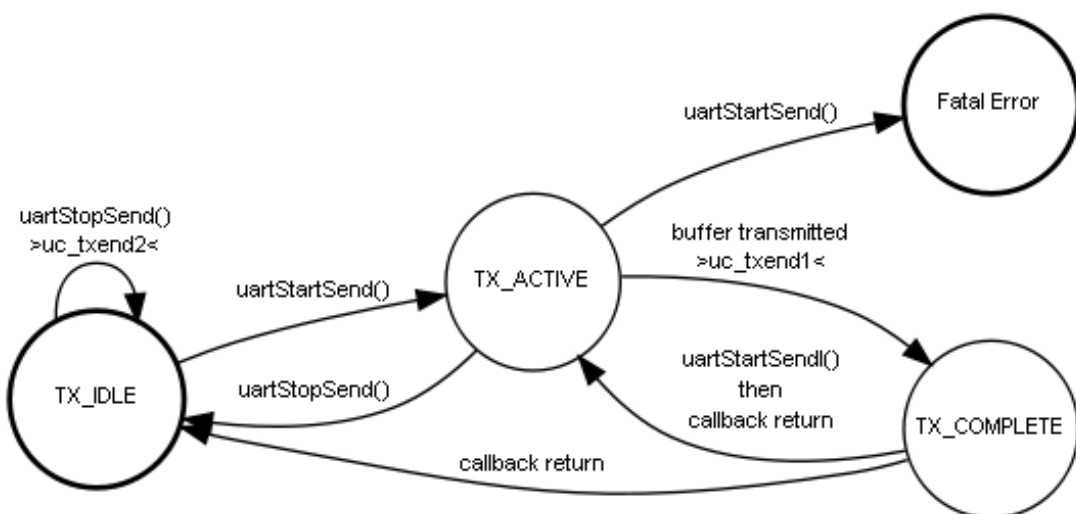


Figura 5.5 - Diagramma per la trasmissione UART

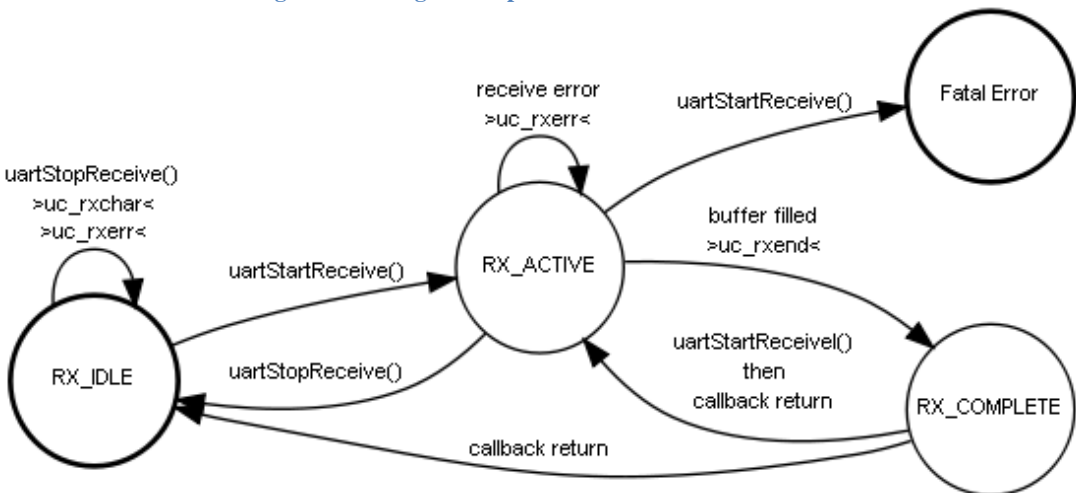


Figura 5.6 - Diagramma per la Ricezione UART

## 5.2.8 Control surface position Thread

Il Thread per il monitoraggio della posizione delle superfici di controllo, si occupa dell'acquisizione dei segnali analogici che arrivano da potenziometri collegati a queste ultime. Dovendo acquisire segnali analogici, è fondamentale l'utilizzo del convertitore ADC. Il Thread deve quindi configurare l'HAL e il modulo ADC di ChibiOS [21].

I driver ADC lavorano secondo una logica a stati. Questo comporta che non tutte le funzionalità possono essere usate in ogni momento. Infatti ogni comando non espressamente espresso nel diagramma in Figura 5.7 è da considerarsi un errore.

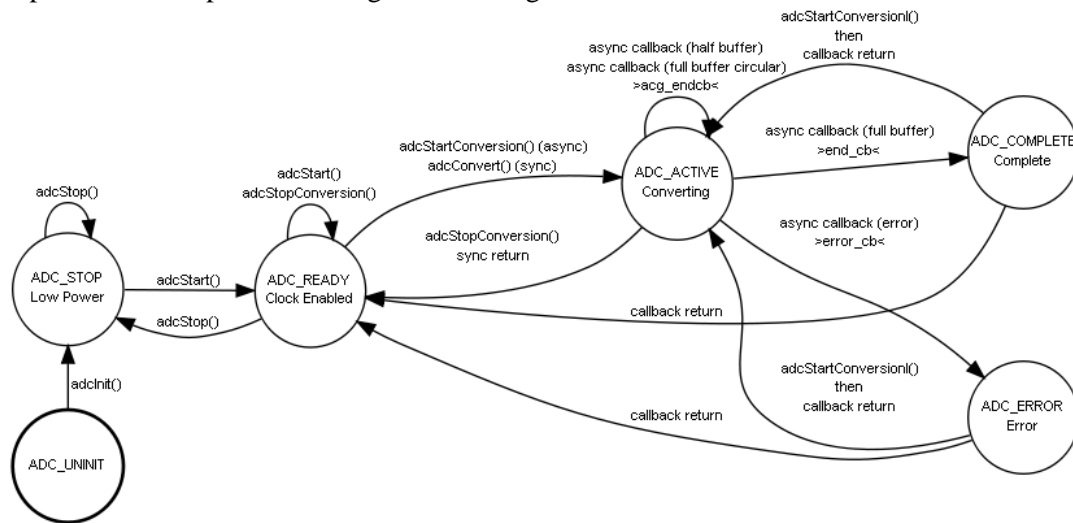


Figura 5.7 - Diagramma per la gestione ADC

L'ADC Conversion Group è colui che definisce le operazioni di conversione da misura analogica a dato digitale.

I Driver ADC supportano tre diverse modalità di conversione:

- **One Shot:** il driver esegue la conversione e poi si arresta;
- **Linear Buffer:** il driver esegue una serie di conversioni e si arresta, in pratica esegue una conversione One Shot N volte. A ogni conversione, il puntatore del buffer viene incrementato. Questo fino all'esaurimento della capienza del buffer. Dopo di che il processo di conversione viene arrestato.
- **Circular Buffer:** esegue la medesima operazione del Linear Buffer, ma il processo non si arresta con il riempimento dello stesso ma continua facendo ripartire il puntatore dall'inizio.

Il driver è in grado di inviare un segnale durante il processo di conversione. Questo segnale generalmente viene inviato alla fine del processo di conversione o, come nel caso del circular buffer, quando il Buffer è saturo e il puntatore viene resettato. Nelle modalità linear e circular buffer, un segnale viene inviato anche quando il Buffer ha raggiunto la metà della sua capienza.

È quindi possibile utilizzare una diversa modalità a seconda della necessità. Il Circular Buffer è molto interessante quando si ha la necessità di eseguire una media su una serie di misurazioni per eseguire un primo filtro del dato acquisito.

### 5.2.9 Air Data Thread

Questo Thread non interessa il nodo principale di Mnemosine ma il modulo dati aria di cui si è parlato nel capitolo “Progetto del nodo dati aria”.

Il software Real Time che verrà caricato su questo dispositivo avrà i threads per Can-Bus e segnali analogici già precedentemente oltre al main e al time scheduler.

Per quanto riguarda i sensori di pressione, questi comunicano tramite protocollo I<sup>2</sup>C. Per poterlo implementare, dovranno essere configurati l’HAL e il modulo I<sup>2</sup>C di ChibiOS [21]. Il driver implementa il diagramma agli stati di Figura 5.8.

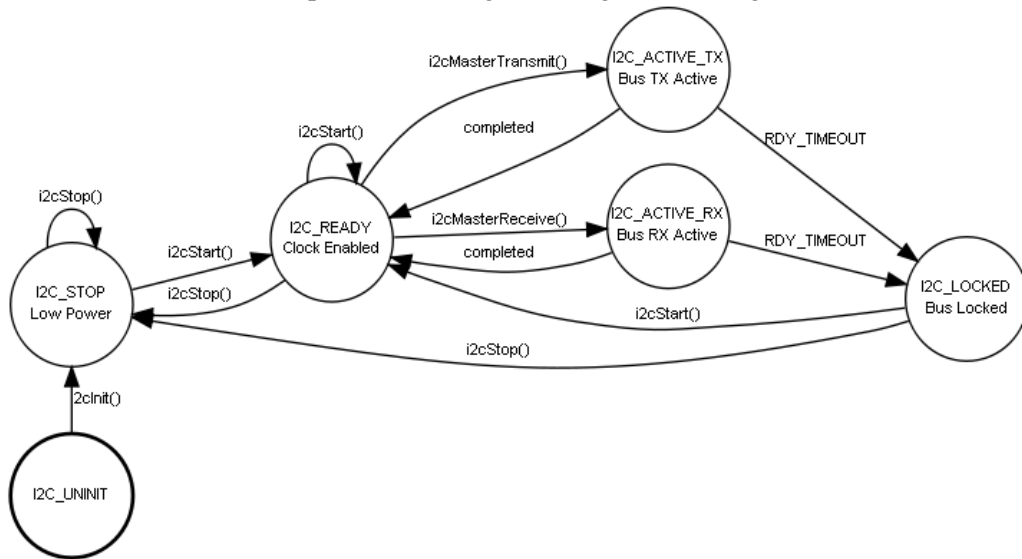


Figura 5.8 - Diagramma agli stati del driver I<sup>2</sup>C

## 6 Conclusioni e sviluppi futuri

Mentre viene redatto questo documento, sono in corso i primi test di funzionamento di tutti i vari dispositivi a bordo di Mnemosine MK IV. Lo sviluppo si trova ancora ad un livello iniziale. Il prossimo step di sviluppo fondamentale per la riuscita in tempi brevi del progetto riguarda innanzitutto la redazione di un software completo che permetta di implementare tutte le dinamiche del modulo madre di Mnemosine MK IV contemporaneamente.

Dovrà essere realizzato in tempi ragionevoli anche il prototipo hardware del nodo dati aria, da decidere se in tecnologia PTH o SMD, e del relativo software in modo da poter avere un sistema FTI con la minima dotazione necessaria per poter ottenere una banca dati completa dai Flight Test.

L'obiettivo in termini di tempo è di poter allestire l'FTI composto da Modulo principale e dal nodo dati aria in tempo per l'utilizzo durante il corso di Sperimentazione in Volo nella primavera del 2014.

Alcuni sviluppi futuri per l'implementazione di nuove funzionalità e affinamento di quelle già previste sono già stati pensati e studiati nel corso di alcune tesi.

Questi sono:

- L'utilizzo del modulo GPS Lea 6n, ricevitore GPS in grado di utilizzare anche la costellazione GLObal NAVigation Satellite System (GLONASS), equivalente russo del più famoso GPS;
- Lo sviluppo di nuovi moduli implementino nuove funzionalità come:
  - Comunicazione con una Ground Station;
  - Possibilità di interfacciamento con una strumentazione EFIS;
  - Interfacciamento con una CDU con tecnologia bluetooth in dotazione all'FTE;
- Realizzazione del modulo madre con tecnologia SMD ottenendo una sicura diminuzione degli ingombri.

Per la loro implementazione si rimanda a tesi future.

## Bibliografia

- [1] «[www.sfte.org](http://www.sfte.org),» ottobre 2013. [Online].
- [2] «<http://www.ulm.it/>,» ottobre 2013. [Online].
- [3] *Decreto del Presidente della Repubblica, Nuovo regolamento di attuazione della legge 25 marzo 1985, n. 106, concernente la.*
- [4] «<http://www.cwc-ae.com/flight-test>,» ottobre 2013. [Online].
- [5] C. Cardani, A. Folchini e A. Rolando, «MNEMOSINE: A FEDERATED FLIGHT TEST INSTRUMENTATION,» in *XIX CONGRESSO NAZIONALE AIDAA, FORLÌ (FC)*, dal 16 al 21 settembre 2007.
- [6] Farnell, «PROFET® Data Sheet BTS50085-1TMA».
- [7] LEM, «Current Trasducer LTS6-NP Data Sheet».
- [8] TRACO POWER, «DC/DC Converters, THL 20WI Series, 20 Watt Data Sheet».
- [9] MURATA, «78xxSR Series Data Sheet».
- [10] Olimex, «STM32-E407 development board Revision A,» Luglio 2012.
- [11] Adrio Communications Ltd, «EIA RS 232 Standard,» Ottobre 2013. [Online]. Available: [http://www.radio-electronics.com/info/telecommunications\\_networks/rs232/eia-rs232-c-d-standards.php](http://www.radio-electronics.com/info/telecommunications_networks/rs232/eia-rs232-c-d-standards.php).
- [12] Adrio Communications Ltd, «RS-485 tutorial,» Ottobre 2013. [Online]. Available: [http://www.radio-electronics.com/info/telecommunications\\_networks/rs485/rs-485-tutorial-basics.php](http://www.radio-electronics.com/info/telecommunications_networks/rs485/rs-485-tutorial-basics.php).
- [13] Adrio Communications Ltd, «RS-422 serial transmission,» Ottobre 2013. [Online]. Available: [http://www.radio-electronics.com/info/telecommunications\\_networks/rs422/rs-422-tutorial-basics.php](http://www.radio-electronics.com/info/telecommunications_networks/rs422/rs-422-tutorial-basics.php).
- [14] U-Blox, «LEA-4T ANTARIS 4 Programmable GPS Module whith Precision Timing Data Sheet».
- [15] HP - Hewlet Packard, «High-linearity Linearity Analog Optocouplers technical data HCNR200 HCNR201».
- [16] CAN in Automation (CiA), «Controller Area Network,» ottobre 2013. [Online]. Available: <http://www.can-cia.org/index.php?id=can>.

- [17] OLIMEX, «STM32-H107 development board,» Giugno 2011.
- [18] Sensor Technics, «HCLA Series - Miniature amplified low pressure sensors Data Sheet».
- [19] Sensor Technics, «HCA-BARO Series - Miniature amplified barometric pressure sensors Data Sheet».
- [20] RS component, «Design Spark,» Ottobre 2013. [Online]. Available: <http://www.designspark.com/>.
- [21] G. D. Sirio, «ChibiOS/RT Homepage,» Ottobre 2013. [Online]. Available: <http://www.chibios.org/dokuwiki/doku.php?id=start>.
- [22] Real Time Engineers Ltd, «FreeRTOS,» Ottobre 2013. [Online]. Available: <http://www.freertos.org/>.