

POLITECNICO DI TORINO

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

Tesi di Laurea Specialistica

**Sviluppo di un Modulo Riconfigurabile
integrato per Nano Satelliti**



Relatori:

Prof. Leonardo Reyneri

Prof. Dante Del Corso

Prof. Claudio Sansoè

Candidata:

Elena Ruo Rui

Novembre 2012

Indice

Sommario	5
1. Introduzione	8
1.1. Politecnico di Torino e Nano Satelliti.....	8
1.2. Standard AraMiS e Specifiche.....	8
1.3. Ambiente operativo.....	12
1.3.1. Radiazioni.....	12
1.3.2. Temperatura.....	14
1.4. Il linguaggio UML	15
1.4.1. Diagramma dei casi d'uso.	16
1.4.2. Diagramma delle classi.	18
2. Use Case Diagram: 1B81_Power_Management_Honeycomb	20
2.1. Actors	20
2.2. Use Cases	22
3. Class Diagram: 1B81 Power Management Honeycomb.....	28
3.1. 1B111C_Solar_Panel_HoneyComb	31
3.1.1. Attributes	31
3.1.2. Operations	31
3.2. 1B1121C_Primary_Switching_Buck	31
3.2.1. Operations	32
3.3. Bk1B235_Simple_Sun_Sensor	34
3.3.1. Attributes	34
3.3.2. Operations	34
3.4. Bk1B133A_Temperature_Sensor.....	34
3.4.1. Attributes	34
3.5. Test_Bk1B81S_Power_Management_Honeycomb	35
3.5.1. Attributes	35
3.5.2. Operations	35
3.6. 1B81_Power_Management_Honeycomb - Progetto Mentor Graphics.....	36
4. 1B111C Solar Panel – HoneyComb	41
4.1. 1B111C Assembly Instructions	42
4.1.1. 1B1113C PCB.....	42
4.1.2. 1B1112 EMI filter.....	42
4.1.3. 1B1111 Solar Cell.....	42
4.1.4. CV15-2500 Optical Resin.....	43
4.1.5. CF1-135 Primer	43
4.1.6. SP-120 Primer.....	43
4.1.7. CV2-2646 Resin	44
4.1.8. UPR10E3 Diode.....	44
4.2. Analisi delle Celle solari	45
4.2.1. Modello matematico delle celle solari.....	46
4.2.2. Analisi e modello Spice	46
4.2.3. Scelta del numero di celle solari.....	48
4.2.4. 14 celle in serie	50
4.2.5. 7 gruppi in serie di 2 celle in parallelo.....	53
4.2.6. 2 rami di 7 celle in serie.....	56
4.3. Configurazione definitiva.....	60
4.4. Presenza di guasti	61
4.5. 1B111C_Solar_Panel_Honeycomb - Progetto con Mentor Graphics	62

5.	1B1121C_Primary_Switching_Buck	64
5.1.	1B1121C_MPPT.....	66
5.1.1.	Attributes	66
5.1.2.	Operations	66
5.1.3.	Descrizione generale MPPT	68
5.1.4.	Prototipi Maximum Power Point Tracker	70
5.2.	1B1121C_BUCK.....	77
5.2.1.	Attributes	77
5.2.2.	Operations	77
5.2.3.	Modello matematico convertitore Buck.....	78
5.2.4.	Progetto convertitore Buck	85
5.3.	1B1121C_PmosDriver	88
5.3.1.	Attributes	88
5.3.2.	Operations	88
5.3.3.	Progetto P-mos driver	89
5.4.	1B1121C_CompHyst.....	92
5.4.1.	Attributes	92
5.4.2.	Operations	92
5.4.3.	Modello matematico del comparatore di soglia	93
5.4.4.	Progetto comparatore di soglia.....	94
5.5.	Bk1B132E_LowSideCurrentSensor	96
5.5.1.	Attributes	96
5.5.2.	Operations	96
5.5.3.	Progetto sensore di corrente low side.....	97
5.6.	Bk1B131D_Voltage_Sensor	98
5.6.1.	Attributes	98
5.6.2.	Operations	98
5.7.	1B1121C_PWMtoVref.....	98
5.7.1.	Attributes	98
5.7.2.	Operations	98
5.7.3.	Progetto PVM to Vref.....	99
5.8.	1B1121C_LowPowerReg.....	100
5.8.1.	Attributes	100
5.8.2.	Operations	100
5.9.	Bk1B132C_Current_Sensor	101
5.9.1.	Attributes	101
5.10.	1B1121C_3VRef.....	101
5.10.1.	Attributes.....	101
5.10.2.	Operations	101
5.11.	1B1121C_3V3Power	102
5.11.1.	Attributes.....	102
5.11.2.	Operations	102
5.12.	1B1121C_5VPower	102
5.12.1.	Attributes.....	103
5.12.2.	Operations	103
5.13.	Bk1B131C_Voltage_Sensor.....	103
5.13.1.	Attributes.....	103
6.	1B81 Software.....	104
6.1.	Test_Bk1B81S_Power_Management_Honeycomb	105

6.1.1. Attributes	105
6.1.2. Operations	105
6.2. Testmain_Bk1B81S_Power_Management_Honeycomb.....	105
6.2.1. Attributes	105
6.2.2. Operations	105
6.3. Bk1B1121CS_Primary_Switching_Buck.....	106
6.3.1. Attributes	106
6.3.2. Operations	108
6.3.3. Template parameters	114
6.4. Bk1B235S_Simple_Sun_Sensor.....	115
6.4.1. Operations	115
6.5. Bk1B133S_Temperature_Sensor	115
6.5.1. Operations	115
6.6. HK_fields_1B8_Honeycomb.....	116
6.6.1. Attributes	116
APPENDICI.....	118
A. Layout 1B81_Solar_panel_honeycomb	118
1. PCB esterno.....	118
3. PCB interno.	119
B. Schemi elettrici	120
1. 1B81_power_management_honeycomb.....	120
2. 1B4222_Tile_Processor_8M_V1.	121
3. Power_interface.....	122
4. 1B133A_Temperature_Sensor_V1.....	123
5. 1B1121C_Primary_Switching_Buck.	124
6. MPPT.....	125
7. Pmos_Driver_Block.	126
8. Hysteretic_Comparator.	127
9. Low_power_regulator.	128
10. PWM_to_Vref.....	129
11. Low_side_current_sensor.....	130
12. 1B131D_voltage_sensor.....	131
Bibliografia	139

Sommario

Recentemente è cresciuta l'attenzione alla progettazione di satelliti di piccole dimensioni da parte delle università italiane. In particolare il Politecnico di Torino ha adottato alcuni standard e si è anch'esso impegnato allo sviluppo di un proprio satellite.

Il progetto AraMiS (Architettura Modulare per Satelliti), in particolare, costituisce essenzialmente la prosecuzione ed il miglioramento del precedente progetto PiCPoT, prevede cioè lo sviluppo di nano e pico satelliti orbitanti in LEO (Low Earth Orbit) in ambito universitario.

In particolare il progetto AraMiS, oltre agli obiettivi funzionali già presenti nel precedente satellite, mira a realizzare un satellite modulare ri-configurabile in diverse forme, nel quale ogni scheda è progettata in modo indipendente dalle altre, ed allo stesso tempo è in grado di funzionare anche senza le altre. La comunicazione tra i vari elementi, detti tiles avviene per mezzo di opportuni bus. La peculiarità di questo progetto è appunto la modularità, ossia i moduli realizzati hanno ruoli specifici e sono completamente indipendenti, ma in grado, nello stesso tempo, di lavorare con altri moduli dello stesso tipo o diversi, per aumentare le prestazioni complessive del sistema. Il progetto viene considerato low-cost, in quanto utilizza componenti COTS, e sfrutta la riutilizzabilità dei vari moduli realizzati, riducendo appunto i costi di progetto.

Oggetto di questa tesi è la realizzazione di un modulo AraMiS di power management, partendo dall'analisi di come questo è stato realizzato nel precedente progetto PiCpoT, e procedendo con la progettazione, fino alla realizzazione di un prototipo su circuito stampato, che soddisfi le nuove specifiche.

In particolare questa tesi descrive la realizzazione di un modulo di dimensioni 165 x 330 mm, composto da due circuiti stampati, fissati su una struttura centrale di honeycomb in grado di ospitare da un lato il maggior numero di celle solari, e dall'altro il dispositivo MPPT per generare le alimentazioni per il satellite, due processori ed il maggior numero di connettori per l'aggiunta di eventuali moduli esterni. In particolare lo stampato esterno avrà le celle solari, mentre quello

interno, presenterà il microprocessore, i connettori, alcuni sensori e tutta la circuiteria necessaria a convertire l'energia proveniente dalle celle solari in modo adeguato, per poi dividerla con il resto del sistema.

Nella fase iniziale sono state analizzate le celle solari ed è stato ricavato e studiato il loro modello matematico, in modo da poter effettuare alcune simulazioni. Successivamente per mezzo di simulatori software, valutando le varie alternative, è stata scelta la configurazione migliore per le celle solari da utilizzare ed anche il numero di queste, conformemente con le dimensioni del modulo. Sono inoltre state introdotte opportune accortezze per quanto riguarda la ridondanza, necessaria a prevenire i problemi dovuti alle radiazioni presenti nello spazio.

A questo punto, scelta la configurazione di celle solari, e note le curve di tensione e corrente ricavabili da queste, è iniziata la progettazione del prototipo del power supply alloggiato nel PCB interno. Questo prototipo è basato su un convertitore switching isteretico, in grado di funzionare sempre nel punto di massima efficienza, chiamato appunto MPPT (Maximum Power Point Tracker), trasformando l'energia proveniente dalle celle solari in una compatibile al resto del sistema ed anche in grado di caricare le batterie per mantenere attivo il sistema in caso di periodi di "buio".

Sono state studiate diverse ipotesi per il progetto del convertitore switching ed i componenti necessari a realizzare l'MPPT (Maximum Power Point Tracker), e dopo varie simulazioni è stato utilizzato un convertitore Buck isteretico con un opportuno circuito composto da un comparatore ed alcuni transistori per realizzare il driver ed il controllo del punto di lavoro.

Sullo stesso PCB del convertitore, sono stati inoltre inseriti due microprocessori, i sensori necessari al corretto funzionamento e per la raccolta dati, ed infine il maggior numero possibile di connettori per altri moduli plug&play.

Il seguente elaborato è organizzato come segue:

Capitolo 1 – Introduzione generale al progetto realizzato in questa tesi, con descrizione dello standard AraMiS e dei vari progetti intrapresi dal Politecnico di Torino, studio delle condizioni di lavoro e descrizione dei tool software utilizzati.

Capitolo 2 – Descrizione del progetto principale chiamato 1B81 Power Management Honeycomb tramite diagramma dei casi d'uso: verrà descritto come il sistema progettato in questa tesi potrà e dovrà essere utilizzato, e verranno elencate le sue caratteristiche e specifiche.

Capitolo 3 – Descrizione tramite diagramma delle classi del progetto 1B81 Power management Honeycomb e dei suoi sottosistemi.

Capitolo 4 – Descrizione della parte relativa all'acquisizione dell'energia solare tramite celle solari, chiamata 1B111C Solar Panel Honeycomb, che comprende uno studio matematico e relative simulazioni delle celle solari.

Capitolo 5 – Descrizione della parte relativa alla gestione e conversione della potenza, chiamata 1B1121C Primary Switching Buck. In questo capitolo verrà trattato lo studio e la realizzazione di un dispositivo MPPT in grado di utilizzare l'energia proveniente dalle celle solari e di trasformarla in alimentazione per il resto del sistema. Verranno progettate ed analizzate tutte le varie parti che compongono l'MPPT realizzato con le necessarie simulazioni ed i relativi conti matematici.

Capitolo 6 – Descrizione del software di controllo utilizzato per il corretto funzionamento del convertitore MPPT mediante linguaggio C.

Appendici – Riportano gli schemi elettrici ed i layout realizzati con Mentor Graphics delle schede progettate per il satellite.

1.Introduzione

1.1. Politecnico di Torino e Nano Satelliti

Negli ultimi anni, il mercato dell'aeronautica spaziale, e quindi di quella satellitare, ha subito una notevole crescita, dovuta principalmente all'abbassamento dei costi dei lanciatori e dei lanci utili a portare i satelliti in orbita. Questo ha fatto sì che il Politecnico, come anche molte altre università in tutto il mondo, ma anche numerose aziende operanti nel settore, prendessero interesse nella realizzazione di piccoli satelliti di costo ridotto. Sono nati così diversi standard per piccoli satelliti, come CUBESAT e ARAMIS, in modo da fissare specifiche e metodologie di progetto in grado di facilitare la progettazione di questi piccoli satelliti. Un'altra interessante peculiarità di questi progetti universitari è relativa all'utilizzo di componenti COTS, ossia Commercial Off-The-Shelf, cioè componenti a basso costo, facilmente reperibili. Il lavoro di questa tesi si è focalizzato sulla realizzazione di un modulo riconfigurabile per un satellite universitario con standard AraMiS.

1.2. Standard AraMiS e Specifiche

AraMiS, acronimo di Architettura Modulare per Satelliti [1], è un progetto intrapreso dal Politecnico di Torino a partire dal 2006, che ha come scopo la realizzazione di satelliti di piccole dimensioni con struttura modulare e serve ad ampliare il precedente progetto chiamato CubeSAT che era invece costruito ad hoc. I satelliti di piccole dimensioni, sono generalmente divisi in categorie; in particolare esistono micro-satelliti con massa compresa tra 10 e 100 Kg, nano-satelliti con massa compresa tra 1 e 10 Kg, e infine pico-satelliti con masse inferiori ad 1 Kg.

L'architettura base di un satellite AraMiS, che rientra nella categoria dei nano-satelliti, si basa su una o più mattonelle, disposte sulla parte esterna del satellite, queste hanno quindi anche una funzione strutturale meccanica. Componendo

insieme più mattonelle, è possibile creare un satellite a forma cubica o prismatica, con dimensioni variabili.

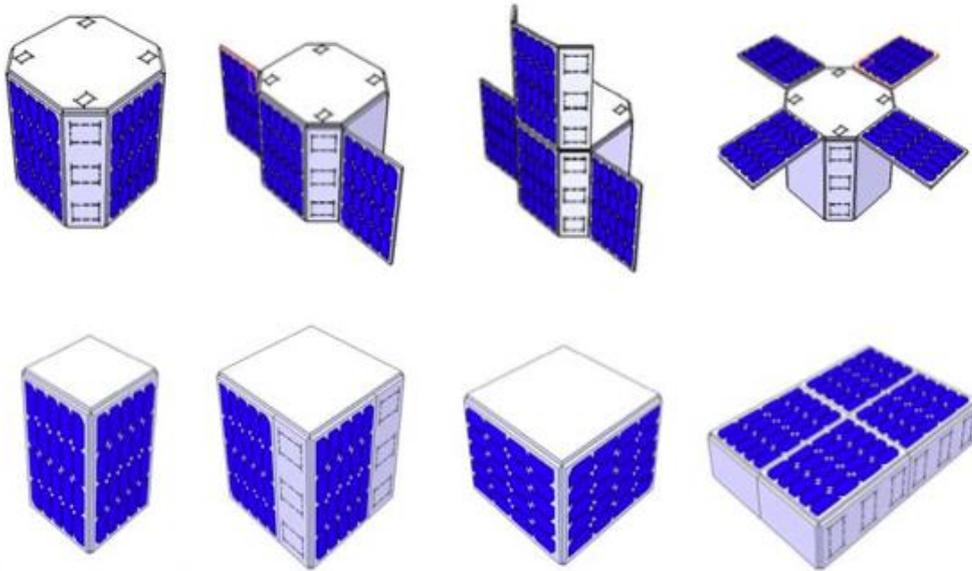


Figura 1: Esempi di possibili configurazioni di un satellite AraMiS

I moduli standard che costituiscono AraMiS sono:

- **Power Management Tile:** servono a generare, controllare, ed immagazzinare l'energia necessaria ad alimentare tutto il sistema. Convenzionalmente vengono utilizzati dei pannelli solari che per mezzo di opportuni circuiti alimentano il sistema e caricano le batterie. Queste ultime verranno utilizzate come riserva di energia, nel caso in cui quella proveniente dal sole non fosse sufficiente. La struttura meccanica del satellite è realizzata principalmente dall'unione di più tiles di questo tipo. Questi tiles sono gestiti da microcontrollori in grado di comunicare per mezzo di bus con l'OBC (On Board Computer).
- **Telecommunication Tile** hanno la funzione di mettere in comunicazione il satellite con la stazione di terra, gestisce inoltre le informazioni relative al controllo d'assetto provenienti dalla terra, e le trasferisce ai moduli di Power Management per farle attuare.
- **On Board Computer** è il modulo che contiene il processore in grado di coordinare il funzionamento di tutto il satellite. Effettua ad esempio i calcoli

relativi al controllo di assetto, e comanda gli altri tiles in modo che questi vengano attuati.

I moduli o tile, che compongono questa architettura presentano delle caratteristiche comuni. In particolare in progetti precedenti sono stati utilizzati moduli di dimensione 16,4 X 16,5 cm realizzati su lamine d'acciaio opportunamente trattate, come ad esempio quello mostrato in figura 2, con la previsione di affiancare questi moduli in modo da realizzare una struttura di dimensioni maggiori [2].

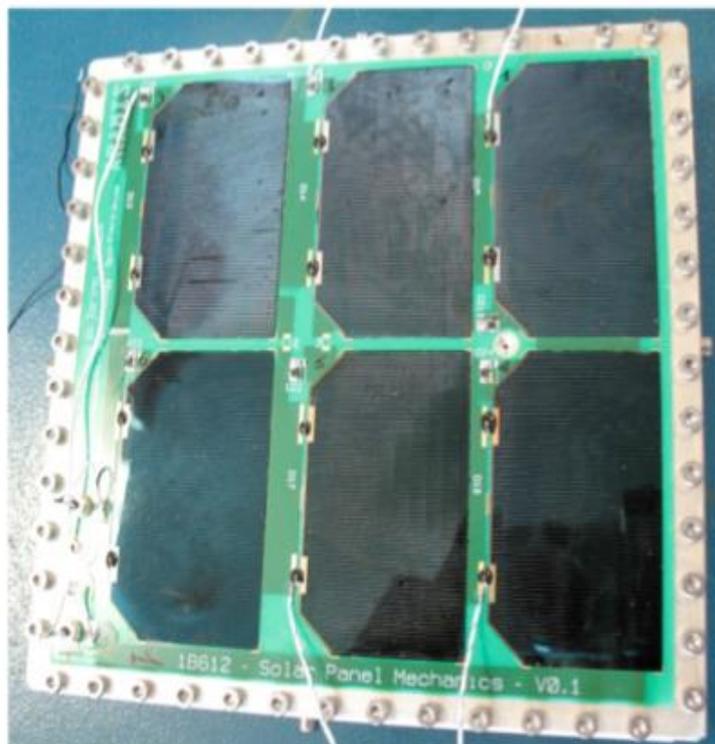


Figura 2 : Modulo di dimensioni 164x165 mm con celle solari

Nel progetto descritto in questa tesi, invece, verrà realizzato un tipo di tile leggermente diverso da quello precedentemente realizzato, in particolare si avranno dimensioni maggiori e cambierà la struttura portante:

- Dimensione di 16, 5 X 33 cm (pari al doppio delle strutture precedentemente realizzate);
- struttura portante realizzata in honeycomb, un particolare materiale molto resistente ma anche molto leggero; sulla quale verranno incollati i due PCB disegnati.

- fori di passaggio sull'intero perimetro con passo di 11,75 mm, per permettere l'assemblaggio del satellite grazie ad opportune strutture meccaniche.

Una delle peculiarità di questo progetto è la suddivisione in blocchi funzionali, ognuno indipendente dagli altri. La modularità permette di ottenere un'importante riduzione dei costi di progetto, che rappresentano uno dei principali ostacoli quando si parla di missioni spaziali; un altro aiuto per quanto riguarda i costi è dovuto al fatto che le varie unità vengono realizzate, come già accennato, con componenti COTS. Inoltre siccome i vari moduli vengono progettati in modo da funzionare indipendentemente dal resto del sistema, possono quindi essere riutilizzati in diverse missioni.

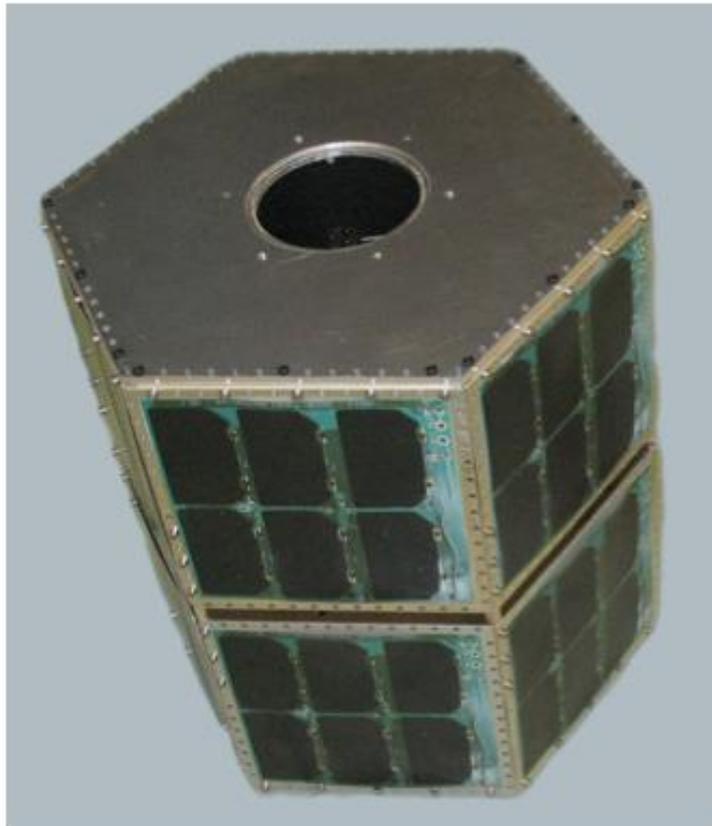


Figura 3: Esempio di struttura a Parallelepipedo

In questa tesi è descritta la realizzazione di un modulo composto da una parte interna ed una esterna al satellite, accuratamente fissate tra di loro per mezzo di una struttura in honeycomb per garantire il sostegno meccanico necessario. La parte esterna del modulo risulterà essere anche esterna rispetto al satellite, perciò

sarà quella nella quale verranno alloggiare le celle solari; la parte interna invece presenterà un microprocessore, il circuito per utilizzare l'energia proveniente dalle celle solari, i sensori necessari alla regolazione, i connettori per condividere l'energia con gli altri moduli, ed inoltre quattordici appositi connettori per poter successivamente ampliare le funzionalità del satellite, aggiungendo dei moduli ad-hoc.

1.3. Ambiente operativo

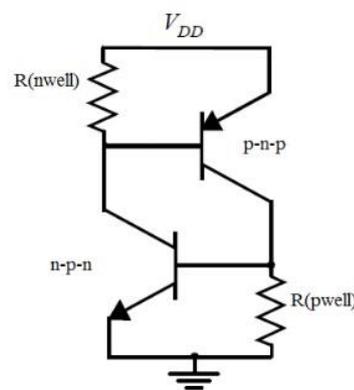
Prima di iniziare lo studio dei sistemi da realizzare e la progettazione ad esso correlata è bene studiare brevemente le condizioni atmosferiche nelle quali verrà a trovarsi il satellite una volta lanciato in orbita. Innanzitutto è da considerare che i satelliti AraMiS sono progettati per funzionare in quote comprese tra i 600Km e gli 800Km, all'interno di orbite dette LEO (Low Earth Orbit), ossia in una fascia compresa tra l'atmosfera terrestre e le fasce di Van Allen. Se si andasse oltre i 2000Km di distanza dall'atmosfera terrestre aumenterebbero a dismisura le radiazioni, e questo comprometterebbe il funzionamento del satellite.

1.3.1. Radiazioni

Nonostante la vicinanza alla terra, sono comunque presenti radiazioni, per questo motivo quando sarà possibile, verranno utilizzati componenti precedentemente testati e dei quali si conosce il comportamento in caso di radiazioni.

In particolare bisognerà tener conto del fatto che possono verificarsi diversi tipi di eventi dovuti al passaggio di una singola particella che genera una variazione del funzionamento di uno o più dispositivi del circuito a cui appartengono. Ad esempio la ionizzazione in un semiconduttore può comportare la generazione di un impulso transitorio spurio di corrente: se questo fenomeno si presenta in zone critiche del circuito può diventare un bit-flip. Allo stesso modo l'interazione di una particella ionizzante con un dispositivo a semiconduttore può provocare la sua ionizzazione e quindi un eventuale cambiamento di stato logico.

Un altro importante accorgimento è legato alla tecnologia CMOS che oggi giorno costituisce la maggior parte dei dispositivi elettronici in commercio: i dispositivi CMOS sono soggetti a fenomeni di Latch-up che può venire facilmente innescato dalle radiazioni. Questo fenomeno una volta innescato si conclude, molto spesso, con la distruzione del dispositivo. Per descrivere brevemente il fenomeno del latch-up, possiamo dire che è un particolare tipo di corto circuito, o meglio, un cammino a bassa impedenza, che può formarsi tra le alimentazioni di un circuito MOSFET dove esistono strutture parassite, detto SCR. In particolare nelle tecnologie CMOS, per costruzione, vengono a formarsi numerose strutture n-p-n-p parassite che se attivate, portano ad un corto circuito tra la V_{DD} e GND con conseguente distruzione del chip. Un esempio di struttura parassita è mostrato nello schema seguente.



In questo caso quando uno dei due transistori parassiti viene polarizzato (a causa della corrente che scorre attraverso il well, o il substrato), alimenta la base dell'altro transistor; viene così a crearsi un feedback positivo che porta ad un veloce aumento della corrente finché il circuito smette di funzionare correttamente o peggio si brucia. In generale le tecnologie considerate radiation tolerant, e quindi utilizzabili senza particolari problemi nello spazio sono:

- Diodi (tranne quelli Zener).
- Logiche TTL Logiche ECL (Emitter Coupled Logic).
- Tecnologie GaAs (Arsenuro di Gallio).
- Dispositivi a microonde.
- Oscillatori a cristallo.
- La maggior parte dei dispositivi passivi.

Per questo, molto spesso sarà necessario progettare da zero dispositivi, altrimenti presenti in commercio con tecnologia CMOS, in modo da evitare problemi di latch-up.

Un ottimo sostituto alla tecnologia CMOS è ovviamente il BJT, che presenta però molto spesso qualità peggiori per quanto riguarda la potenza.

Possono essere considerate altre importanti prevenzioni, come circuiti LPT in grado di spegnere il chip nel caso si verifichi un fenomeno di latch-up (individuato con un anormale aumento della corrente), oppure possono essere utilizzate particolari tecnologie, ad esempio SOI, strutture basate su uno strato di ossido in grado di ridurre la formazione di transistori parassiti.

In generale esistono inoltre tecniche di prevenzione basate sulla ridondanza delle parti critiche, oppure tecniche software basate su controlli ciclici di ridondanza (CRC), oppure tecniche come quella del watchdog o anche tecniche SHIFT utili ad individuare la presenza di guasto momentaneo e per mezzo del software, correggere il risultato.

1.3.2. Temperatura

Un altro fattore importante da considerare è la temperatura. Infatti un sistema orbitante in LEO sarà sottoposto ad alte pressioni e a bassissime concentrazioni d'aria (ossia condizioni di quasi vuoto). Per questo il satellite dovrà funzionare a temperature bassissime, inoltre questo non servirà a risolvere i problemi di surriscaldamento in quanto la condizione di vuoto annulla quasi completamente la dissipazione di calore per convezione. Inoltre è da considerare che la distanza dal sole sarà molto minore, quindi il lato esposto al sole avrà temperature molto superiori a quello in ombra, e perciò minimi movimenti, che portano ad esporre una parte piuttosto che un'altra, porteranno ad aumenti o diminuzioni brusche delle temperature. Sono state ipotizzate quindi condizioni di lavoro all'interno del satellite che variano dai -30°C ai 20°C , che saranno quindi utilizzate come specifiche per la scelta di componenti adatti.

1.4. Il linguaggio UML

Per la progettazione del satellite AraMis si è scelto di utilizzare il linguaggio UML (Unified Modeling Language) [3]. Infatti questo linguaggio permette non solo di descrivere in modo accurato tutte le varie parti che compongono il satellite, ma permette anche il lavoro contemporaneo di più persone che possono quindi utilizzare materiale altrui e condividere il proprio. Questo linguaggio risulta quindi molto adatto per sviluppare i diversi moduli del satellite AraMiS in parallelo, da parte di studenti diversi.

Il modulo 1B81 ed i suoi due sotto-blocchi principali, studiati e progettati in questa tesi, sono stati analizzati e descritti per mezzo di questo linguaggio. E' quindi utile una breve descrizione di UML e della sua simbolistica per facilitare la comprensione dei capitoli successivi.

L'UML è un linguaggio di programmazione nato nel 1996 e definito dai così detti "tre amigos" (Grady Booch, Jima Rubaugh e Ivar Jacobson) unificando gli approcci allora presenti. UML consente di costruire modelli object-oriented per rappresentare o descrivere un sistema software o hardware ed il comportamento di questo, sia a scopo di analisi che di progetto, facilitando inoltre la collaborazione tra individui diversi che lavorano allo stesso progetto.

Il modello realizzabile in UML è strutturato secondo un insieme di viste che rappresentano i diversi aspetti dell'oggetto modellato (funzionamento, struttura, comportamento, etc), inoltre i tool di programmazione UML forniscono meccanismi automatici di traduzione dei diagrammi UML in codice e viceversa, è anche possibile esportare un report del progetto descritto in formato .doc o .pdf, questo report è stato utilizzato per la stesura della tesi stessa.

Il tool utilizzato per la creazione dei diagrammi e del progetto in UML è *Visual Paradigm for UML*[4] che permette, tra le altre cose di generare automaticamente i file sorgente .c e header .h che serviranno al microcontrollore per gestire ed utilizzare i blocchi descritti direttamente a partire dagli oggetti rappresentati nei vari diagrammi. Per la gestione del progetto software è stato utilizzato anche il tool

IAR che consente appunto la programmazione del microcontrollore MPS460 e il debug del software realizzato.

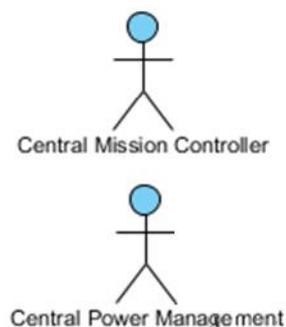
Il progetto esposto in questa tesi utilizza principalmente due tipi diversi di diagrammi UML: il *diagramma dei casi d'uso* ed il *diagramma delle classi*.

1.4.1. Diagramma dei casi d'uso.

Il diagramma dei casi d'uso descrive le funzionalità e le specifiche del progetto; definisce inoltre chi o cosa può utilizzarle. In generale questo tipo di modello rappresenta il punto di partenza nella modellizzazione del sistema.

Un diagramma dei casi d'uso è caratterizzato dalle seguenti categorie di elementi:

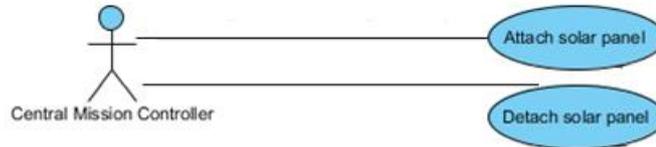
- L'**Attore** è ogni qualsivoglia entità (utente umano, sistema o ambiente esterno) che interagisce con il sistema descritto. Viene rappresentato con un omino stilizzato ed è collegato tramite una linea ad uno o più casi d'uso che esso può realizzare.



Ad esempio, in questo progetto i due attori principali sono il *Central Power Management* ed il *Central Mission Controller*; il primo è l'entità che si occupa del sistema nelle operazioni nominali (può ad esempio essere una routine software che sta girando nell'OBC), il secondo rappresenta l'entità in carica che si occupa di gestire e supervisionare il satellite, individuando i guasti e gestendo le emergenze.

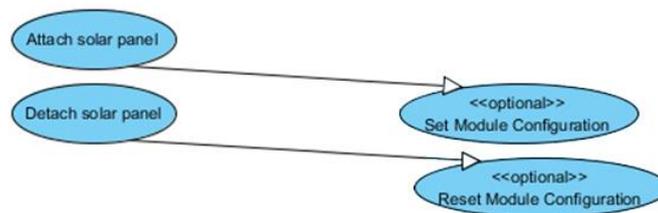
- I **Casi d'uso** sono i compiti che l'attore richiede al sistema. Tali compiti costituiscono gli obiettivi del sistema progettato e sono rappresentati

graficamente tramite delle ellissi blu all'interno delle quali è scritta in breve la funzione realizzata dal sistema.



I casi d'uso possono a loro volta relazionarsi con altri casi d'uso.

- **Associazioni** tra attore e caso d'uso sono realizzate con linee rette (come si può vedere nel diagramma precedente) ed indicano appunto quale attore può utilizzare determinati casi d'uso. Solitamente un attore viene associato a più casi d'uso ed inoltre un caso d'uso può ovviamente essere associato a più di un attore.
- Le relazioni tra casi d'uso diversi possono essere di diverso tipo, ad esempio ricordiamo le *generalizzazioni*, le *inclusioni* e le *estensioni*. In questo progetto sono utilizzate perlopiù associazioni semplici oppure le generalizzazioni come mostrato nella figura sottostante.



Infatti i casi d'uso descritti ed utilizzati in questo progetto sono perlopiù generalizzazioni dei casi d'uso:

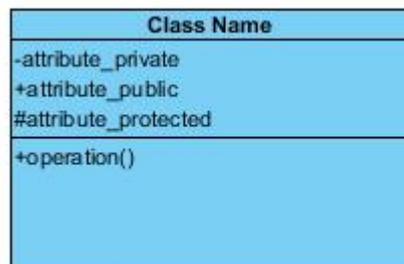
- “*set module configuration*”, “*reset module configuration*” e “*get module configuration*” che comprendono i casi d'uso che controllano e impostano o meno la presenza delle varie parti del sistema.
- “*get module housekeeping*” che comprende tutti i casi d'uso delle varie acquisizioni dei valori utili alle varie operazioni che deve effettuare l'MPPT

- “*module commands*” che comprende tutti i vari comandi che l'MPPT è in grado di risolvere.

1.4.2. Diagramma delle classi.

I *diagrammi delle classi* sono composti da numerosi oggetti e da associazioni tra questi. Questo tipo di diagramma viene utilizzato per descrivere in modo preciso il sistema progettato, descrivendo in particolare tutte le sue componenti sia software che hardware. Un diagramma delle classi riporta quindi una raffigurazione grafica dei vari oggetti che compongono il sistema, rappresentando inoltre la gerarchia e le connessioni che legano i vari oggetti.

Con *classe* si intende la generalizzazione di un oggetto che ne costituisce un'istanza specifica. Una classe viene rappresentata da un rettangolo suddiviso solitamente in tre campi distinti nei quali sono descritti gli elementi caratteristici e fondamentali dell'oggetto stesso, come mostrato in figura.



I tre campi che caratterizzano una classe sono:

- *Nome*, che appunto identifica una classe specifica ed univoca; occupa il primo riquadro.
- *Attributi*, che descrivono le proprietà dell'oggetto sia logiche che fisiche; compaiono nel primo riquadro subito dopo il nome della classe.

Se consideriamo ad esempio un sensore, possiamo notare che come attributi esso avrà la sensibilità, il consumo di potenza, etc. Nel caso di classi software gli attributi sono le variabili, le strutture e i define del linguaggio C.

Ogni attributo è usualmente caratterizzato da un tipo (ad esempio int in caso di intero), un valore iniziale, l'accessibilità (che definisce chi può avere accesso a quell'attributo), dalla visibilità e da altre proprietà più specifiche. Gli attributi possono inoltre essere utilizzati per collegare, o meglio, associare classi diverse, in modo che queste possano servirsi anche di attributi altrui.

- *Metodi* che indicano quali operazioni può compiere l'oggetto, sono descritti nell'ultimo riquadro, dopo quello degli attributi.

In caso di oggetto hardware i metodi equivalgono ai collegamenti di segnale che l'oggetto ha verso altri oggetti, mentre per un oggetto software equivalgono alle funzioni C che l'oggetto è in grado di realizzare. I metodi possono avere valori di ritorno che vengono restituiti al chiamante.

In questa tesi il sistema progettato è descritto principalmente grazie a diagrammi delle classi, mostrando in modo semplice ed intuitivo la suddivisione del sistema in blocchi e sotto-blocchi che sono di tipo hardware (oggetti colorati in giallo) e/o software (oggetti colorati in verde).

Come già accennato in precedenza, la descrizione del progetto realizzato avverrà generalmente mediante diagrammi di classe che comprendono oggetti ed associazioni di questi. In ciascuno dei capitoli e sotto-capitoli seguenti verranno descritti in modo dettagliato attributi e metodi degli oggetti utilizzati e realizzati mediante la relativa documentazione UML.

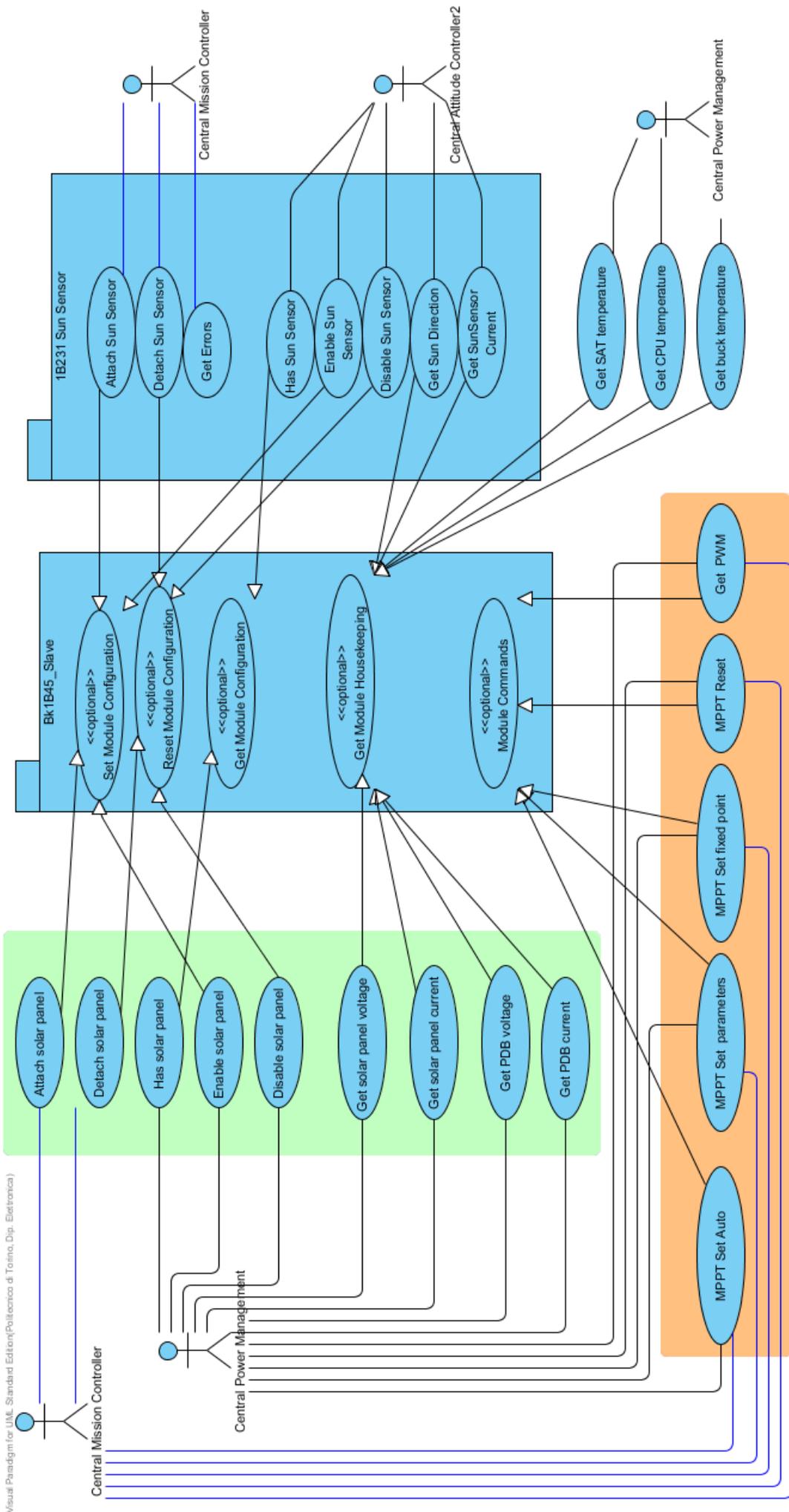
Per ogni nuovo diagramma verranno descritti tutti gli oggetti in esso contenuti purchè abbiano importanza per quanto riguarda questo progetto. Questa descrizione dettagliata è stata possibile utilizzando il report generato dal tool di sviluppo UML utilizzato, ossia Visual Paradigm.

2. Use Case Diagram: 1B81_Power_Management_Honeycomb

Nel diagramma presente nella pagina successiva sono descritti tutti i casi d'uso del progetto realizzato in UML. Questi casi d'uso descrivono appunto tutti i modi in cui il sistema può essere utilizzato ma anche tutti i compiti che è in grado di svolgere. Fissa inoltre gli attori, ossia chi può effettuare determinate operazioni sul sistema. In particolare i casi d'uso circondati dai quadrati arancione e verde sono dei casi d'uso direttamente progettati in questa tesi, mentre gli altri fanno parte di progetti esterni e sono quindi stati inclusi in questo progetto.

2.1. Actors

Name	Documentation
Central Power Management	The entity (likely a SW routine running on the OBC in charge of managing the 1B1 Power Management Subsystem in nominal operation. Fault and emergency handling are left to the Central Mission Controller.
Central Attitude Controller2	The entity (likely a SW routine running on the OBC in charge of managing the 1B2 Attitude Control Subsystem in nominal operation. Fault and emergency handling are left to the Central Mission Controller.
Central Mission Controller	The entity (likely a SW routine running on the OBC) in charge of satellite supervision, fault detection and management and emergency management.



2.2. Use Cases

Name	Documentation
Get Errors	
Detach Sun Sensor	
Attach Sun Sensor	
Has Sun Sensor	
Get Sun Sensor Current	
Enable Sun Sensor	
Disable Sun Sensor	
Get Sun Direction	<p>Returns the last measured direction of sun; two numbers, 16-bits each, giving sun direction along X and Y axis. It can be read using getHousekeeping method of 1B45 package.</p> <ul style="list-style-type: none"> Value housekeeping[SUN_SENSOR_X] returns the angle between sun direction and the y-z plane, in units of in 180/2048 degrees (= pi/2048 radians); positive value indicates that sun is towards positive x-axis. In case sun is not visible, it returns -32768. Value housekeeping[SUN_SENSOR_Y] returns the angle between sun direction and the x-z plane, in units of in 180/2048 degrees (= pi/2048 radians); positive value indicates that sun is towards positive y-axis. In case sun is not visible, it returns -32768. <p>The (model element not found) will read magnetic field components as variable t_sensor housekeeping using the Get Module Housekeeping use case of Bk1B45_Slave package.</p>
Get SAT temperature	This command allows to get the temperature inside the satellite, far away from power electronics
Get CPU temperature	This command allows to get the temperature near the CPUs.
Get buck temperature	This command allows to get the temperature near the buck converter.
Get PWM	The actual PWM used by the MPPT can be easily

	checked by taking his value inside HK vector at MPPT_PWM location.
Get Module Housekeeping	<p>Returns last measured housekeeping data (see use case Module Housekeeping for details).</p> <p>It uses the Read Data use case by issuing the CMD_GET_HOUSEKEEPING command.</p> <p>This use case is optional, therefore the (model element not found) shall activate it, in the global header file platform.h, by #define an identifier whose name is contained in the tagged value define of this use case.</p>
MPPT Set Auto	<p>This use case enable the automatic tracking of maximum power point. This means that the working point of the MPPT is calculated by the CPU after reading the sensors. This is the opposite use case of MPPT Set fixed point</p>
Module Commands	<p>The Master actor sends a data-less command to the System.</p> <p>The Master can use up to 8 different Module Commands, to issue as many commands to the system.</p> <p>It uses the Command Only use case, by issuing the CMD_COMMAND_0 through CMD_COMMAND_7, where x (0..7) identifies the message type; x does not identify the sequence in which commands are issued, but the (model element not found)-defined type of command.</p> <p>The (model element not found) can use as many message types he wants.</p> <p>This use case is optional, therefore the (model element not found) shall #define an identifier whose name is contained in the tagged value define.</p> <p>After command has been received and checked, the 1B45 package calls the CommandInterpreter.interpret operation and passes the command as an argument.</p> <p>The (model element not found) shall then write his command interpretation routine inside the interpret operation.</p>
Set Module Configuration	<p>Sets to 1 some bits of the internal configuration word (configRegister) of one of many spacecraft Tiles.</p> <p>The OBC shall send to the Tile as many bits as are in its configuration word configRegister. Each bit which is 1 in the message will set to 1 the corresponding bit in the Tile configuration word. The other bits are left</p>

	<p>unchanged.</p> <p>This use case sends configuration bits by means of the Write Data use case, by issuing the <code>CMD_SET_CONFIGURATION</code> command with the address of the desired Tile.</p> <p>This use case differs from a set of related use cases; in particular:</p> <ul style="list-style-type: none"> • the use case Configure Module allows a compile-time configuration. • the use case Reset Module Configuration resets to 0 a few bits in the configuration word. It cannot set any bit to 1. • the use case Set Module Configuration sets to 1 a few bits in the configuration word. It cannot reset any bit to 0. • the use case Write Module Configuration writes (either 0 or 1) all the bits of the configuration word. <p>This use case is optional, therefore the Configurator shall <code>#define</code> inside file <code>platform.h</code> an identifier whose name is contained in the tagged value <code>define</code>.</p>
<p>Reset Module Configuration</p>	<p>Resets to 0 some bits of the internal configuration word (<code>configRegister</code>) of one of many spacecraft Tiles.</p> <p>The OBC shall send to the Tile as many bits as are in its configuration word <code>configRegister</code>. Each bit which is 1 in the message will reset to zero the corresponding bit in the Tile configuration word. The other bits are left unchanged.</p> <p>This use case sends configuration bits by means of the Write Data use case, by issuing the <code>CMD_RESET_CONFIGURATION</code> command with the address of the desired Tile.</p> <p>This use case differs from a set of related use cases; in particular:</p> <ul style="list-style-type: none"> • the use case Configure Module allows a compile-time configuration. • the use case Reset Module Configuration resets

	<p>to 0 a few bits in the configuration word. It cannot set any bit to 1.</p> <ul style="list-style-type: none"> • the use case Set Module Configuration sets to 1 a few bits in the configuration word. It cannot reset any bit to 0. • the use case Write Module Configuration writes (either 0 or 1) all the bits of the configuration word. <p>This use case is optional, therefore the Configurator shall #define inside file platform.h an identifier whose name is contained in the tagged value define.</p>
<p>Get Module Configuration</p>	<p>It returns to the OBC the current configuration word (configRegister) of one of many spacecraft Tiles. The configuration word was last written by the OBC, or the default for the Tile if not yet modified by the OBC.</p> <p>This use case first uses the Read Data use case by issuing the CMD_GET_CONFIGURATION command with the address of the desired Tile, which then returns its configRegister to the Master.</p> <p>This use case is optional, therefore the Configurator shall #define inside file platform.h an identifier whose name is contained in the tagged value define.</p>
<p>MPPT Set parameters</p>	<p>This command allows to set two different parameters according to the values selected by the Central Mission Controller or Central Power Management.</p> <p>This limits have to be written in a specific location that is pointed by the returned pointer of the buffer. Each limit uses two byte and it's a number between 0 and FFFF. In particular the syntax in the byte vector will be AAAABBBB where AAAA is the upper limit and BBBB is the lower limit.</p> <ul style="list-style-type: none"> • upper_MPPT_limit if this value is different from zero, it means that the function evaluate_PWM uses this value as upper limit. • lower_MPPT_limit if this value is different from zero, it means that the function evaluate_PWM uses this value as lower limit <p>To set new limits or change the actual ones it is necessary re-launch this command again.</p> <p>To unset those limits the MPPT Reset use case or</p>

	MPPT Set Auto use case have to be used.
MPPT Set fixed point	<p>This command fix a working point of MPPT according to the value selected by the Central Mission Controller or Central Power Management.</p> <p>This mean that the MPPT will not track automatically the maximum power point, but stands still at the fixed point.</p> <p>The fixed point is written in a specific location that is pointed by the returned pointer of the buffer, uses two byte and it's a number between 0 and FFFF.</p> <p>To set another fixed working point it is necessary to re-launch the command.</p>
MPPT Reset	<p>This command reset the software that evaluate the PWM. It clear all variables about power:</p> <ul style="list-style-type: none"> • present_power • old_power • ERR • fixed_PWM • lower_MPPT_limit • upper_MPPT_limit <p>It also set:</p> <ul style="list-style-type: none"> • MPPT_tracking_enable = true
Get PDB current	<p>This is the current generated by the 1B1121C_MPPT and measured by the sensor Bk1B132C_Current_Sensor.</p> <p>After the acquisition, the value of MPPT current can be found in I_PDB in the HK vector.</p>
Get PDB voltage	<p>This is the voltage PDB for the satellite generated by the 1B1121C_MPPT.</p> <p>It is measured by the sensor Bk1B131C_Voltage_Sensor and if all work correctly it has a value of 14V.</p> <p>After the acquisition, the value of this voltage can be found in V_PDB in the HK vector.</p>
Get solar panel current	<p>This is the current generated from the solar cells and measured by the sensor Bk1B132E_LowSideCurrentSensor.</p> <p>After the acquisition, the value of solar panel current can be found in I_SOLAR in the HK vector.</p>
Get solar panel voltage	<p>This is the voltage generated by the solar panel, and measured by the Bk1B131D_Voltage_Sensor.</p>

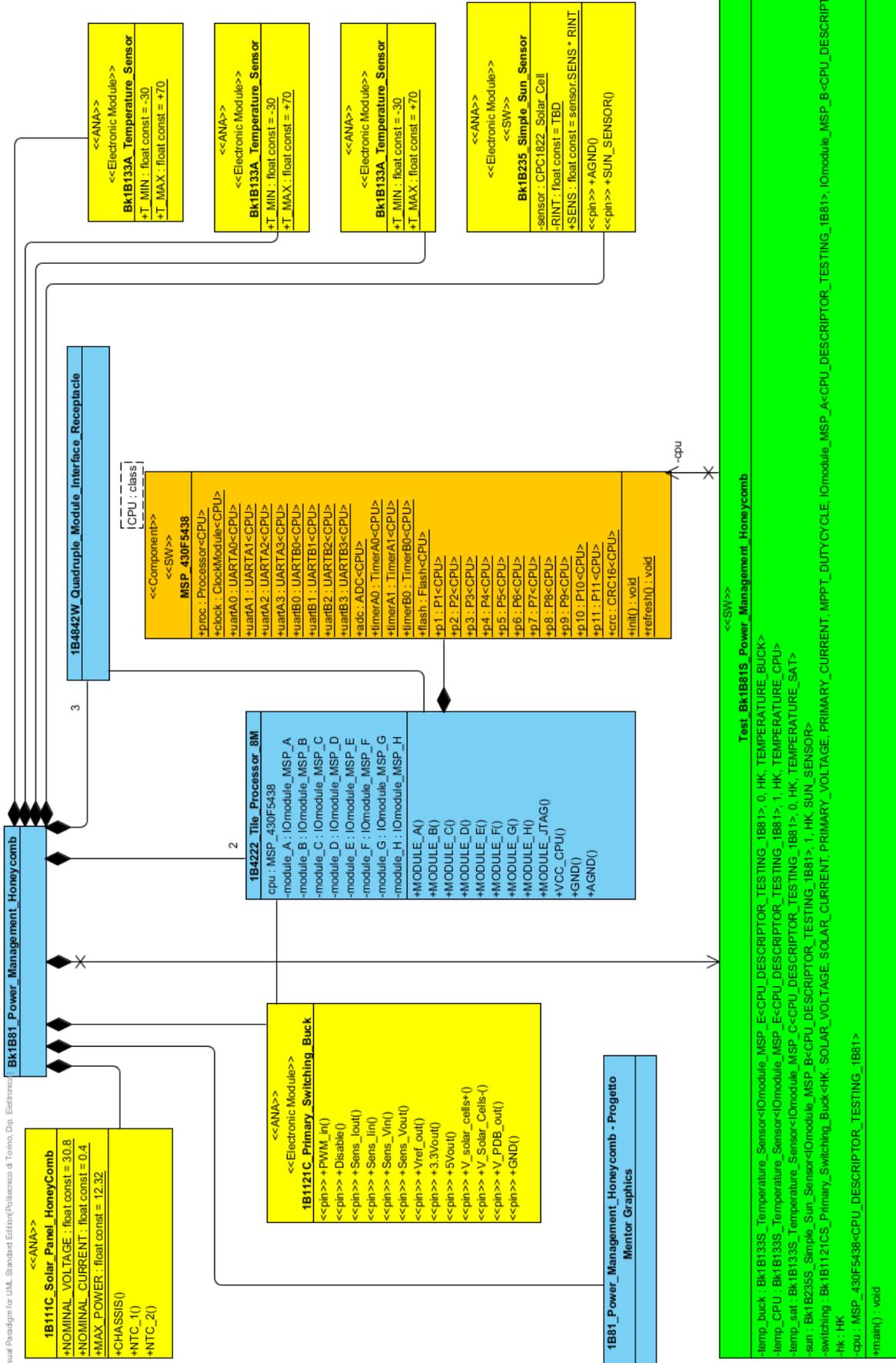
	<p>After the acquisition, the value of solar panel voltage can be found in V_SOLAR in the HK vector.</p>
Disable solar panel	<p>Disables solar generator. Any further action of it will be disregarded.</p> <p>If the generator is in process of converting power while the Disable solar panel use case is applied, the generator is turned off immediately.</p> <p>When disabled, the generator shall draw negligible power from the PDB.</p>
Enable solar panel	<p>Enables solar generator. Any further action of it will be carried on regularly.</p> <p>When enabled, the generator shall be active only when in the sunlight.</p> <p>When enabled but not active (no sunlight), power consumption should be as little as possible.</p>
Has solar panel	<p>Returns the presence of the solar generator in the system.</p>
Detach solar panel	<p>Flags solar generator as unavailable to the user.</p> <p>Generator is considered "detached" either if it is NOT physically present on the tile or it has not been attached (see Attach solar panel).</p> <p>If the generator is either detached or disabled, no further action of it can be carried on.</p> <p>The difference between Detach solar panel and Disable Solar Panel is that the former prevents the enable of the MPPT, and so the generation of the voltage for the entire satellite.</p> <p>The Detach solar panel use case is intended only in rare occasions, to permanently disable a faulty generator.</p>
Attach solar panel	<p>Flags solar generator as available to the user.</p> <p>Generator is considered "attached" if it is physically present on the tile and it has not been detached.</p> <p>If the solar generator is attached and enabled, any further action of it will be carried on.</p>

3. Class Diagram: 1B81 Power Management Honeycomb

This class diagram represent the Bk1B81_Power_Management_Honeycomb subsystem, which is a part of the AraMiS project involved in generating and optimizing power.

It is composed mainly by:

- Class 1B1121C_Primary_Switching_Buck contains all the electronics needed to create the various supply for the whole satellite and some sensor to control the behavior.
- Class 1B111C_Solar_Panel_HoneyComb represent the solar panel array composed by 14 solar cells and some other few electronics; that convert the energy coming from the sun to power usable to supply the satellite.
- Class Bk1B1121CS_Primary_Switching_Buck contains the software that realizes and creates the Maximum Power Point device.
- Class 1B4222_Tile_Processor_8M is composed by the CPU MSP_430F5438 and allows the communication between the CPUs and the connectors for external modules
- Class 1B4842W_Quadruple_Module_Interface_Receptacle represent the modules plug that are present on this project. In total there are 12 free connector for external modules.
- Various sensor described by class Bk1B133A_Temperature_Sensor and class Bk1B235S_Simple_Sun_Sensor



Il diagramma di classe mostrato in figura nella pagina precedente, rappresenta in modo generale il progetto che è stato sviluppato in questa tesi.

In particolare il blocco 1B81 Power Management Honeycomb è l'oggetto di più alto livello gerarchico ossia è l'unione della parte che acquisisce l'energia solare (1B111C_Solar_Panel_Honeycomb) e la parte che la converte in energia utilizzabile per il resto del satellite, in modo ottimale (1B1121C_Primary_Switching_Buck). Si nota inoltre che sono presenti altri oggetti non direttamente trattati in questa tesi, in quanto sono stati precedentemente progettati ed analizzati in altri elaborati da altri studenti, come ad esempio i moduli di interfaccia, i sensori ed i connettori che permettono di ampliare le funzionalità del sistema aggiungendo moduli esterni, o il microprocessore.

Come possiamo vedere in figura il sistema generale è composto da due microprocessori e tre blocchi di interfaccia, ognuno dei quali è composto da quattro connettori (in totale 12 connettori singoli disponibili). Infatti le specifiche di progetto richiedevano di utilizzare il maggior numero possibile di microcontrollori ed anche di connettori. In particolare il quarto blocco di connettori è stato sacrificato per poter utilizzare i sensori necessari al corretto funzionamento dell'MPPT.

La parte principale è quindi quella descritta nel progetto 1B1121C_Primary_Switching_Buck ossia quella che realizza il Maximum Power Point Tracker. E' composta principalmente dal blocco 1B1121C_MPPT che contiene al suo interno la descrizione di tutti i dispositivi hardware progettati.

Il blocco software colorato in verde (Bk1B1121CS_Primary_Switching_Buck), è quello nel quale viene descritto il software di controllo ed i vari sensori utili a misurare le tensioni e correnti del sistema ed a comandare correttamente il dispositivo MPPT progettato.

3.1. 1B111C_Solar_Panel_HoneyComb

3.1.1. Attributes

Signature	Documentation
<u>+NOMINAL VOLTAGE</u> : float const = 30.8	Voltage (V) at MPP with 1,360W/m2 at AM0
<u>+NOMINAL CURRENT</u> : float const = 0.4	Current (A) at MPP with 1,360W/m2 at AM0
<u>+MAX POWER : float</u> const = 12.32	Maximum Power Point (W) with 1,300W/m2 at AM0

3.1.2. Operations

Electrically connected to metallic chassis.

Signature: CHASSIS()

Connection to an external NTC (or any other external sensor).

Signature: NTC_1()

Connection to an external NTC (or any other external sensor).

Signature: NTC_2()

3.2. 1B1121C_Primary_Switching_Buck

This class represents the module of power conversion that generates the various supply for the entire satellite.

The parts involved in converting the power coming from solar cells are parts of the so-called 1B1121C_MPPT.

In particular will generated:

- 5V (1A) from 1B1121C_5VPower block.
- 3,3V (1,6A) from 1B1121C_3V3Power block.
- 3V reference, named REF, from 1B1121C_3VRef block.
- 14V named PDB from 1B1121C_MPPT block.

Other important devices that compose this class are the sensors, in particular in particular there are four different sensors:

- Block Bk1B131C_Voltage_Sensor measures a voltage with range from 0 to 20V.
- Block Bk1B131D_Voltage_Sensor measures a voltage with range from 0 to 40V.
- Block Bk1B132C_Current_Sensor measures a current that can be 2.5A maximum.
- Block Bk1B132E_LowSideCurrentSensor measures a "low side" current, which means current between a point and GND, using an operational amplifier.

The MPPT and his sub-blocks are controlled by a CPU MSP_430F5438 thanks to the PWM signal; also all the measurement done by sensors are sent to the processor.

3.2.1. Operations

This is the PWM pin.

The PWM (it is a Cycle value) has to be generated by the CPU and connected to this pin in order to drive the 1B1121C_MPPT.

Signature: PWM_in()

It is the signal that disables the 1B1121C_MPPT.

If the value it's 0, the attached block are enabled, else, disabled.

Signature: Disable()

This is the current value coming from Bk1B132C_Current_Sensor.

This current it's also the current generated by the 1B1121C_MPPT.

Signature: Sens_Iout()

This is the output current coming from Bk1B132E_LowSideCurrentSensor that measure the input current of the 1B1121C_Primary_Switching_Buck.

Signature: Sens_Iin()

This is the voltage coming from Bk1B131C_Voltage_Sensor that measure the input voltage generated by the solar cells.

Signature: Sens_Vin()

This is the output voltage coming from Bk1B131D_Voltage_Sensor that measure the output voltage generated by the 1B1121C_Primary_Switching_Buck.

Signature: Sens_Vout()

This is the Voltage reference (3V) generated by the
1B1121C_Primary_Switching_Buck.
Signature: Vref_out()

This is the 3.3V voltage generated by the 1B1121C_Primary_Switching_Buck.
Signature: 3.3Vout()

This is the 5V voltage generated by the 1B1121C_Primary_Switching_Buck.
Signature: 5Vout()

This signal is the positive voltage input of the 1B1121C_Primary_Switching_Buck
and comes from solar cells.
It is expected a maximum voltage around a value of 31V.
Signature: V_solar_cells+()

This signal is the negative voltage input of the 1B1121C_Primary_Switching_Buck
and comes from solar cells.
Signature: V_Solar_Cells-()

This signal is the PDB (14V) generated by the 1B1121C_Primary_Switching_Buck.
Signature: V_PDB_out()

Ground signal
Signature: GND()

3.3. Bk1B235_Simple_Sun_Sensor

3.3.1. Attributes

Signature	Documentation
<u>-sensor : CPC1822 Solar Cell</u>	
<u>-RINT : float const = TBD</u>	Value of internal sense resistor, in Ohm.
<u>+SENS : float const = sensor.SENS * RINT</u>	Output sensitivity of SUN_SENSOR output, in V/(W/m ²).

3.3.2. Operations

Signature: AGND()

Output voltage proportional to sun intensity.

Signature: SUN_SENSOR()

3.4. Bk1B133A_Temperature_Sensor

A specific implementation of a Bk1B133_Temperature_Sensor with temperature range -30 to +70oC.

The block has a temperature to voltage shown in referenced plot.

3.4.1. Attributes

Signature	Documentation
<u>+T MIN : float const = -30</u>	Minimum measurable temperature in °C.
<u>+T MAX : float const = +70</u>	Maximum measurable temperature in °C.

3.5. Test_Bk1B81S_Power_Management_Honeycomb

3.5.1. Attributes

Signature

Documentation

-temp_buck : Bk1B133S_Temperature_Sensor
-temp_CPU : Bk1B133S_Temperature_Sensor
-temp_sat : Bk1B133S_Temperature_Sensor
-sun : Bk1B235S_Simple_Sun_Sensor
-switching : Bk1B1121CS_Primary_Switching_Buck
-hk : HK
-cpu : MSP_430F5438

3.5.2. Operations

Signature: main() : void

Il blocco mostrato in figura 4, costituisce la parte fondamentale del progetto 1B81, le parti che lo compongono saranno ampiamente descritte nei capitoli seguenti, come già ripetuto, questo blocco prende la tensione proveniente dalle celle solari e genera le uscite di tensione che servono al resto del sistema.

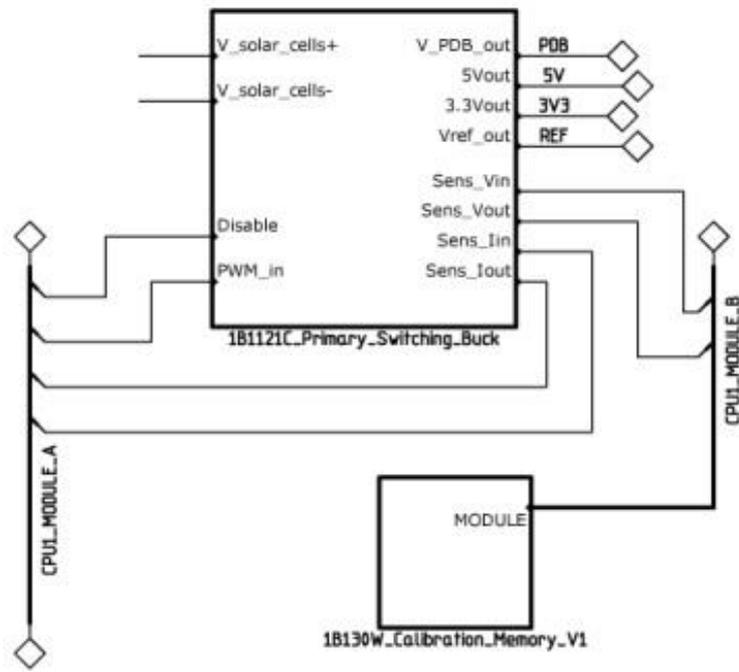


Figura 4: Dettaglio del blocco principale progettato in Mentor Graphics

Siccome presenta al suo interno dei sensori che devono essere controllati dal microcontrollore, porta in uscita, oltre alle tensioni generate, anche le uscite di questi sensori, che sono opportunamente collegate ai BUS della CPU.

La comunicazione tra i blocchi principali, ossia tra il blocco dello switching e il microprocessore avviene mediante bus. Questi bus, come anche il blocco CPU e le interfacce sono standardizzati per il progetto AraMiS, e quindi non verranno descritti in modo completo, in quanto realizzati da altri studenti.

In particolare come mostrato nella figura successiva, ogni CPU ha 8 bus, ognuno dei quali è composto a sua volta da 16 linee, delle quali quattro sono le tensioni 3V3, Ref, 5V e PDB generate dal blocco 1B1121C_Primary_Switching_Buck e le

restanti sono linee analogiche/digitali e sono collegate ai pin del microprocessore.

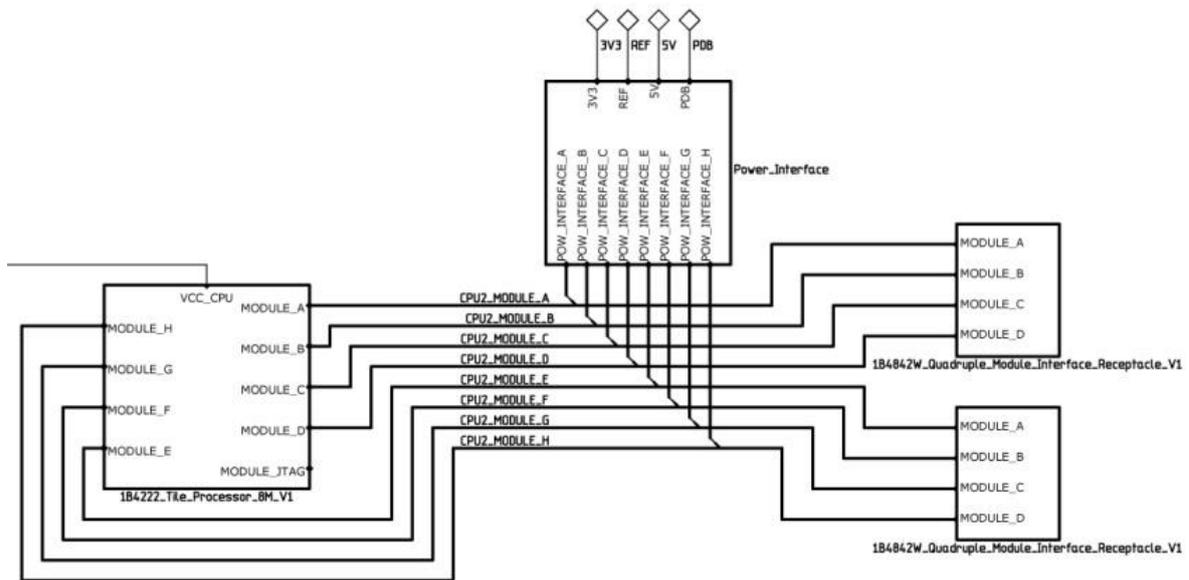


Figura 5: interconnessioni tra CPU e connettori per moduli esterni.

Nel progetto sono presenti 12 connettori esterni, per poter ampliare le funzionalità del sistema con moduli esterni plug&play; lo standard di questi moduli plug&play e del modo con i quali essi debbano comunicare, ed anche le loro dimensioni fisiche sono stati standardizzati, ed in questo caso utilizzati come tali, in altri progetti [14].

Come si può notare in figura, i connettori sono raggruppati in blocchi da quattro, che sono caratterizzati da un particolare layout fisico. E' importante notare il fatto che nel progetto sono presenti due CPU, quindi in teoria si avrebbero a disposizione 16 bus e quindi logicamente sarebbe possibile utilizzare 16 connettori extra. Questo non è possibile in quanto i sensori inseriti all'interno dell'MPPT, utili a regolare il funzionamento di questo, ed i sensori di temperatura, devono anch'essi essere collegati al processore, in particolare agli ingressi analogici della CPU, quindi è stato necessario sacrificare quattro connettori per collegare i suddetti sensori.

Dopo aver realizzato lo schematico, è stato creato il PCB, che è mostrato in piccolo nella figura seguente.

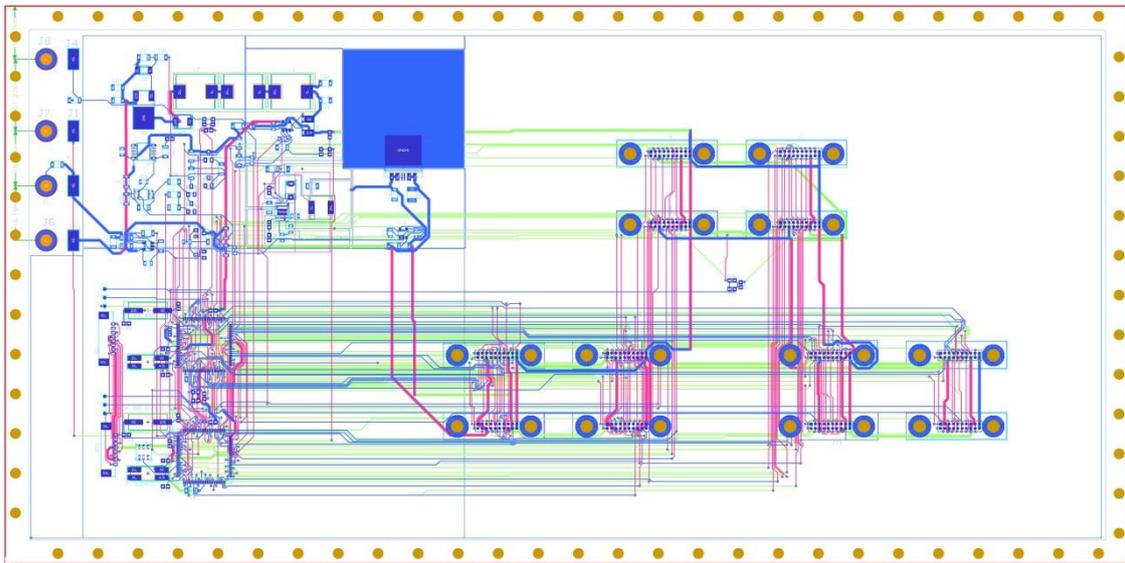


Figura 6: Layout del pcb interno senza plane shape

Le varie linee colorate rappresentano appunto i vari collegamenti, in particolare ogni colore corrisponde ad un layer. Questo PCB è stato realizzato appunto con 4 layer, dove quello più basso è un plane, quindi non è percorso da segnali, ma costituisce il GND.

Nel progetto sono presenti anche dei plane shape, in questo caso sono stati nascosti per migliorare la resa visiva. In particolare tutte le aree che non presentano dispositivi o interconnessioni sono ricoperte da un layer di GND ed i relativi via per interconnetterli con il layer 4.

Si nota facilmente come i connettori siano disposti secondo uno standard, che permette appunto di utilizzare moduli di determinate dimensioni senza precludere inutilmente altri connettori.

La circuiteria in alto a sinistra è quella che realizza l'MPPT, mentre in basso a sinistra sono alloggiate le due CPU con i loro componenti esterni.

Al centro dei blocchi dei connettori è possibile intravedere il sensore di temperatura che misura la temperatura chiamata nel software SAT_TEMP.

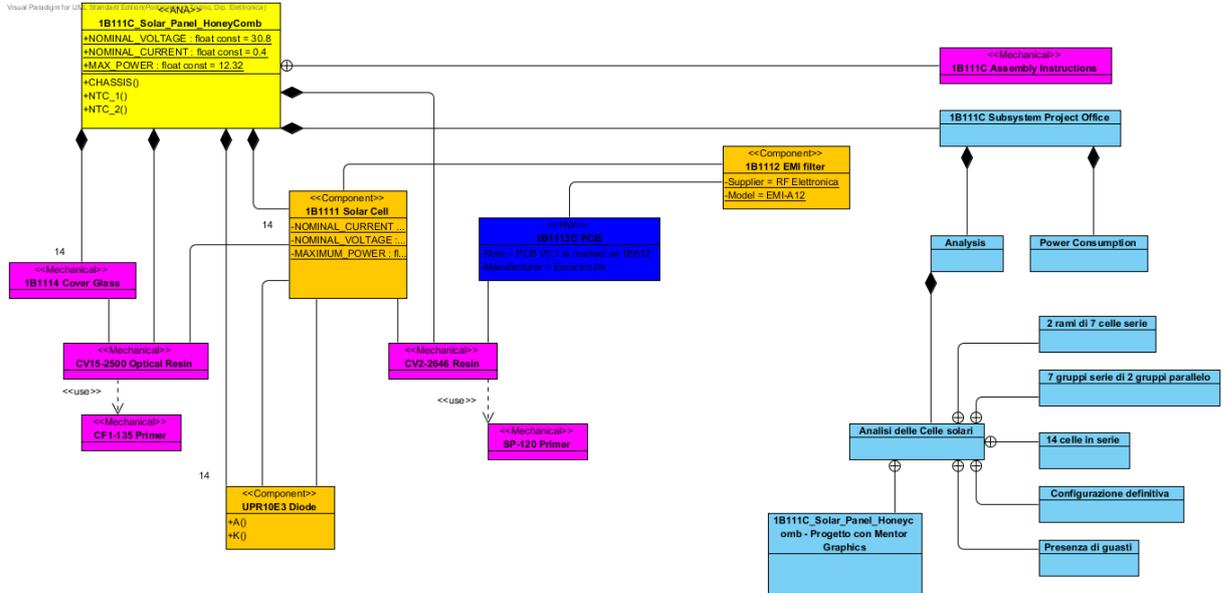
L'evidente piazzola blu è stata realizzata per migliorare la dissipazione del regolatore utilizzato.

Come nel PCB delle celle solari, anche in questo sono presenti i fori passanti che permetteranno di collegarli. E' da notare come sia rispettata la distanza dei fori passanti ed essi siano speculari rispetto all'altro PCB, in quanto appunto i due devono essere incollati insieme con una struttura di honeycomb.

E' inoltre importante notare come le linee di potenza siano più spesse delle restanti, questo appunto poiché devono sopportare stress maggiori e sopportare più corrente.

Per quanto riguarda i dispositivi che gestiscono la potenza, ossia i regolatori, sono stati utilizzati dei layout particolari, descritti nei vari datasheet dai produttori.

4.1B111C Solar Panel – HoneyComb



Questo diagramma delle classi rappresenta il blocco 1B111C_Solar_Panel_HoneyComb che fa parte a sua volta del sistema 1B81 Power management honeycomb.

Il sistema 1B111C, caratterizzato dall'utilizzo di quattordici celle solari, si occupa della trasformazione dell'energia proveniente dal sole in energia elettrica per mezzo di celle solari a tripla giunzione descritte nei paragrafi seguenti

Questo sistema è caratterizzato in particolare da 14 celle solari che verranno posizionate ed incollate al loro posto su un PCB che sarà situato sul lato esterno del satellite e quindi potrà essere illuminato dal sole. La procedura di incollaggio descritta nello schema è stata sviluppata e perfezionata anche grazie all'azienda Neohm che ha permesso appunto di effettuare diversi test a riguardo.

4.1. 1B111C Assembly Instructions

4.1.1. 1B1113C PCB

4.1.1.1. Attributes

Signature	Documentation
-Note = PCB V0.1 is marked as 1B612	
-Manufacturer = Eurocircuits	

4.1.2. 1B1112 EMI filter

4.1.2.1. Attributes

Signature	Documentation
-Supplier = RF Elettronica	
-Model = EMI-A12	

4.1.3. 1B1111 Solar Cell

4.1.3.1. Attributes

Signature	Documentation
- <u>NOMINAL CURRENT</u> : float = <u>0.4</u>	Current at MPPwith 1,350W/m2 at AM0, in A
- <u>NOMINAL VOLTAGE</u> : float = 2.2	Voltage at MPPwith 1,350W/m2 at AM0, in V
- <u>MAXIMUM POWER</u> : float = 0.88	Maximum Power Pointwith 1,300W/m2 at AM0, in W.

4.1.4. CV15-2500 Optical Resin

Electrically and thermally conductive epoxy

4.1.4.1. Attributes

Signature

Documentation

- Manufacturer = NUSIL
 - Shear stress (MPa) = 1.6
 - Shear modulus (MPa) = 2 (above -40oC)
 - Electrical conductivity (Ohm m) = $1e13$
 - Thermal conductivity (W/mK)
 - Application = Silicone
-

4.1.5. CF1-135 Primer

4.1.5.1. Attributes

Signature

Documentation

- Manufacturer = NUSIL
 - Application = Primer for CV15-2500
-

4.1.6. SP-120 Primer

Non-conductive epoxy

4.1.6.1. Attributes

Signature

Documentation

- Manufacturer = NUSIL
 - Application = Primer for CV2-2646
-

4.1.7. CV2-2646 Resin

4.1.7.1. Attributes

Signature	Documentation
-Manufacturer = NUSIL	
-Shear stress (MPa) = 1.2	
-Shear modulus (MPa)	
-Electrical conductivity (Ohm m) = $2e-5$	
-Thermal conductivity (W/mK) = 1.5	
-Application = Thermally & electrically conducting	

4.1.8. UPR10E3 Diode

4.1.8.1. Operations

Signature: A()

Signature: K()

4.2. Analisi delle Celle solari

Una cella solare è un dispositivo in grado di convertire l'energia solare in energia elettrica, sfruttando l'effetto fotovoltaico, in generale le celle solari vengono usate come generatore di corrente. In particolare le celle solari sono dei diodi con superficie estesa, in grado di convertire l'energia solare incidente sul semiconduttore in energia elettrica. Questa conversione avviene in quanto i fotoni emessi dal sole che hanno un'energia superiore all'energy gap del semiconduttore utilizzato, vengono da esso intrappolati e passano dalla banda di valenza, a quella di conduzione, generando così una corrente.

Le celle solari che utilizzeremo sono celle in Arseniuro di gallio (GaAs) che assicurano rendimenti elevati dovuti al fatto che questo semiconduttore è caratterizzato da un gap diretto, l'unico problema rilevante di questo materiale è l'elevatissimo costo che è legato appunto all'ottima qualità.



Figura 7: Layout delle celle solari utilizzate

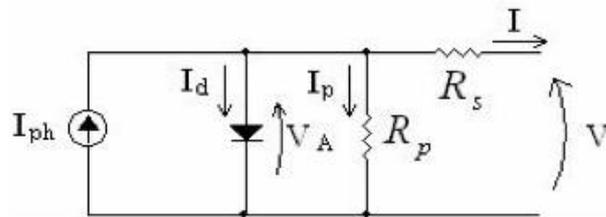
Le celle solari che abbiamo a disposizione hanno una forma a parallelepipedo (come mostrato in figura 7) di dimensioni 39,55mm per 68,96 mm, con efficienze che vanno dal 23% al 24% a seconda della cella.

4.2.1. Modello matematico delle celle solari.

Analizziamo ora il modello matematico che costituisce una cella solare [5]: la corrente elettrica generata in una cella ideale, ossia senza cadute di tensione e correnti parassite, può essere rappresentata dalla seguente equazione:

$$I = I_{ph} - I_d = I_{ph} - I_S \left(e^{\frac{V_A}{V_T}} - 1 \right)$$

In particolare I_S è la corrente di saturazione inversa del diodo, V_A è la tensione diretta ed I_{ph} è la corrente generata dalla cella.

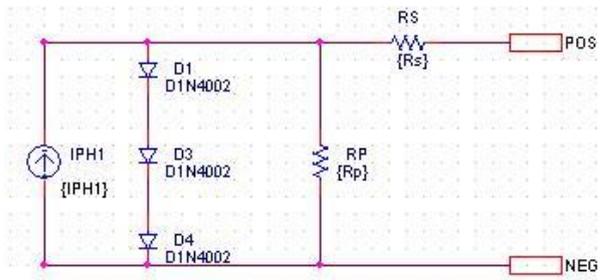


Nello schema è raffigurato il modello di cella solare utilizzato. In questo modello, oltre al diodo ed al generatore di corrente, sono presenti due resistenze: R_p rappresenta la resistenza di shunt, dovuta ai difetti della giunzione, eventuali impurità e piccoli corto circuiti; R_s invece rappresenta la resistenza elettrica presente intrinsecamente all'interno del semiconduttore.

Questo modello è sufficientemente accurato per studiare e simulare il comportamento delle celle solari.

4.2.2. Analisi e modello Spice

Per analizzare il comportamento delle celle solari, e per poter quindi scegliere la configurazione più adatta a questo progetto, è stato realizzato un modello circuitale mediante il simulatore Pspice [6].



Come si può vedere in figura è stato utilizzato un generatore di corrente IPH1 di valore parametrizzato, tre diodi in serie e due resistenze.

Sono stati utilizzati tre diodi in serie, anziché uno, per ottenere la tensione di soglia necessaria per la corretta modellizzazione della cella solare, in quanto non esistono in commercio diodi con tensioni di soglia superiori a 0.6V.

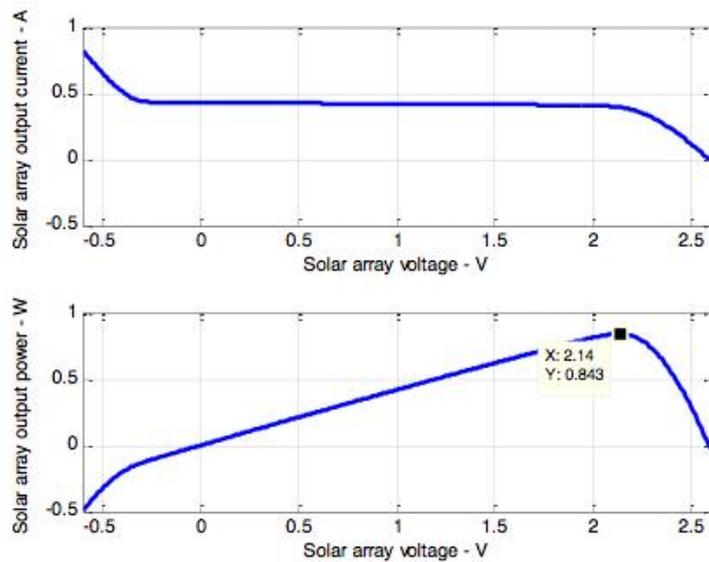
Sulla base dei datasheet delle celle che verranno utilizzate, sono stati fissati i valori dei vari componenti:

$$I_{ph} = 420mA \pm 2\%$$

$$R_p = 1k\Omega$$

$$R_s = 0,1\Omega$$

La simulazione del comportamento della cella modellizzata in Spice mostra i seguenti risultati:

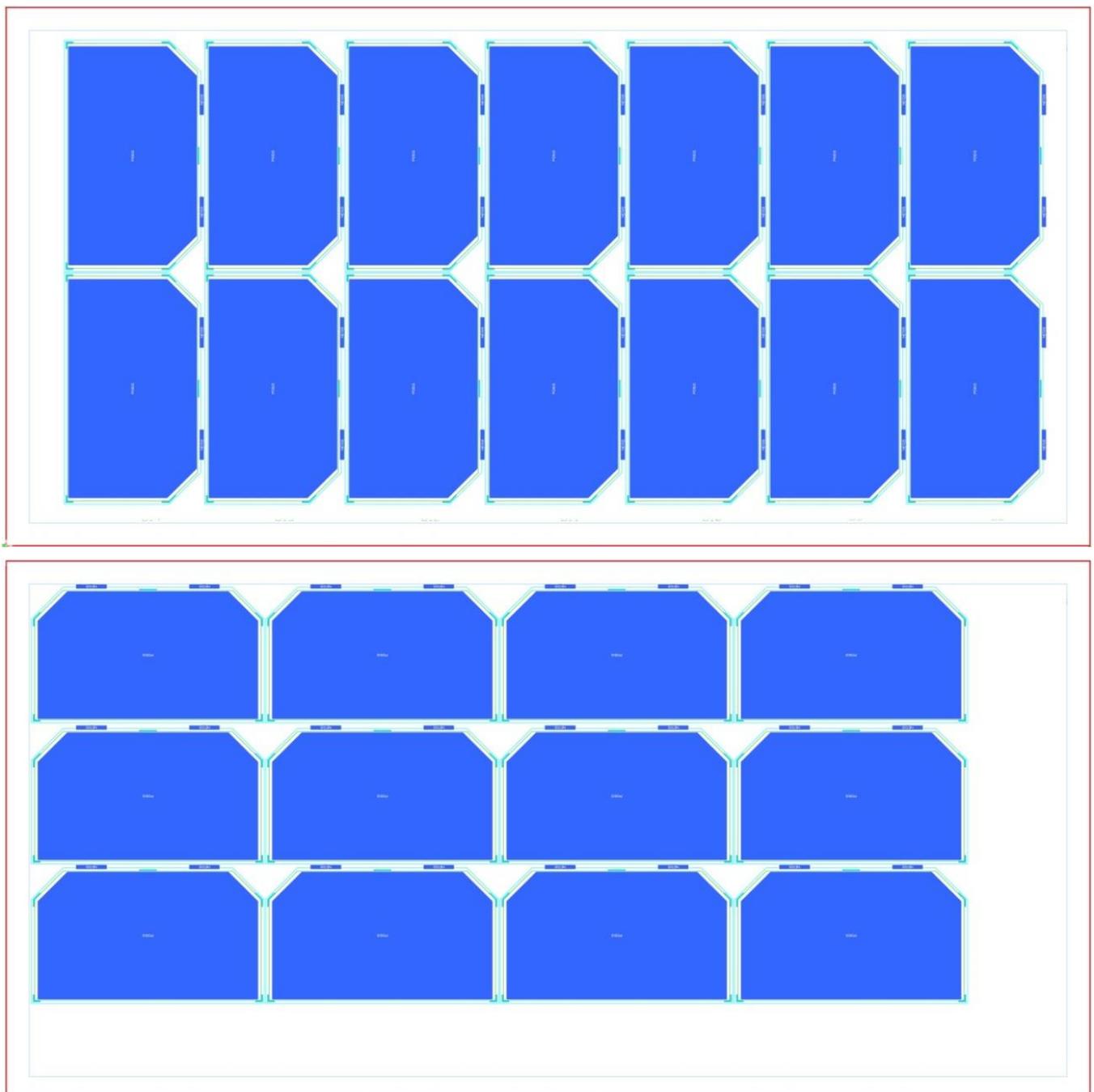


Confrontando l'andamento dei dati ottenuti con quelli forniti nei datasheet delle celle, è possibile concludere che il modello che si vuole utilizzare è coerente e corretto.[7] Il grafico più interessante da analizzare è quello della potenza. Infatti il sistema di gestione dell'energia che sarà a valle delle celle solari cercherà di far funzionare sempre il sistema nel punto a massima potenza. In questo caso, con una singola cella otteniamo una potenza di 0,84W ad una tensione di 2,14V.

4.2.3. Scelta del numero di celle solari

Nel progetto del sistema 1B111C_Solar_Panel_Honeycomb è necessario utilizzare il massimo numero di celle fisicamente alloggiabili sul PCB di dimensioni 165mm per 330mm, in modo da ottenere più energia per il sistema.

E' stato analizzato il modo più efficiente per utilizzare il massimo numero di celle, in particolare le celle sono state affiancate in modi diversi, ma come mostrato nelle figure, si riesce ad utilizzare il numero massimo di celle, ossia quattordici, con la configurazione verticale.



E' da notare come questa sia semplicemente una prova generale, in quanto nella realizzazione definitiva le celle non potranno essere posizionate così vicine le une alle altre, in quanto sarà necessario effettuare i collegamenti ed inserire ulteriori componenti necessari al funzionamento del sistema.

Avendo scelto il numero di celle da utilizzare, è ora necessario scegliere la configurazione elettrica più opportuna per collegare queste. In particolare esistono diverse configurazioni possibili:

- 14 celle in serie
- 2 rami di 7 celle serie
- 7 gruppi in serie di 2 celle parallelo
- 14 celle in parallelo

A seconda della configurazione scelta si otterranno valori di tensione, corrente e potenza diverse [8]. Per scegliere la configurazione ottimale è necessario tener conto di questi valori, ma considerarne anche altri relativi alle possibilità che avvenga un guasto sulla scheda. Possiamo ad esempio pensare che nel caso di 14 celle in serie, se una di questa subisse dei danni e smettesse di funzionare causando un circuito aperto, il risultato sarebbe di compromettere l'intero pannello.

La configurazione quattordici celle in parallelo è stata scartata fin dall'inizio, in quanto, in questo caso si ottiene una tensione troppo bassa, ed è un controsenso utilizzare un numero così alto di celle ricavando tensioni basse.

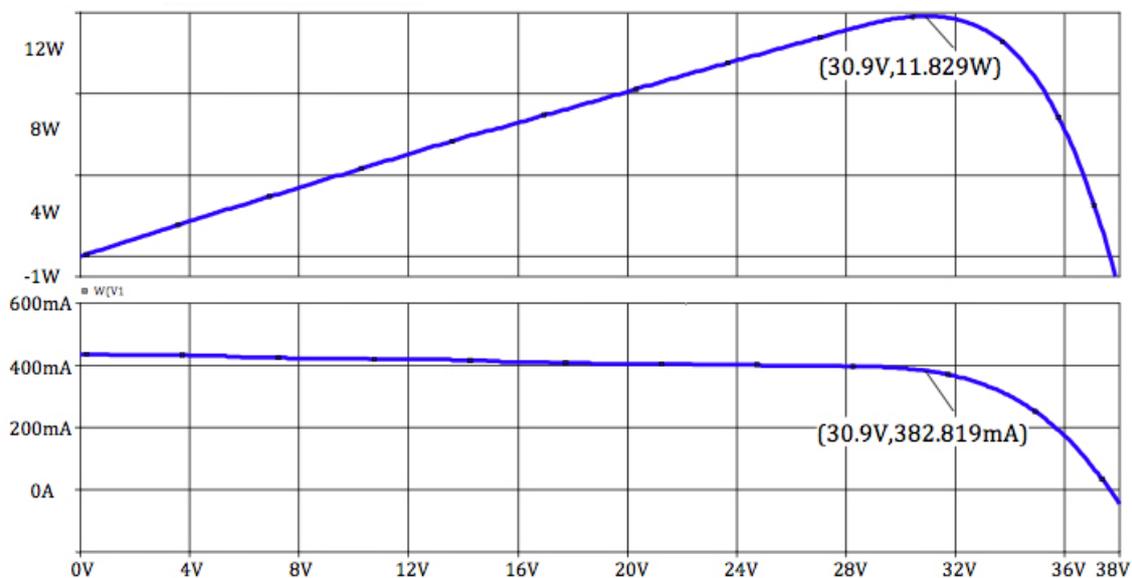
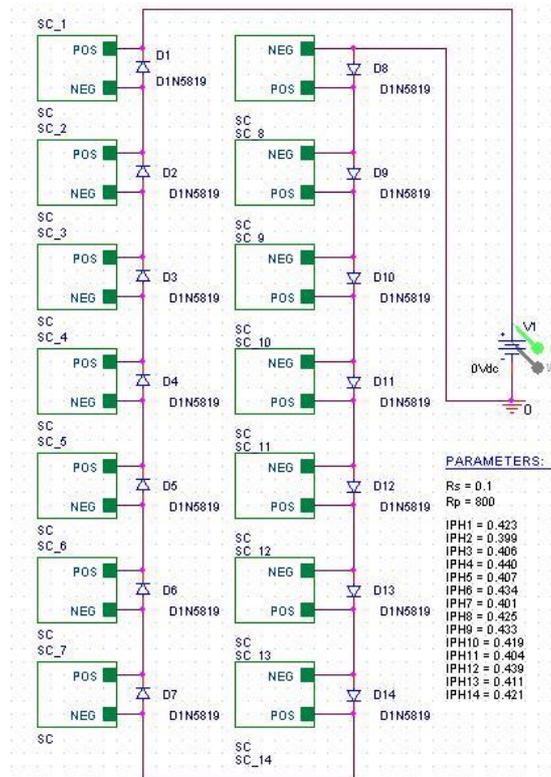
Per risolvere questi problemi, come verrà meglio spiegato successivamente, per ogni cella verranno inseriti uno o più diodi di protezione: per poter scegliere correttamente quale configurazione utilizzare è utile analizzare cosa accade quando c'è un guasto su una cella, ma anche quando c'è un guasto sul diodo in parallelo alla cella.

Nelle sezioni seguenti verranno analizzate tutte le configurazioni ed i relativi guasti.

4.2.4. 14 celle in serie

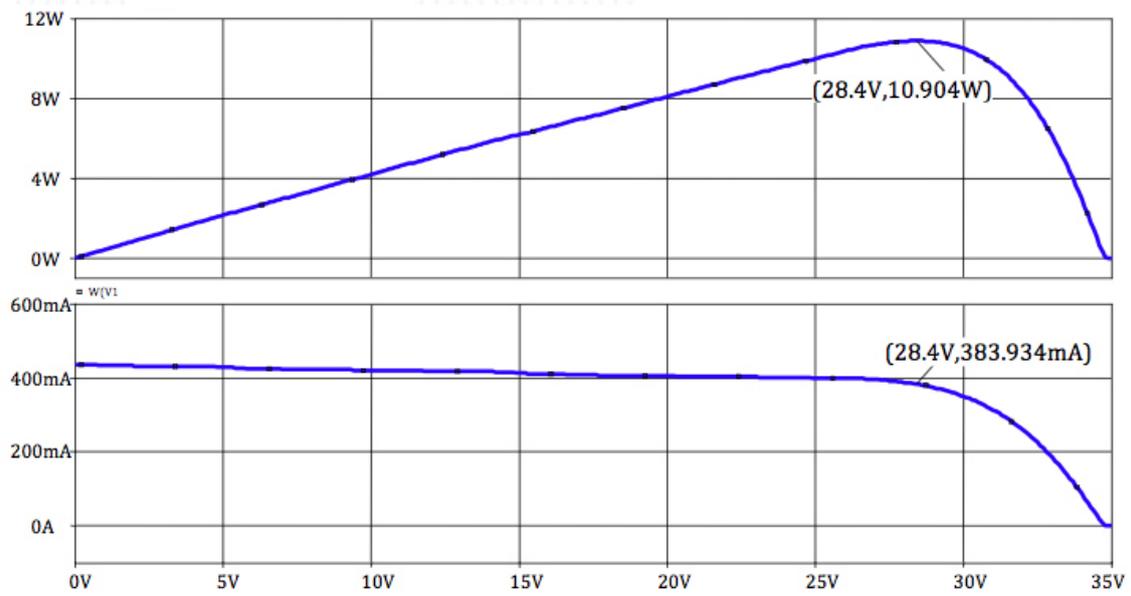
La configurazione che prevede quattordici celle in serie è stata analizzata non solo nel caso di funzionamento normale, ma anche nel caso in cui si verificano dei differenti casi di guasto, come quelli che possono presentarsi ad esempio sul diodo.

- Configurazione di 14 celle in serie, comportamento normale.

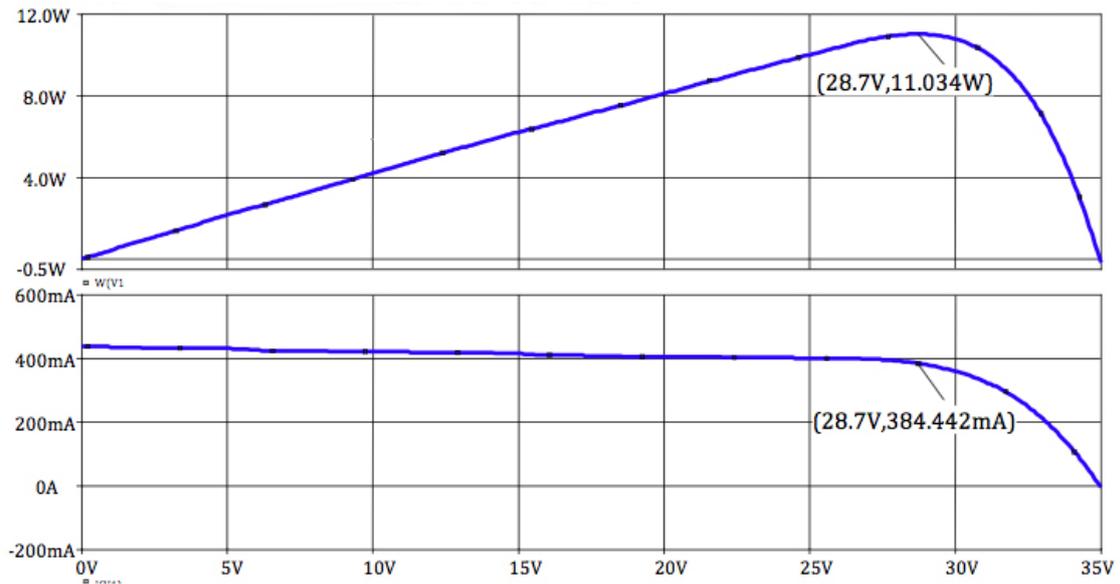


con una configurazione di questo tipo si ottiene una potenza massima di 11,8W ad una tensione di quasi 31V.

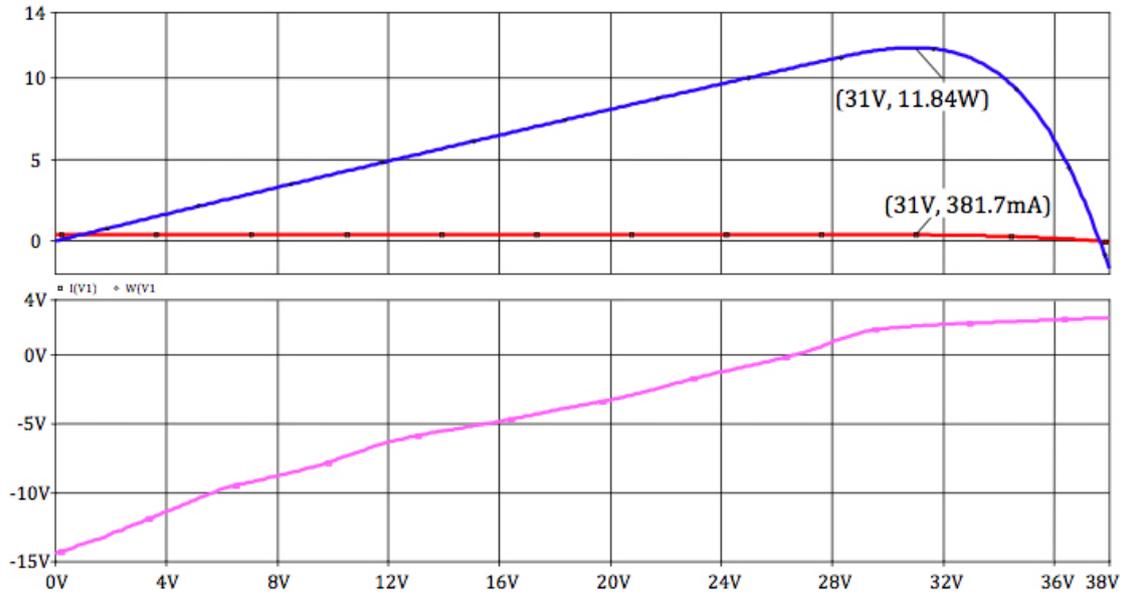
- Guasto circuito aperto sulla cella



- Guasto corto circuito sulla cella



- Guasto diodo in circuito aperto



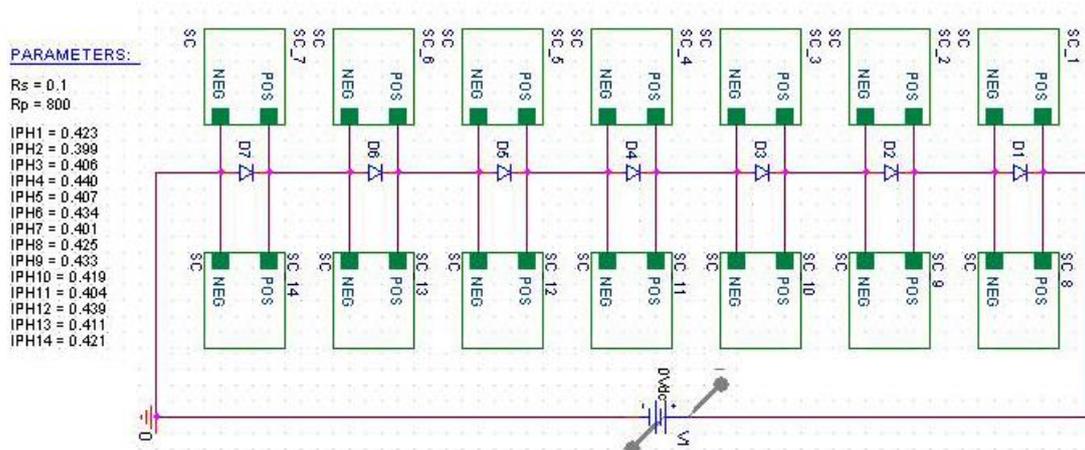
Riassumendo i valori ottenuti nelle varie simulazioni, possiamo notare come in caso di guasto alla cella si ha una perdita di potenza del 7% circa, rispetto a quella massima.

Guasto	Potenza massima	Corrente @P _{MAX}	Tensione @P _{MAX}	Potenza persa (%)
-	11.8W	768mA	30.9V	-
guasto cella CA	10,9W	383mA	28.4V	7.63%
guasto cella CC	11W	384mA	28.7V	6.78%
guasto diodo CA	11.8W	382mA	31V	-

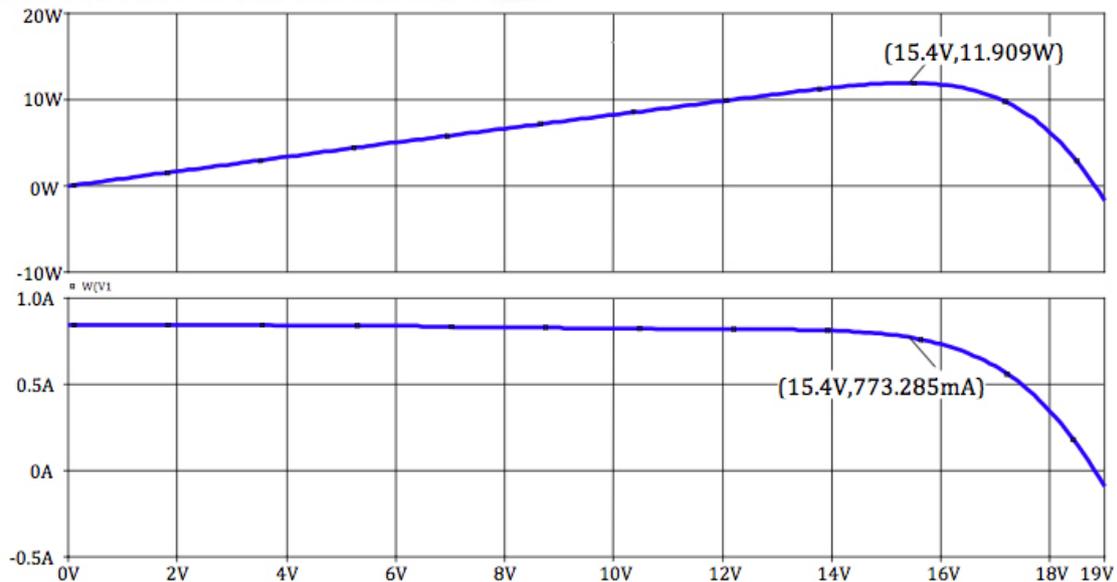
4.2.5. 7 gruppi in serie di 2 celle in parallelo

La configurazione che prevede sette gruppi in serie di due celle parallelo è stata analizzata non solo nel caso di funzionamento normale, ma anche nel caso in cui si verificano dei differenti casi di guasto, come quelli che possono presentarsi ad esempio sul diodo.

- Configurazione 7 gruppi serie di 2 celle parallelo, comportamento normale.

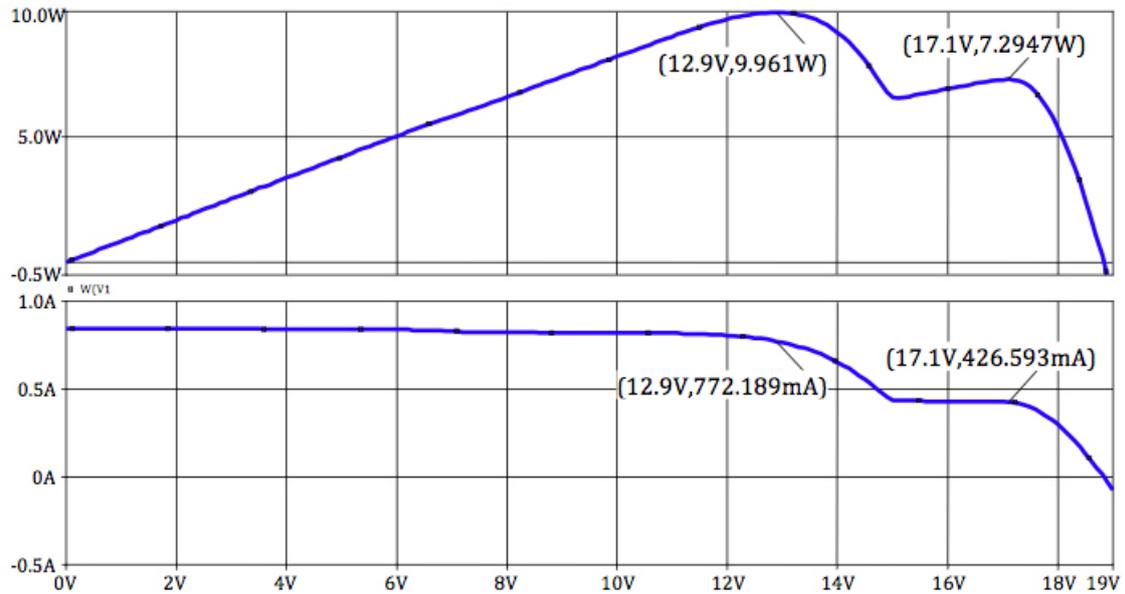


L'andamento grafico del comportamento, simulato in Spice, in questo caso è mostrato nella seguente figura:



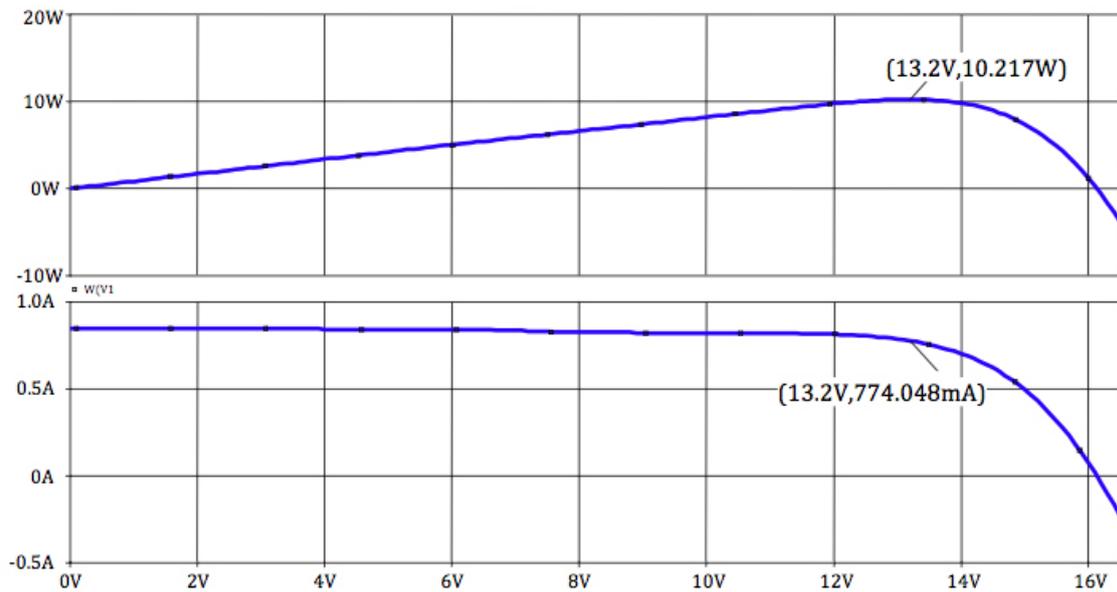
Con questa configurazione si ha una potenza massima di quasi 12W con una tensione di 15.4V.

- Guasto circuito aperto sulla cella

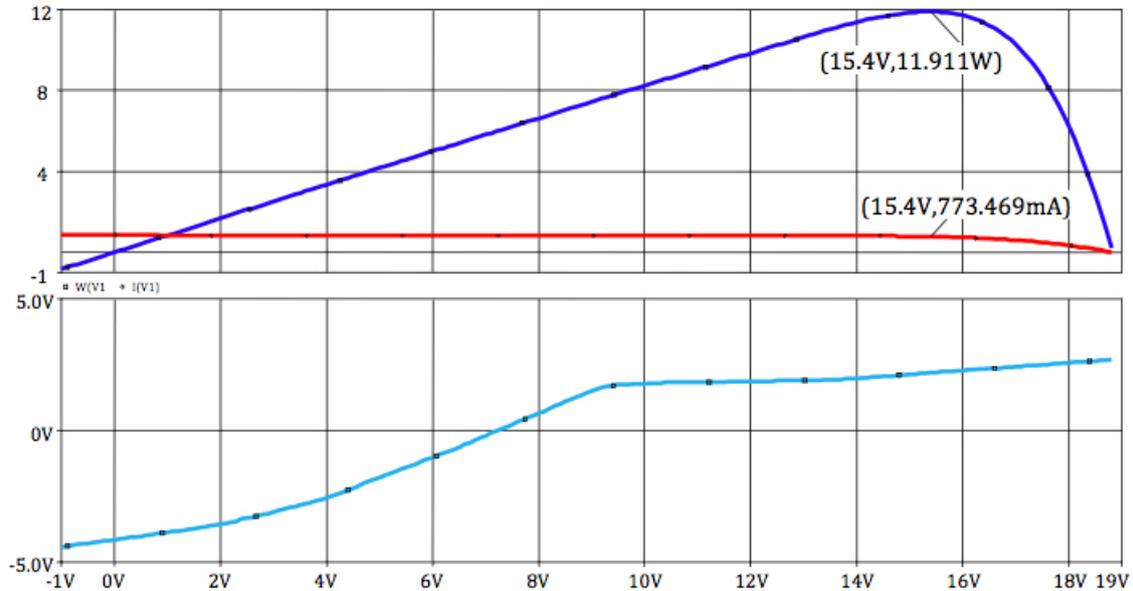


E' importante notare come in questo caso venga a crearsi un secondo massimo relativo nella caratteristica.

- Guasto corto circuito sulla cella



- Guasto diodo in circuito aperto



In questa figura è possibile vedere l'andamento della potenza (in alto) e della corrente (in basso) in funzione della tensione, nel caso in cui si presenti un guasto sul diodo di protezione.

Riassumendo i valori ottenuti nelle varie simulazioni, otteniamo la seguente tabella:

Guasto	Potenza massima	Corrente @P _{MAX}	Tensione @P _{MAX}	Potenza persa (%)	Secondo massimo
-	11.9W	773mA	15.4V	-	-
guasto cella CA	9.96W	773mA	12.9V	16.3%	7.29W
guasto cella CC	10.2W	773mA	13.2V	14.3%	-
guasto diodo CA	11.9W	773mA	15.4V	-	-

In questo caso si può notare che nel caso in cui si presenti un guasto singolo ad una cella si ha una perdita di potenza pari a circa il 16% della potenza totale.

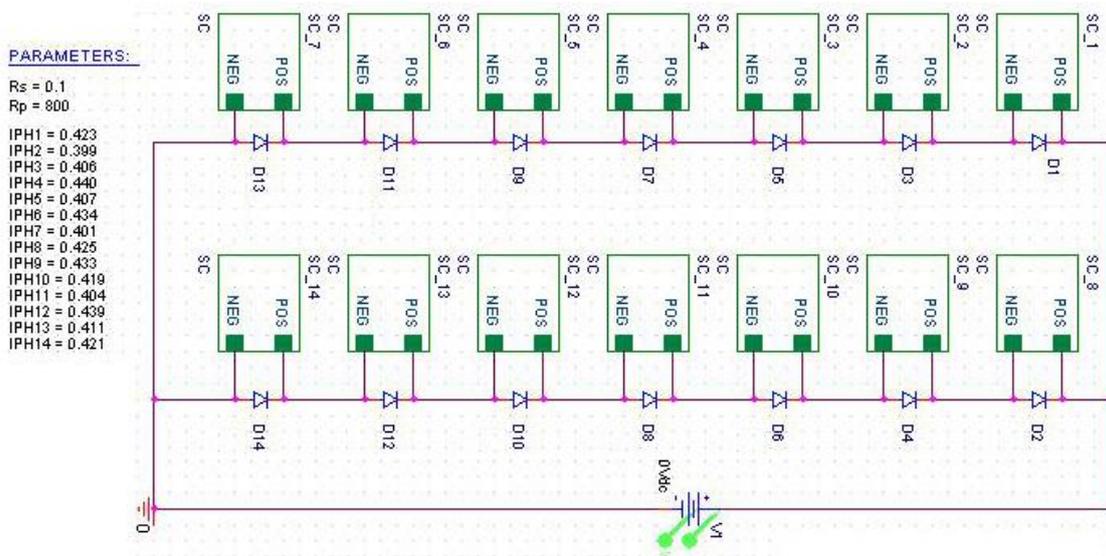
Inoltre nel caso in cui si presenti un guasto su una cella di tipo circuito aperto sussiste un nuovo problema: come si nota dal grafico simulato, viene a crearsi un secondo massimo ad una potenza inferiore a quella ottimale. Questo può causare gravi perdite di rendimento del sistema, infatti si vuole progettare un convertitore

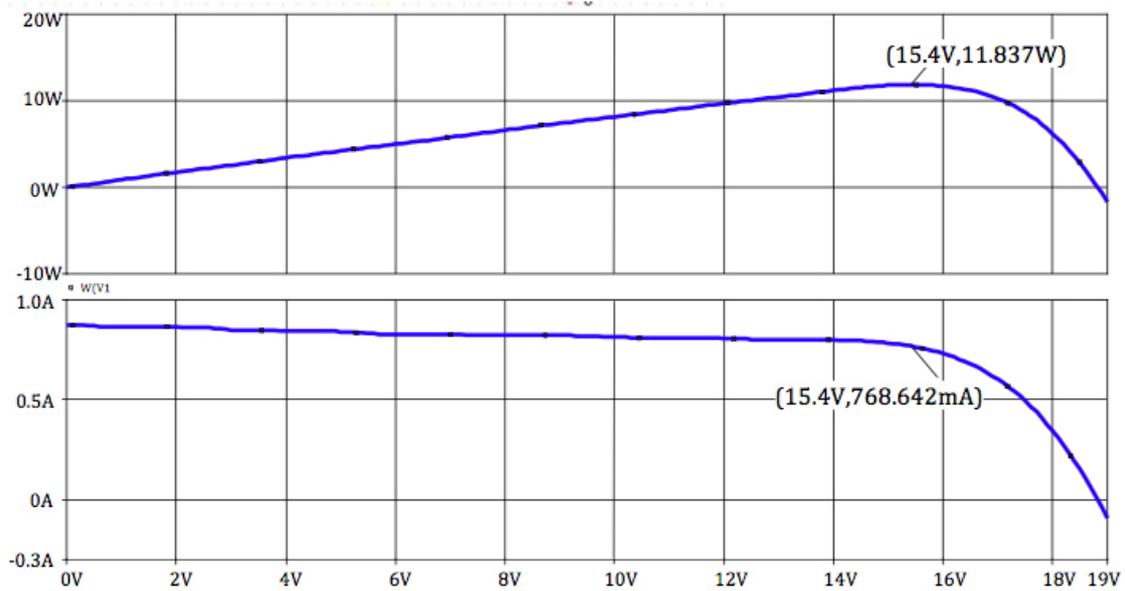
MPPT che lavori sul punto a massima potenza, ossia un sistema che sia in grado di muoversi lungo la caratteristica fino ad impostare come punto di lavoro il punto a potenza massima, ma la presenza di un secondo massimo relativo, potrebbe generare un falso positivo, bloccando il sistema in un punto che non è il massimo assoluto.

4.2.6. 2 rami di 7 celle in serie

La configurazione che prevede due rami paralleli di sette celle in serie è stata analizzata non solo nel caso di funzionamento normale, ma anche nel caso in cui si verificano dei differenti casi di guasto, come quelli che possono presentarsi ad esempio sul diodo.

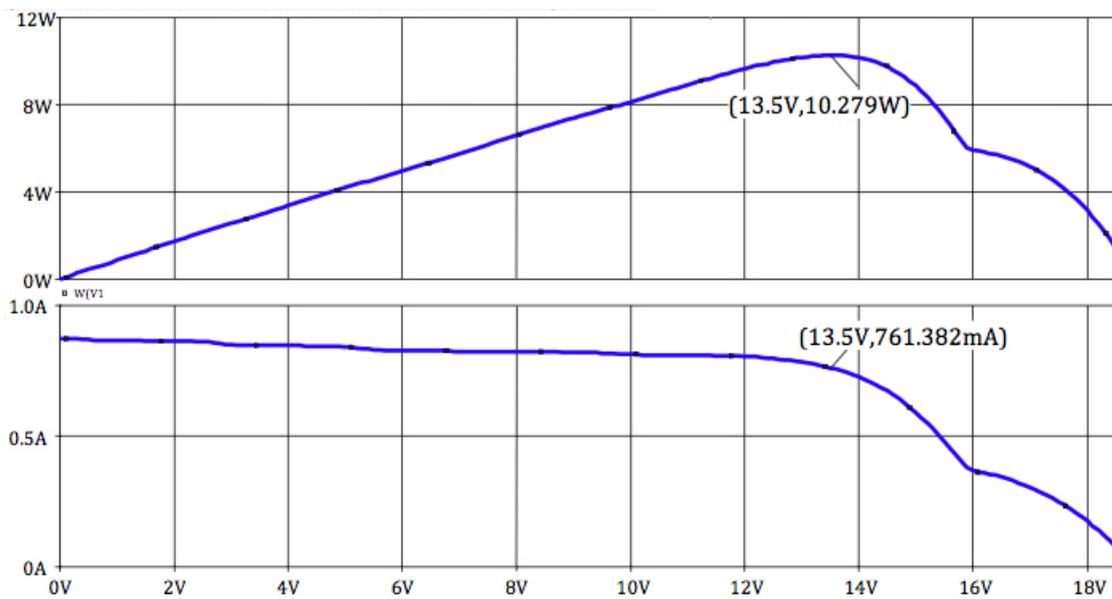
- Configurazione due rami parallelo di 7 celle serie, caso nominale.



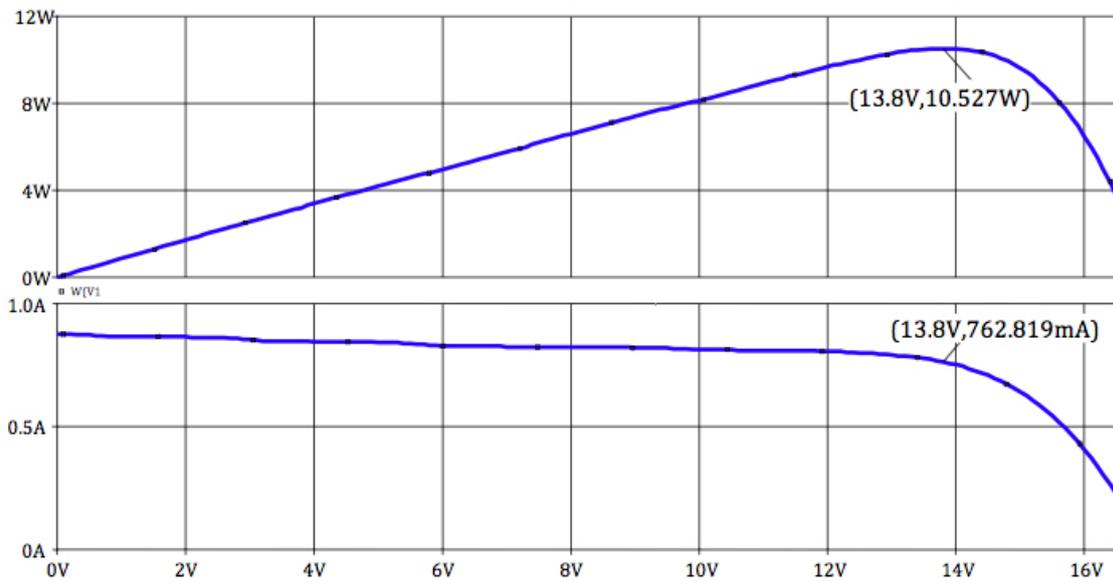


Con questa configurazione si ottiene una potenza massima di 11.9W ad una tensione di 15.4V.

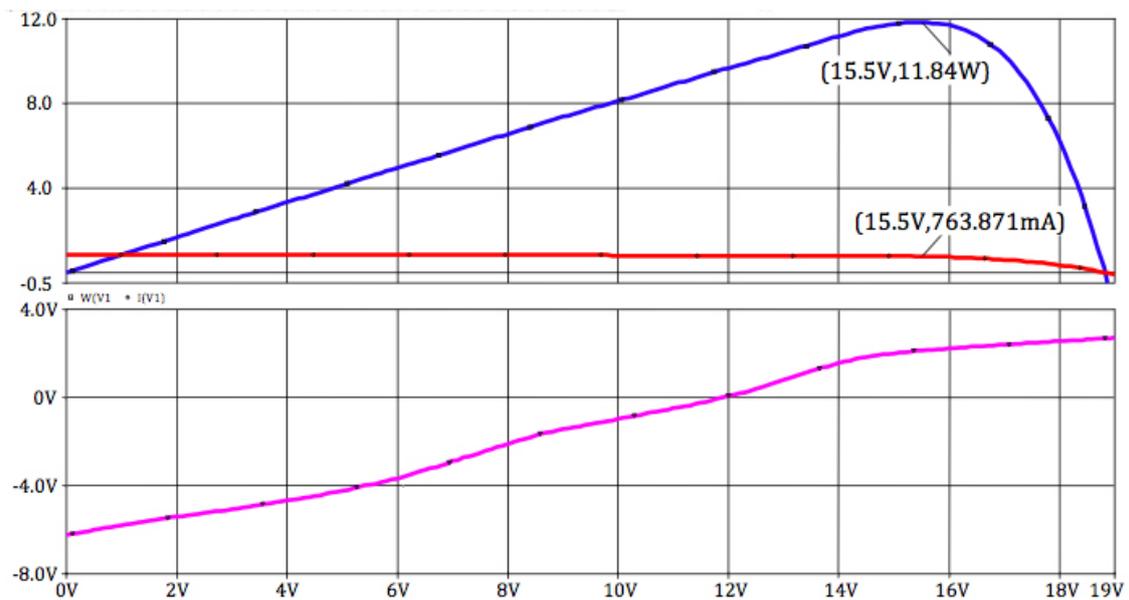
- Guasto circuito aperto sulla cella



- Guasto corto circuito sulla cella



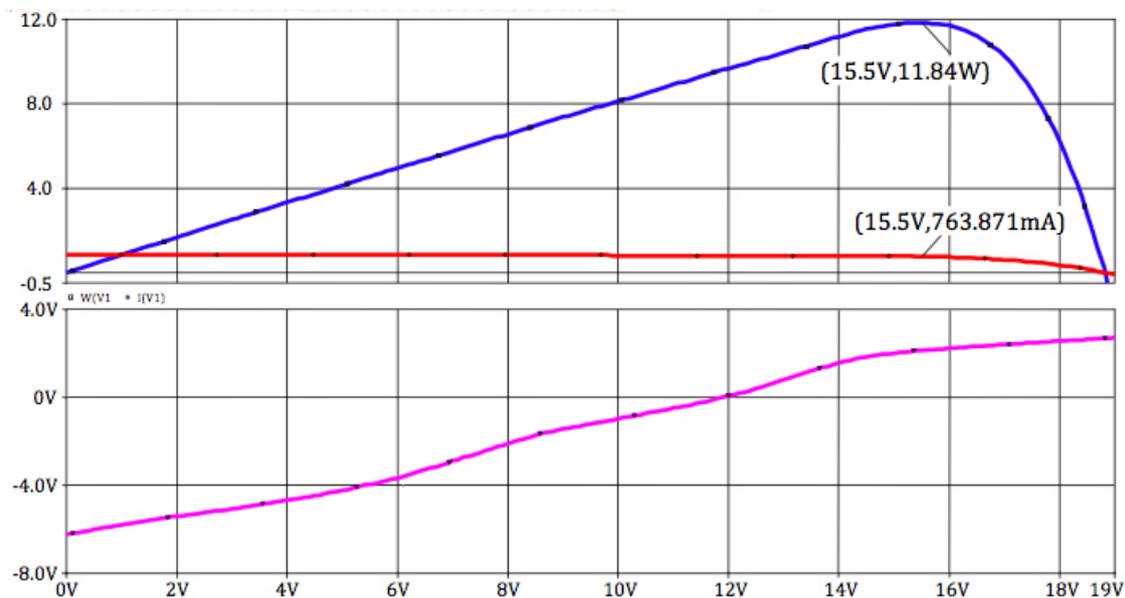
- Guasto diodo in circuito aperto



Riassumendo i valori ottenuti nelle varie simulazioni, possiamo notare come in questo caso, nel caso si presenti un guasto singolo ad una cella, il risultato è una perdita di potenza pari a circa il 13% della potenza totale generata.

Nella seguente tabella sono riassunti i punti più importanti delle caratteristiche simulate.

Guasto	Potenza massima	Corrente @PMAX	Tensione @PMAX	Potenza persa (%)
-	11.8W	768mA	15.4V	-
guasto cella CA	10.5W	762mA	13.8V	11%
guasto cella CC	12.2W	761mA	13.5V	13.56%
guasto diodo CA	11.8W	763mA	15.5V	-



Riassumendo i valori ottenuti nelle varie simulazioni, possiamo notare come in questo caso, nel caso si presenti un guasto singolo ad una cella, il risultato è una perdita di potenza pari a circa il 13% della potenza totale generata.

Nella seguente tabella sono riassunti i punti più importanti delle caratteristiche simulate.

Guasto	Potenza massima	Corrente @PMAX	Tensione @PMAX	Potenza persa (%)
-	11.8W	768mA	15.4V	-
guasto cella CA	10.5W	762mA	13.8V	11%
guasto cella CC	12.2W	761mA	13.5V	13.56%
guasto diodo CA	11.8W	763mA	15.5V	-

4.3. Configurazione definitiva

Considerando tutte le varie configurazioni riassunte in tabella, abbiamo scelto di utilizzare 14 celle in serie con un convertitore di tipo Buck, ossia la configurazione che presenta una perdita in termini di potenza inferiore.

Configurazione celle	Guasto	Power (MAX)	Current @Pmax	Voltage @Pmax	%Power lost	Secondo massimo
Cella singola		0.84 W	392 mA	2.14 V	-	-
14 celle in serie	-	11.8W	382mA	30.9V	-	-
14 celle in serie	CC	11W	384mA	28.7V	6.78%	-
14 celle in serie	CA	10.9W	383mA	28.4V	7.63%	-
14 celle in serie	diodo	11.8W	382mA	31V	-	-
2 rami parallelo di 7 celle serie	-	11.8W	768mA	15.4V	-	-
2 rami parallelo di 7 celle serie	CC	10.2W	761mA	13.5V	13.56%	-
2 rami parallelo di 7 celle serie	CA	10.5W	762mA	13.8V	11%	-
2 rami parallelo di 7 celle serie	diodo	11.8W	763mA	15.5V	-	-
7 gruppi di 2 celle in parallelo	-	11.9W	773mA	15.4V	-	-
7 gruppi di 2 celle in parallelo	CA	9.96W	772mA	12.9V	16.3%	SI
7 gruppi di 2 celle in parallelo	CC	10.2W	774mA	13.2V	14.3%	-
7 gruppi di 2 celle in parallelo	diodo	11.9W	773mA	15.4V	-	-

Questa scelta è stata condizionata anche da altri fattori. Infatti nelle restanti configurazioni, oltre a perdite maggiori, si ha la presenza di più massimi locali, che potrebbero compromettere il funzionamento dell'MPPT. Inoltre è utile considerare anche il fatto che, volendo ottenere una tensione in uscita dopo il convertitore di

circa 14V, nei casi di 7 gruppi da 2 celle serie o di 2 paralleli da 7 celle, otterremmo delle tensioni che sono sia superiori che inferiori al valore desiderato: questo implicherebbe l'utilizzo di un convertitore Buck-Boost che presenta molti più problemi di progettazione ed in generale risulta più problematico da utilizzare.

4.4. Presenza di guasti

Data l'importanza del modulo che stiamo progettando, è stato necessario analizzare in modo accurato tutti i possibili guasti. Come è stato dimostrato con i precedenti calcoli, un malfunzionamento di una cella o di un componente può causare elevate perdite di potenza, è quindi stato necessario ridurre al minimo la possibilità che un guasto influenzi il rendimento dell'intero sistema. Nonostante non sia possibile eliminare in alcun modo la possibilità che una cella solare vada in corto circuito o circuito aperto, è invece possibile in modo semplice ovviare ai problemi di guasto sul diodo, utilizzandone due in parallelo. Nel progetto finale infatti sono stati utilizzati due diodi di protezione in parallelo per ogni cella, in modo da evitare che il danneggiamento di uno di questi causi malfunzionamenti. Inoltre, come verrà spiegato successivamente, le piste che collegano i vari componenti sono state realizzate con dimensioni molto superiori al necessario, sempre per evitare che un danno fisico di queste possa portare ad un circuito aperto, compromettendo uno o più elementi del sistema.

4.5. 1B111C_Solar_Panel_Honeycomb - Progetto con Mentor Graphics

In seguito all'analisi matematica ed allo studio relativo al numero di celle solari da utilizzare, per mezzo del tool Mentor Graphics [9][10][11], sono stati realizzati gli schematici ed il PCB caratterizzati appunto dalle 14 celle solari.

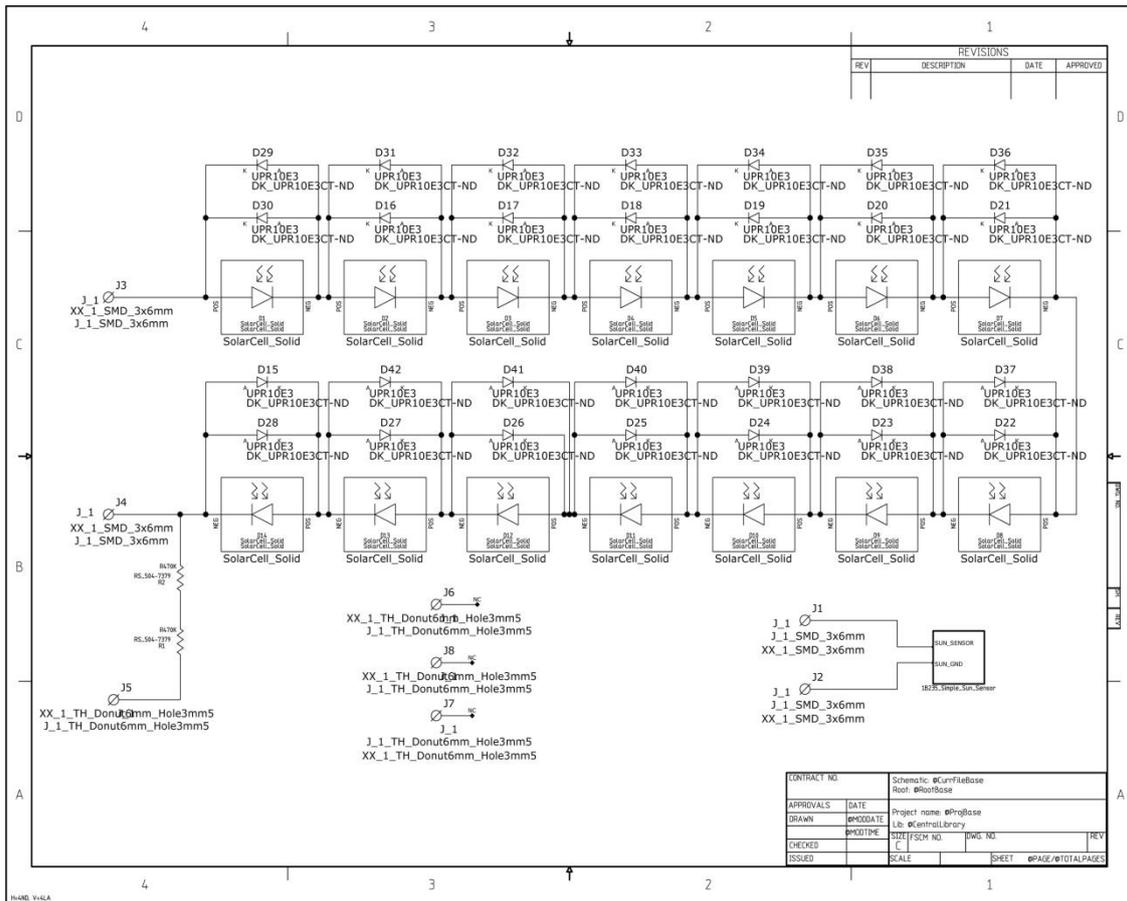


Figura 8: schematico 1B111C Solar Panel Honeycomb

In particolare è da notare come nello schema delle celle solari sia presente anche un sensore di sole, che non è stato realizzato in questa tesi, ma preso dal materiale AraMiS pre-esistente. Come per il sensore di sole, i connettori passanti utilizzati in questo progetto sono stati precedentemente definiti in altri progetti, perciò vengono considerati come componenti pre-esistenti e non sono direttamente analizzati in questa tesi.

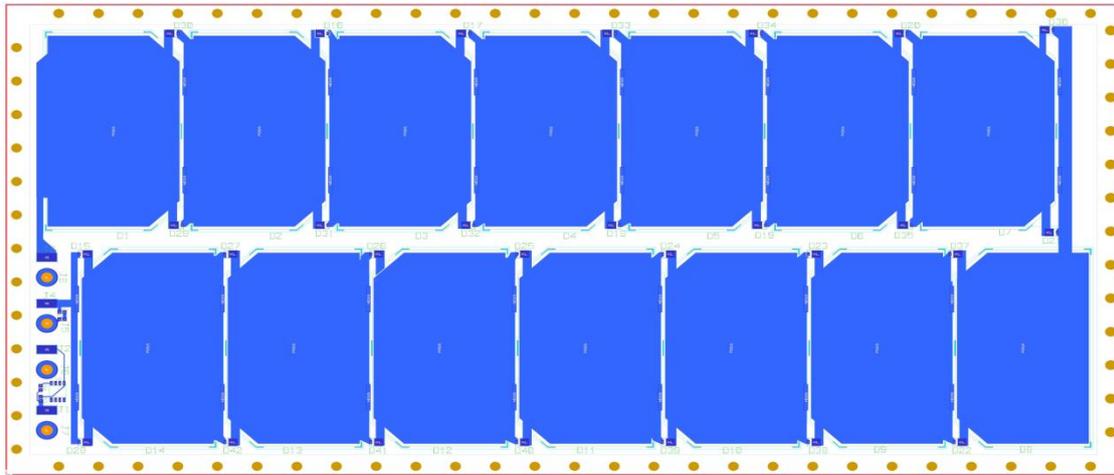


Figura 9: layout 1B111C Solar Panel Honeycomb

In generale come si può notare dal PCB realizzato, sono utilizzati quattro connettori passanti: due per il sensore di sole, ed altri due per trasferire l'energia delle celle solari al convertitore MPPT realizzato su un altro PCB; infatti vengono portati al convertitore sia il morsetto negativo che quello positivo, provenienti dal pannello solare.

Sull'esterno del PCB sono presenti dei fori (rappresentati con pallini arancioni) che servono per semplificare il fissaggio di questo pannello alla struttura del satellite per mezzo di supporti meccanici ancora in fase di sviluppo.

5.1B1121C_Primary_Switching_Buck

This class represents the module of power conversion that generates the various supply for the entire satellite.

The parts involved in converting the power coming from solar cells are parts of the so-called 1B1121C_MPPT.

In particular will generated:

- 5V (1A) from 1B1121C_5VPower block.
- 3,3V (1,6A) from 1B1121C_3V3Power block.
- 3V reference, named REF, from 1B1121C_3VRef block.
- 14V named PDB from 1B1121C_MPPT block.

Other important devices that compose this class are the sensors, in particular in particular there are four different sensors::

- Block Bk1B131C_Voltage_Sensor measures a voltage with range from 0 to 20V.
- Block Bk1B131D_Voltage_Sensor measures a voltage with range from 0 to 40V.
- Block Bk1B132C_Current_Sensor measures a current that can be 2.5A maximum.
- Block Bk1B132E_LowSideCurrentSensor measures a "low side" current, that means current between a point and GND, using an operational amplifier.

The MPPT and his sub-blocks are controlled by a CPU MSP_430F5438 thanks to the PWM signal; also all the measurement done by sensors are sent to the processor.

5.1. 1B1121C_MPPT

This is the general module that contains all the sub-module to generate the PDB voltage. It is composed by:

- Module 1B1121C_BUCK that convert 30V to 14V.
- Module 1B1121C_PmosDriver that drive the pMOS inside the 1B1121C_BUCK block.
- Module 1B1121C_CompHyst that use the PWM to drive the 1B1121C_PmosDriver block.
- Module 1B1121C_PWMtoVref that convert the PWM coming from the cpu to V_REF_OUT, a voltage usable by the 1B1121C_CompHyst.

5.1.1. Attributes

Signature	Documentation
-V_OUTPUT	

5.1.2. Operations

This is the voltage coming from solar cells

Signature: Vin()

This is the voltage generated by the 1B1121C_MPPT block.

It's a value around 14V.

Signature: Vout()

This is the voltage reference coming from 1B1121C_PWMtoVref block.

It is a voltage in V.

Signature: Vref()

This is the disable pin of the MPPT. In particular can be used to disable the 1B1121C_CompHyst block.

Can be enabled by setting it at the logic value of 1; it disable the 1B1121C_MPPT block if 0.

Signature: Disable()

5.1.3. Descrizione generale MPPT

L'acronimo MPPT significa Maximum Power Point Tracker, ed è un modo di chiamare determinati dispositivi di generazione della potenza che sono in grado di adattare il proprio punto di lavoro al variare delle condizioni del sistema, in modo da generare sempre la massima potenza [13].

La funzione principale dell'MPPT progettato, che in questo caso è un convertitore Buck (o step-down) MPPT, è quella di sfruttare sempre la massima potenza generata dalle celle solari in funzione dell'esposizione solare. Questo significa che migliore sarà l'esposizione e maggiore sarà la potenza disponibile e viceversa, indifferentemente dalle condizioni solari un MPPT è in grado di funzionare nel punto di massima potenza con massimo rendimento.

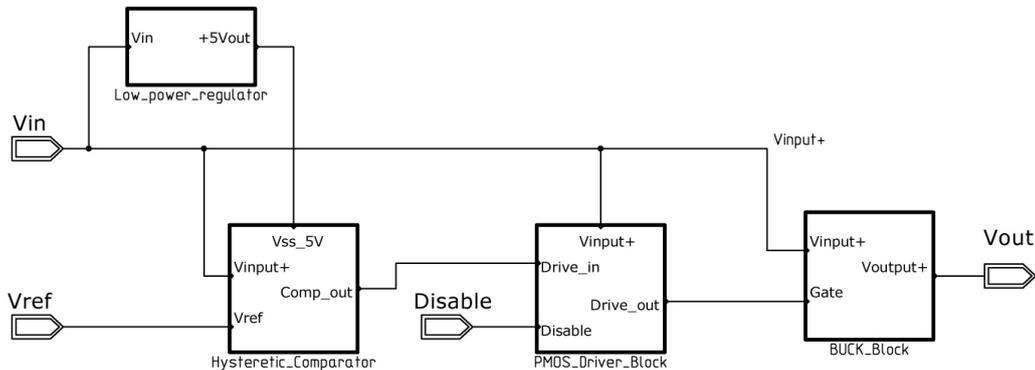
Come già mostrato nel diagramma delle classi UML i principali elementi che compongono l'MPPT progettato sono:

- 1B1121C_BUCK è il blocco che realizza la conversione dell'energia proveniente dalle celle solari in energia utile al resto del sistema.
- 1B1121C_CompHyst è il comparatore di soglia che permette l'adattamento del punto di lavoro sulla base di conti effettuati dal microprocessore, comandando il MOS del convertitore.
- 1B1121C_PWMtoVref, è il blocco che serve per adattare il PWM generato dal microprocessore alla tensione di riferimento utile al comparatore di soglia.
- 1B1121C_PmosDriver è il blocco che serve a pilotare il MOS del convertitore buck.
- 1B1121C_LowPowerReg è il regolatore di tensione che serve ad alimentare il comparatore di soglia utilizzato.

Dopo la progettazione UML si è passati a quella per mezzo del tool Mentor Graphics, che permette l'utilizzo di schemi e layout per la realizzazione fisica del sistema.

Lo schema mostrato nella figura in pagina seguente è appunto il blocco base del sistema di generazione della potenza realizzato con il tool Mentor Graphics. Schematicamente i blocchi di maggiore rilevanza sono gli stessi precedentemente

illustrati tramite UML. E' da notare come la tensione Vref sia generata da un apposito blocco chiamato 1B1121C_PWM_to_Vref che non compare però in questo schema.



La tensione V_{IN} in ingresso a questo blocco è quella proveniente dalle celle solari, mentre la tensione V_{ref} è la tensione generata dal blocco PWM to Vref, ossia quella comandata dal segnale PWM generato dal microprocessore. La tensione V_{out} è quella che verrà chiamata PDB nei blocchi di livello superiore.

E' presente un segnale di disabilitazione per il MOS driver, comandato dalla CPU, che serve a mettere in stand-by il convertitore e spegnere il driver. Questa disabilitazione può essere utile per eseguire delle diagnostiche sul sistema o in caso si voglia disabilitare il sistema.

Questo blocco MPPT è stato progettato per funzionare indipendentemente dal resto del satellite, infatti, come si può vedere nello schema, è presente un regolatore a bassa potenza, alimentato dalle celle solari, che genera i 5V necessari ad alimentare il comparatore. In breve questo significa che indifferentemente dallo stato del resto del sistema, in presenza di sole, il convertitore crea la tensione PDB.

Analizziamo ora brevemente il funzionamento del dispositivo progettato: principalmente le celle solari si comportano come un generatore di corrente in funzione delle condizioni solari (esposizione solare e quindi quantità di luce); la corrente generata dalle celle alimenta il convertitore buck che genera a sua volta una tensione che verrà chiamata PDB e che assume un valore di 14V. Il comparatore isteretico comanda il MOS-driver e quindi direttamente il convertitore Buck.

Analizzando più in dettaglio il sistema, è possibile affermare che la corrente generata dalle celle solari carica il condensatore in ingresso del convertitore Buck (all'interno del blocco BUCK_BLOCK) fino ad una soglia V_{s1} , raggiunto questo valore fissato di tensione massima, interviene il comparatore e il MOS del convertitore viene chiuso, in questo modo il condensatore di ingresso che finora si stava caricando, inizia a scaricarsi fino a raggiungere una tensione di soglia V_{s2} alla quale il MOS del convertitore si spegne, aprendosi; a questo punto ricomincia il ciclo di carica del condensatore. L'oscillazione di carica/scarica del condensatore d'ingresso genera la frequenza di commutazione f_{sw} del convertitore Buck che risulta quindi in funzione dell'ampiezza dell'isteresi del comparatore di soglia. Il ciclo di carica/scarica è continuo e sfrutta la massima potenza possibile indifferentemente dalla quantità di corrente generata dalle celle solari; infatti come precedentemente accennato, ciò che varia è il tempo che il condensatore impiega a caricarsi e questo fenomeno implica una potenza inferiore in uscita rispetto a quella che si avrebbe con tempi minori.

5.1.4. Prototipi Maximum Power Point Tracker

Dopo aver analizzato il problema delle celle solari, nota la caratteristica di tensione/potenza d'uscita della configurazione scelta, è stato progettato il convertitore Buck MMPT per effettuare la conversione della tensione. In particolare sono state valutate diverse ipotesi, ed alla fine è stata scelta la configurazione più adatta ed ottimale.

In particolare riportiamo in questo elaborato i due prototipi più interessanti e la configurazione definitiva scelta.

5.1.4.1. Primo Prototipo

Inizialmente è stato studiato e disegnato un circuito che svolgesse i compiti necessari, senza focalizzarsi più di tanto sul dimensionamento dei componenti. In particolare il primo prototipo realizzato è quello mostrato in figura 10.

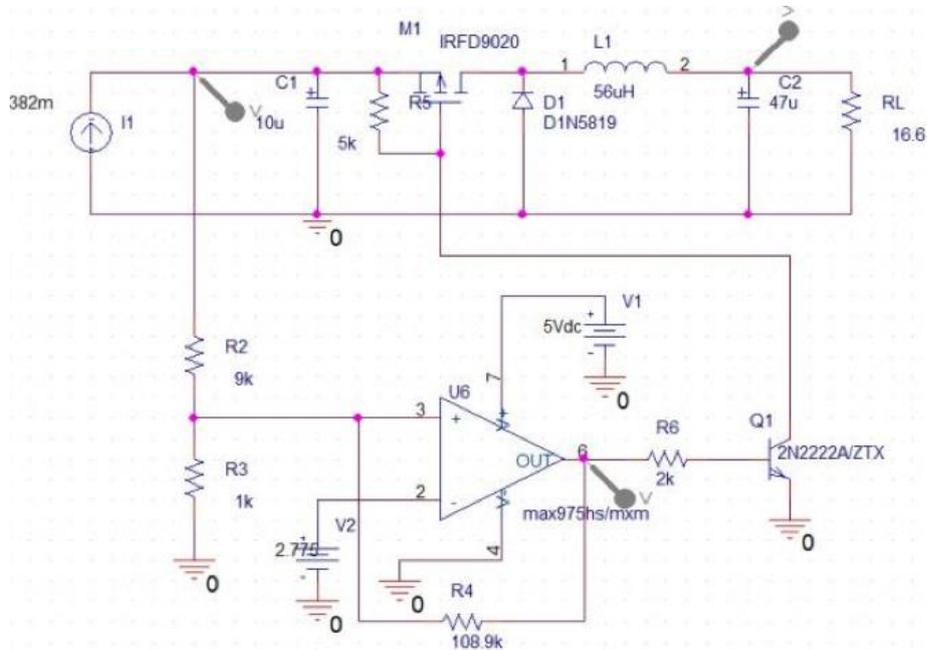


Figura 10: Primo prototipo di MPPT

Nella parte alta del circuito è realizzato il convertitore Buck, caratterizzato dall'interruttore MOS, il diodo, l'induttanza, ed i due condensatori di ingresso e uscita; nella parte bassa invece è inserito il comparatore di soglia, che per mezzo di un partitore, preleva la tensione di ingresso e la confronta con una tensione di riferimento, che sarà generata dal microprocessore.

Un'importante scelta in questo circuito è stata il tipo del MOS, infatti la prima idea era stata quella di utilizzare un MOS di tipo N, che per costruzione è più efficiente, ma per poterlo pilotare correttamente sarebbe stato necessario utilizzare una pompa di carica, o comunque dispositivi con all'interno tecnologia CMOS. Per questi motivi l'idea del MOS-n è stata scartata, ed è stato utilizzato un Mos di tipo P che viene acceso quando il suo gate viene portato a zero. Il modo più semplice per realizzare la struttura descritta è utilizzare un transistor n-p-n con l'emettitore collegato a massa.

Anche se in questo schema non è presente direttamente, il comparatore di soglia sarà alimentato da un opportuno regolatore.

5.1.4.2. Secondo Prototipo

Per realizzare un prototipo più efficiente e per tentare di risolvere i problemi legati all'accensione del MOS di tipo P, abbiamo provato ad utilizzare un MOS-n, come mostrato in figura 11.

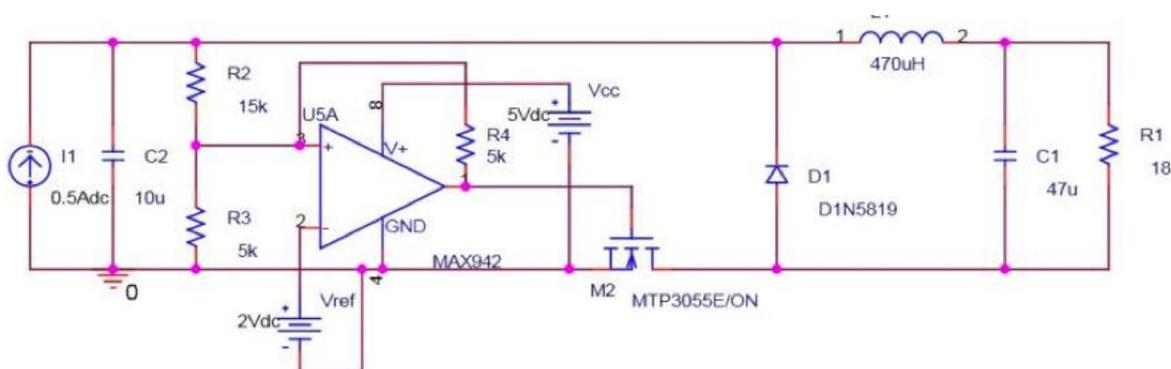


Figura 11: Secondo prototipo di MPPT

In questo caso possiamo utilizzare direttamente l'uscita del comparatore come driver del MOS, che rappresenta, rispetto alla soluzione precedente, un'ottima semplificazione.

Il problema di questo prototipo è che il GND dell'intero satellite non sarà riferito al GND della cella solare, questo creerà un problema nella generazione delle tensioni di riferimento, in quanto V_{ref} del comparatore viene referenziato alla cella solare e non alla CPU. Questo è uno dei motivi fondamentali per i quali questo prototipo è stato scartato.

5.1.4.3. Prototipo definitivo

Il prototipo definitivo, presenta molte similitudini con il primo prototipo. Infatti come possiamo vedere in figura, è stato utilizzato un P-MOS.

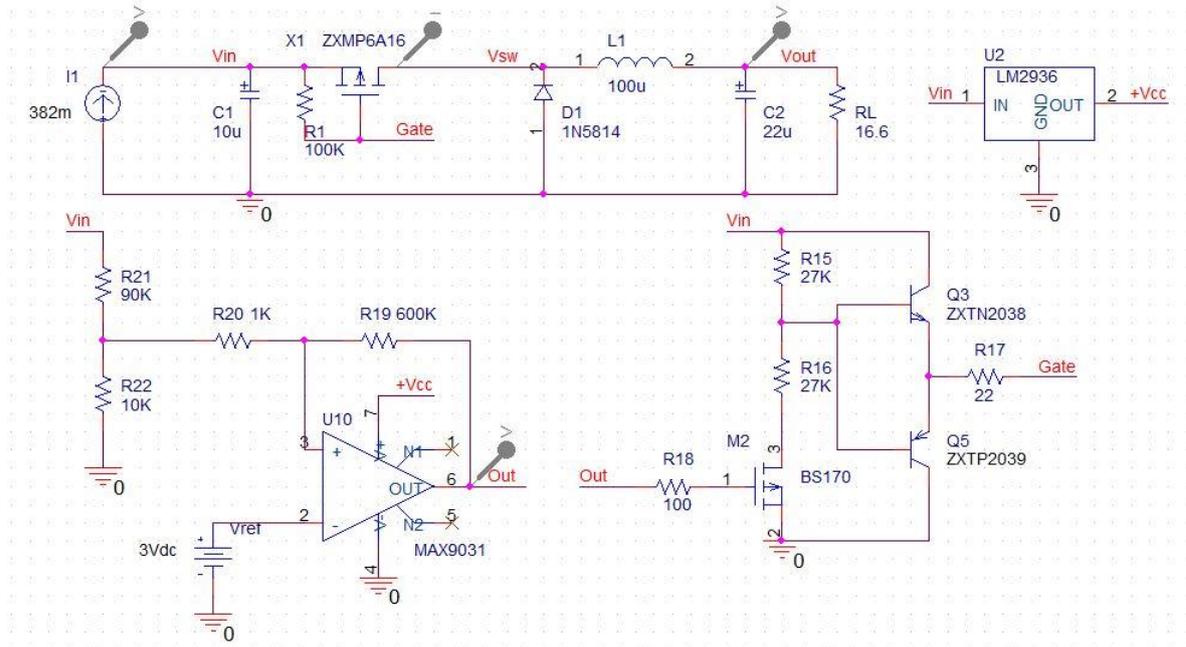


Figura 12: Prototipo MPPT definitivo

Analizziamo ora le varie parti cruciali del circuito:

- Viene risolto il problema delle masse riferite, che era sorto nel secondo prototipo.
- Il problema di pilotaggio del P-mos che era già presente nel primo prototipo è stato risolto con un'attenta analisi, e dopo aver studiato il problema è stato utilizzato un opportuno circuito, suggerito in un'application note della Texas Instrument, che serve a realizzare un pull up ed un pull down utilizzando dei transistor in modo efficiente.
- M2 che nei precedenti prototipi era un transistor, viene sostituito da un MOS in quanto il comparatore di soglia non fornisce una grande corrente d'uscita, che non sarebbe sufficiente a pilotare un transistor.
- Il convertitore di soglia è di tipo non invertente. In particolare sul pin - del comparatore si ha V_{ref} , mentre sul pin + c'è la tensione proveniente

dalle celle solari opportunamente scalata; quando la tensione sul pin + supera la tensione sul pin -, l'uscita del comparatore di soglia assume il valore di V_{CC} .

Questo significa che quando la tensione proveniente dalle celle solari raggiunge una determinata tensione, controllata dalla tensione di riferimento e quindi dalla CPU, il convertitore assume il valore logico 1 (ossia fornisce in uscita V_{CC}) questo significa che il MOS X1 si accende e la tensione delle celle solari inizia a scendere; quando la tensione proveniente dalle celle solari scende al di sotto della tensione di soglia, il comparatore tornerà al valore logico 0 e questo farà sì che X1 si spenga, aprendo il circuito con un conseguente aumento della tensione, che al raggiungimento della soglia, porterà nuovamente ad una commutazione e così via.

- Per comandare il PMOS viene utilizzato un driver non invertente, questo significa che quando l'uscita del comparatore assume il valore di V_{CC} il PMOS X1 che realizza l'interruttore nel convertitore buck è acceso, infatti il transistor Q5 è acceso e porta il segnale gate a GND.
- R21 ed R22 servono a ripartire la tensione d'ingresso (V_{max} di circa 32V) con una tensione compatibile a quella del comparatore di soglia (V_{max} di 5V).
- La resistenza R19 in retroazione fa sì che quando la tensione di uscita assume il valore di V_{CC} , la tensione sul + sarà maggiore di questa, ossia realizza l'isteresi del comparatore.
- Per rendere l'MPPT autonomo è stato aggiunto un regolatore low power che prende la tensione dalle celle solari ed indifferentemente dal suo valore e genera 5V costanti per il comparatore di soglia, in questo modo si ha una sicurezza maggiore che il convertitore continui a funzionare correttamente.

La simulazione Spice dell'andamento delle varie tensioni in funzione del tempo di questo circuito è riportata in figura ed ha mostrato che con questo tipo di circuito si ottengono i risultati desiderati.

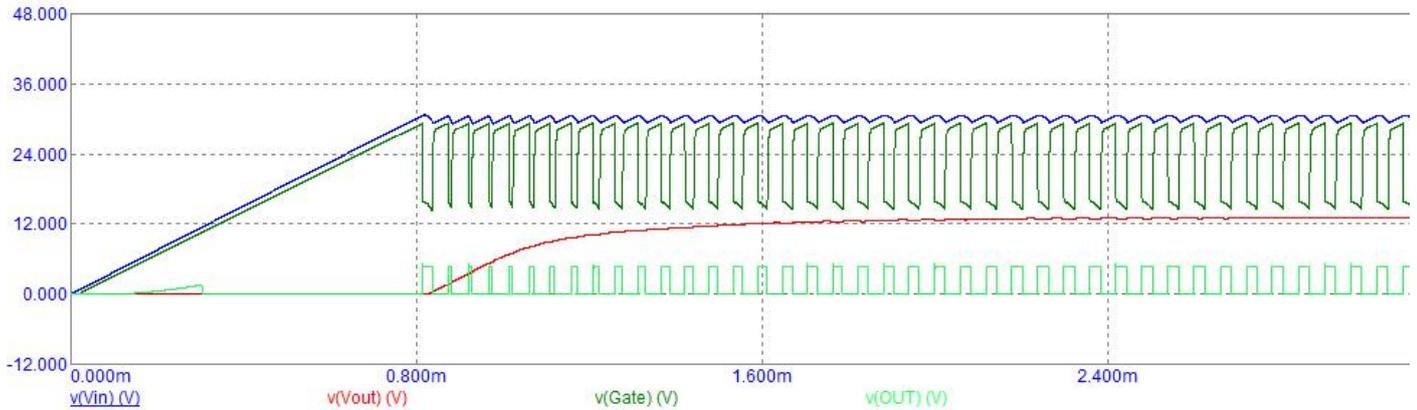


Figura 13: Andamento temporale delle tensioni di interesse dell'MPPT

In verde chiaro è rappresentata l'uscita del comparatore di soglia che viene traslata ad un più alto livello, rappresentato dalla linea in verde scuro. L'andamento della linea rossa rappresenta la tensione V_{OUT} generata dal convertitore progettato, in uscita; in questa simulazione si stabilizza ad un valore di circa 13.3V, ma verrà progettata successivamente in modo più preciso per assumere il valore di 14 V richiesto dalle specifiche. La linea blu rappresenta invece la tensione in ingresso V_{IN} proveniente dalle celle solari che ha un valore di circa 30V.

E' inoltre stato tracciato un grafico per visualizzare graficamente le potenze in gioco in questo sistema, tale grafico è mostrato nella figura seguente.

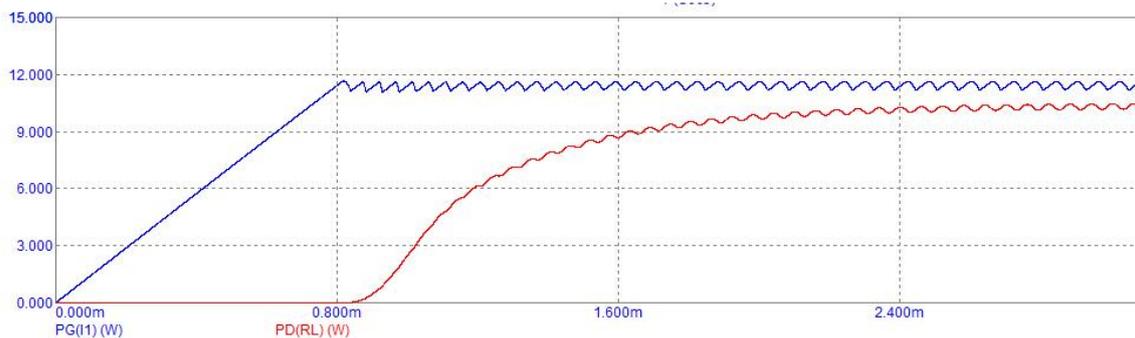


Figura 14: Grafico delle potenze dell'MPPT

La potenza generata dal generatore di corrente che simula il pannello solare è rappresentata dalla linea blu, mentre la potenza assorbita dalla resistenza di carico, ossia la potenza utile, è rappresentata dalla linea rossa.

Con un circuito di questo tipo si ottiene un andamento circa del 93%, infatti si hanno 11.4W in ingresso e 10.7W in uscita, tale rendimento è concorde con i valori di rendimento tipici che si ottengono con convertitori switching come quello progettato in questa tesi.

5.2. 1B1121C_BUCK

This block is a simple buck converter composed of an inductance, a diode, a MOS, and the input and output capacitor plus some small protection for the MOS.

Is designed to convert a voltage of around 30V in to a lower one of 14V that feeds the PDB bus.

5.2.1. Attributes

Signature	Documentation
<u>+V OUTPUT : float</u> <u>const = 14</u>	This is the output voltage generated by the 1B1121C_BUCK converter.

5.2.2. Operations

This is the input voltage that has to be converted by the 1B1121C_BUCK switching. In this case it comes from solar cells and has a maximum value around 31V.

Signature: Vinput+()

This is the signal that drive the Pmos inside the 1B1121C_BUCK block. this signal must assume a value between 10V and 20V in order to set the Pmos OFF.

If this signal is 0V or less than 10V the Pmos is ON.

Signature: gate()

This is the 14V V_OUTPUT voltage generated by the 1B1121C_BUCK converter.

Signature: Voutput+()

5.2.3. Modello matematico convertitore Buck

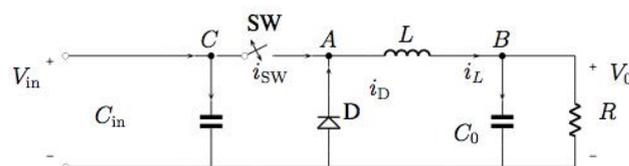
E' necessario utilizzare un convertitore con una buona efficienza, in quanto è fondamentale sfruttare al meglio tutta l'energia che viene generata dai pannelli solari. Lo scopo sarà quello di ottenere un convertitore con un'efficienza più prossima possibile ad 1. Ricordiamo che è possibile ricavare l'efficienza di un convertitore come:

$$\eta = \frac{P_{OUT}}{P_{IN}} \leq 1$$

Questo implica non solo una maggiore qualità del sistema, ma anche una riduzione della potenza sprecata, che verrà quindi dissipata e porterà ad un aumento della temperatura.

I convertitori migliori, per quanto riguarda l'efficienza, sono convertitori switching, questi oltre ad avere efficienze tra il 75% ed il 95%, presentano anche un'ottima versatilità, ossia possono funzionare con qualsiasi tensione di ingresso ed in generale possono trasformarla in una tensione d'uscita scelta.

In questa sezione andremo a studiare in modo generale il principio di funzionamento di un convertitore Buck, ossia un convertitore nel quale la tensione d'ingresso è maggiore di quella d'uscita [12].



Lo schema generale di un convertitore Buck è mostrato nella figura in alto.

I componenti di base necessari al funzionamento di questo convertitore sono:

- induttanza L che serve a immagazzinare energia e continuare ad alimentare il carico anche quando l'interruttore è aperto
- interruttore SW che viene solitamente realizzato con un MOS per ridurre i consumi.

- diodo D, chiamato anche diodo di free wheeling, che permette il circolo della corrente quando l'interruttore è aperto, mentre non la fa passare quando è chiuso.
- condensatori d'ingresso e d'uscita che servono a regolare le tensioni e forniscono anche un aiuto contro i disturbi elettromagnetici.

Nel circuito è inoltre presente R, che rappresenta la resistenza di carico del circuito. Questa resistenza, in simulazione, è facilmente ricavabile per mezzo della legge di Ohm:

$$R = \frac{V_0}{I_0} = \frac{V_0^2}{P_{in}}$$

Nella realtà questo valore dipenderà dal resto del sistema che verrà collegato a valle del convertitore, questo implica che la tensione d'uscita del Buck potrà variare assumendo valori superiori o inferiori a quelli progettati. Questo comportamento può essere spiegato teoricamente, considerando il fatto che la corrente di uscita non sarà fissa, ma potrà variare tra un massimo ed un minimo, creando quindi variazioni correlate di questa resistenza di carico. Solitamente per i conti si tende ad utilizzare la resistenza di carico massima, che è quella che si ottiene quando in uscita si ha la minima corrente desiderata.

Andiamo ora ad analizzare il convertitore: intuitivamente possiamo considerare due fasi principali, la prima presenta l'interruttore chiuso (ossia interruttore ON) ed ha una durata temporale T_{ON} , la seconda ha l'interruttore aperto (ossia interruttore OFF) con una durata temporale T_{OFF} . Il funzionamento normale del convertitore che vogliamo progettare, prevede un susseguirsi infinito di fasi T_{ON} e T_{OFF} .

Dobbiamo inoltre distinguere due casi di funzionamento importanti:

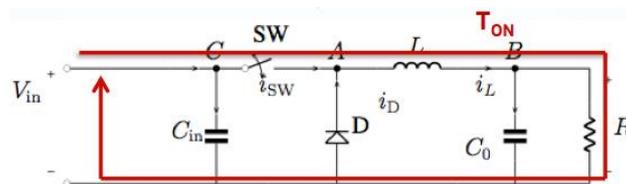
- **Funzionamento CCM:** ossia modo continuo, cioè la corrente all'interno dell'induttore non è mai nulla.
- **Funzionamento DCM:** ossia modo discontinuo, cioè la corrente all'interno dell'induttore rimane nulla per un determinato periodo di tempo. In questo

caso viene a crearsi una terza fase T_{idle} nella quale si ha una corrente nulla nell'induttore.

Il funzionamento CCM è molto più facile da controllare, rispetto al DCM, è più stabile ed inoltre presenta dei calcoli di progetto molto più semplici. Inoltre il comportamento CCM presenta dei picchi massimi di corrente nei vari componenti, più bassi rispetto al comportamento DCM, e questo comporta meno stress per i vari componenti utilizzati.

Studiamo ora il comportamento del circuito nelle due fasi fondamentali:

- **FASE ON:** l'interruttore è chiuso, la corrente circola come mostrato in figura. La corrente nell'induttanza aumenta linearmente durante tutta la fase in quanto nel punto A si ha una tensione pari a quella d'ingresso.



Durante questa fase si ha l'ingresso direttamente connesso all'induttore e perciò:

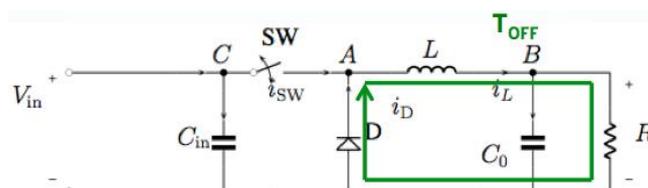
$$V_L = V_{in} - V_0$$

questo significa una pendenza della corrente pari a:

$$\frac{d i_L}{dt} = \frac{V_{in} - V_0}{L}$$

Alla fine di questa fase si ha nell'induttanza una corrente pari a I_{MAX} .

- **FASE OFF:** l'interruttore è aperto, la corrente circola come mostrato in figura grazie al diodo. Nel punto A si ha una tensione di circa 0V e la

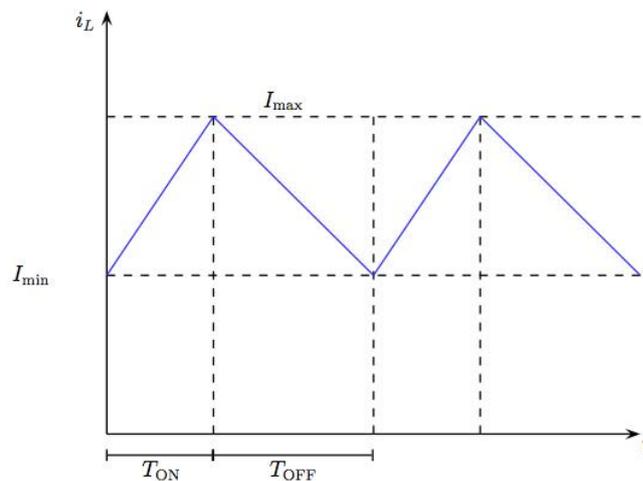


corrente nell'induttanza diminuisce linearmente.

Durante questa fase si ha una pendenza della corrente pari a:

$$\frac{d i_L}{dt} = -\frac{V_0}{L}$$

Ed al termine di T_{OFF} si ha una corrente I_{MIN}



L'andamento generale della corrente nell'induttore è mostrato in figura.

Si nota come la corrente vari tra due valori I_{MAX} e I_{MIN} che sono fissi e non variano mai, se assumiamo di lavorare in condizioni ciclo-stazionarie possiamo scrivere che:

$$\frac{V_{in} - V_0}{L} T_{ON} + \frac{-V_0}{L} T_{OFF} = 0$$

Quindi, chiamando T_{SW} il tempo somma di T_{ON} e T_{OFF} possiamo scrivere:

$$V_{in} T_{ON} - V_0 (T_{ON} + T_{OFF}) = 0 \quad \rightarrow \quad V_{in} T_{ON} = V_0 T_{SW}$$

e quindi si ottiene:

$$\frac{V_0}{V_{in}} = \frac{T_{ON}}{T_{SW}} = D$$

In particolare il tempo $\frac{T_{ON}}{T_{SW}}$ è chiamato duty cycle e viene quindi rinominato D .

Perciò un comparatore di questo tipo ci permette di avere, data una tensione di ingresso V_{in} , una tensione d'uscita desiderata V_0 con una corrente d'uscita pari a $I_0 = \frac{I_{MAX} - I_{MIN}}{2}$.

Per ottenere questi risultati è semplicemente necessario dimensionare opportunamente i vari componenti. Noto il comportamento generale del convertitore buck è utile studiare come viene effettuato il dimensionamento dei componenti nel caso si voglia lavorare in CCM.

- **INDUTTANZA:** Per quanto riguarda l'induttanza, l'analisi parte dai valori di I_{MAX} e I_{MIN} assunti dalla corrente all'interno di questa, in particolare, componendo le precedenti formule possiamo scrivere che:

$$I_{MAX} - I_{MIN} = \frac{V_0}{L} \frac{1 - D}{1/T_{SW}}$$

Si ottengono quindi due equazioni:

$$\frac{V_0}{R} = \frac{I_{MAX} + I_{MIN}}{2}$$

$$I_{MAX} - I_{MIN} = \frac{V_0}{L} \frac{1 - D}{1/T_{SW}}$$

Da queste possiamo ricavare, chiamando $\frac{1}{T_{SW}} = f_{SW}$ frequenza di switching del comparatore:

$$I_{MIN} = \frac{V_0}{R} - \frac{V_0}{2Lf_{SW}}(1 - D) = 0 \quad \rightarrow \quad \frac{1}{R} = \frac{1 - D}{2Lf_{SW}}$$

Quest'ultima equazione è fondamentale ed ha come gradi di libertà sia la frequenza di switching che il valore dell'induttanza L. Questo significa che possiamo scegliere una frequenza di lavoro, e di conseguenza ricavare il valore dell'induttanza, oppure possiamo ottenere un valore di frequenza, volendo usare una determinata induttanza. In particolare si nota che se la frequenza aumenta, si userà un'induttanza più piccola ma si avranno perdite superiori in quanto ogni ciclo perde energia, ed aumentando il numero di cicli si perde più energia.

Perciò scelta la frequenza di lavoro, possiamo facilmente ottenere il valore dell'induttanza da utilizzare, come:

$$L = R \frac{1 - D}{2f_{SW}}$$

Questo valore è detto induttanza critica in quanto se si sceglie di utilizzare un valore superiore, il comparatore smette di funzionare in modo CCM e passa in DCM.

- **DIODO:** Per utilizzare un corretto componente dobbiamo analizzare gli stress ai quali sarà sottoposto, nel caso del diodo si tratta principalmente di correnti.

La prima cosa da analizzare è la tensione inversa massima che in questo caso è pari alla tensione d'ingresso, in quanto il diodo è messo in parallelo all'ingresso.

$$V_{D,rev} = V_{in}$$

Sceglieremo quindi un diodo che sopporti questa tensione.

La corrente di picco nel diodo, che equivale alla corrente di picco presente nell'induttanza è calcolabile come:

$$I_{D,pk} = I_{0,max} + \frac{V_0(1 - D_{min})}{2Lf_{SW}}$$

Anche in questo caso dovremmo scegliere un diodo in grado di sopportare una corrente di picco di quest'ordine di grandezza.

Infine la corrente media che scorrerà normalmente all'interno del diodo, e che sarà quindi utile per la valutazione della potenza dissipata, calcolabile come:

$$I_{D,avg} = I_0 - (1 - D_{MIN})$$

- **CONDENSATORE d'INGRESSO:** Questo condensatore viene progettato considerando principalmente due fattori, la tensione di lavoro del convertitore e la corrente RMS che dovrà scorrere al suo interno. Solitamente in questo caso si usano condensatori elettrolitici che presentano valori di ESR piccoli, ma questa considerazione viene sovente tralasciata, in quanto sono usualmente presenti altri filtri in ingresso.

La sua tensione massima di lavoro sarà $V_{C,max} = V_{in,max}$.

Tenendo conto della tensione di lavoro, si sceglie il valore più piccolo di capacità per ottenere una maggior velocità di commutazione.

La corrente RMS che scorre nel condensatore è valutabile come:

$$I_{C_{in},RMS} = I_{0,max} \sqrt{D - D^2}$$

- **CONDENSATORE d'USCITA** Anche per la scelta di questo condensatore è necessario analizzare la tensione di lavoro, la corrente RMS che scorre nel condensatore, ma in questo caso la scelta del tipo di condensatore influenza notevolmente il comportamento del sistema.

La tensione di lavoro in questo caso è pari alla tensione d'uscita $V_{C_o} = V_o$, mentre la corrente RMS che dovrà scorrere nel condensatore è calcolabile come:

$$I_{C_o,rms} = \frac{I_{pk}}{\sqrt{3}} = \frac{V_o(1-D_{min})}{2Lf_{sw}} / \sqrt{3}$$

La scelta del tipo di condensatore, come ad esempio elettrolitico piuttosto che ceramico o al tantalio, andrebbe fatta considerando i componenti parassiti di questi. Nel nostro caso, verranno utilizzati solo condensatori di tipo ceramico, in quanto gli altri tipi possono avere problemi nello spazio.

- **MOS:** La scelta di questo componente viene effettuata considerando diversi fattori:

- Tensione massima tra drain e source $V_{DS} = V_{in,MAX}$.
- Resistenza del MOS acceso, ossia resistenza equivalente tra drain e source che verrà considerata durante le fasi ON:

$$r_{ds,ON} = \frac{V_{sw}}{I_{pk}}$$

Con $V_{sw} = 2\% \div 3\% V_{in,min}$.

- Tensione necessaria ad accendere il MOS, detta V_G , che dovrà essere fornita al MOS dal driver per accenderlo durante le fasi ON.

5.2.4. Progetto convertitore Buck

Il convertitore Buck progettato per mezzo del tool Design Capture di Mentor Graphics è mostrato in figura 15.

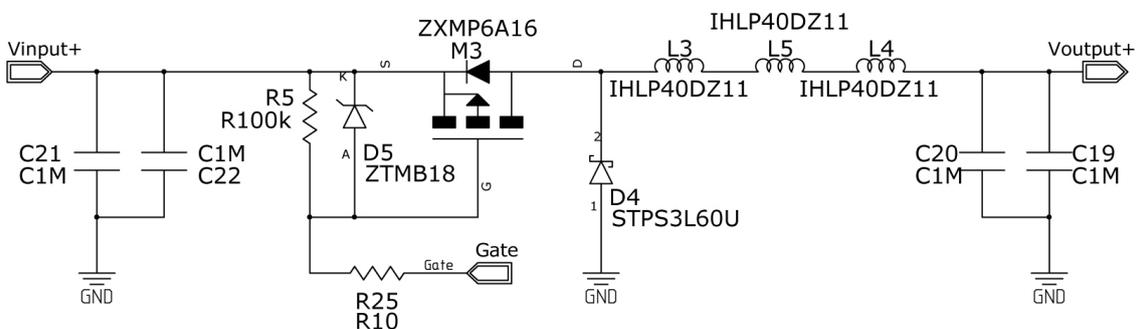


Figura 15: Schematico convertitore buck (Mentor Graphics)

Come già spiegato precedentemente questo blocco ha la funzione di convertire un'alta tensione d'ingresso di circa 30V proveniente dalle celle solari, in una più bassa di 14V, chiamata PDB (Power Data Bus), che serve ad alimentare il resto del sistema.

Lo switching progettato è un convertitore Buck classico (o Step-Down), caratterizzato dal MOS M3, il diodo D4, l'induttanza (in questo caso composta da L3, L4 ed L5) ed i condensatori di ingresso ed uscita necessari al funzionamento.

In particolare, come già accennato, invece che utilizzare un'unica induttanza da 100µH, ne sono state utilizzate tre in serie di dimensioni minori, per ottimizzare lo

spazio occupato da questa, ossia avere tre componenti piccoli e poco ingombranti, invece di uno di dimensioni maggiori.

E' necessario utilizzare un PMOS con una bassa resistenza parassita $R_{DS(on)}$ e dei tempi brevi di accensione e spegnimento, in modo da ottenere buone prestazioni, riducendo la potenza sprecata durante queste fasi. I MOS analizzati, che tra le altre cose devono sopportare una tensione sul Drain di almeno 32V provenienti dalle celle solari, sono mostrati nella tabella seguente:

Pmos	V_D	I_D	$I_{D\ pulse}$	$R_{DS(ON)}$	P_d	C_{iss}	$T_{ON \rightarrow 0}$ FF
BSP613	60V	2,9A	11,6A	130m Ω	1,8W	875pF	52ns
SUD50P06	60V	50A	80A	15m Ω	113W	4,95nF	260ns
ZXMP6A18	60V	10A	37,5A	55m Ω	10,1W	1,58nF	55ns
ZXMP6A16	60V	8,2A	27,2A	85mΩ	4,2W	1nF	10ns
NTF2955	60V	1,7A	17A	145m Ω	1W	492pF	65ns

Dopo aver analizzato questi, ed altri dispositivi MOS, è stato scelto come pMOS il ZXMP6A16 della casa produttrice Zetex, in quanto più adatto per questo progetto. Infatti presenta principalmente una piccola capacità d'ingresso, un piccolo ritardo d'accensione ed una resistenza $r_{DS(on)}$ relativamente piccola.

I valori dei principali componenti utilizzati nel progetto sono mostrati nella seguente tabella:

Componente	Valore	Tensione max.	Corrente max.
L3	33 μ H	-	4A
L4	33 μ H	-	4A
L5	33 μ H	-	4A
D4		60V	3A
C19	1 μ F	50V	-
C20	1 μ F	50V	-
C21	1 μ F	50V	-
C22	1 μ F	50V	-

I restanti componenti che costituiscono il circuito sono:

- Resistenza R5: questa resistenza servirebbe a spegnere il PMOS in un circuito classico. In questo caso, per come è stato progettato il driver del MOS, non deve essere inserita in quanto costituirebbe un inutile spreco di potenza.
- Diodo D5: è un diodo Zener di protezione che blocca la tensione V_{gs} in caso sia superiore a 18V. Normalmente, ossia quando presente un funzionamento corretto, il driver progettato fa sì che questa tensione sia fissa ad un valore di circa 15V; è comunque meglio aggiungere un'ulteriore protezione per evitare danni alla parte fondamentale del convertitore.
- Resistenza R25: serve come protezione per il Gate del MOS, ha un valore pari a 10Ω ; questa resistenza serve quindi ad annullare l'eventuale circuito LC parassita creato dalla pista.

5.3. 1B1121C_PmosDriver

This block represents a P-MOS driver for the (1B1121C_BUCK) buck converter. It is driven by the (1B1121C_CompHyst) hysteretic comparator and the CPU and can be disabled.

5.3.1. Attributes

Signature	Documentation
-disable : bool = 0	This is the disable value. If not connected the 1B1121C_PmosDriver is always enabled. If connected 0 means enabled, 1 means disabled.

5.3.2. Operations

This is the drive voltage of the PMOS.

If has 1 logic value (5V) the 1B1121C_PmosDriver generate a Drive_Out voltage equal to half of Vinput+

If has 0 logic value (0V) the 1B1121C_PmosDriver generate a Drive_Out voltage equal to Vinput+

Signature: Drive_in()

This is the voltage generated by the 1B1121C_PmosDriver in function of the Drive_in value.

Signature: Drive_Out()

This is the input voltage of the driver.

It comes from solar cells, so its nominal value is around 30V.

Signature: Vinput+()

this is the signal that disables the Pmos Driver. It is an active high signal.

If his logic value is high (5V) the 1B1121C_PmosDriver is disabled,

if his logic value is low (0V) the 1B1121C_PmosDriver is enabled.

Signature: disable()

dovesse subire danni o non funzionare correttamente, si perderebbe la generazione d'alimentazione dell'intero satellite.

Descriviamo ora in breve i vari componenti che compongono il driver:

- Q2 serve a spegnere il Pmos perché porta il valore di Drive_out a circa Vinput+, quindi il PMOS avrà una V_{gs} di circa 0V. (il MOS è ben spento).
E' stato utilizzato un transistor n-p-n BC817 che sopporta una V_{CE} di almeno 40V.
- Q1 serve ad accendere il PMOS portando la tensione Drive_out a circa la metà di Vinput+, in questo caso V_{gs} assume un valore definito da R_{26} e R_{25} , in questo caso $\frac{1}{2}$ Vinput+.
E' stato utilizzato un transistor p-n-p ZXT2039 che sopporta una V_{CE} di almeno 40V.
- La resistenza R2 servirebbe a limitare la corrente di spegnimento del PMOS, in questo caso è stata utilizzata una 0Ω in modo da avere la massima velocità.
- M2 si comporta da interruttore andando a pilotare le basi dei transistor Q1 e Q2.
- M1 serve a tener spento M2 in caso si voglia disabilitare l'MPPT, infatti ha il gate collegato al segnale di disable.

Supponendo di lavorare con una tensione Vinput+ di circa 30V possiamo usare $R_{26} = R_{25}$ in modo da accendere il PMOS con una tensione di circa 15V che è un valore perfetto.

Quando il segnale Drive_in va a 1 (5V) il MOS M2 si comporta da interruttore: il suo drain assume un valore di circa 0V, questo fa sì che grazie ad R_{26} e R_{25} ci sia una tensione sulla base del transistor Q1 di $\frac{1}{2}$ Vinput+, quindi il PMOS avrà sul gate una tensione di circa $\frac{1}{2}$ Vinput+ meno la V_{be} di Q1.

Quando il Drive_in è 0 il mos M2 è spento, quindi il suo drain è flottante e la tensione sulla base di Q2 è circa quella di Vinput+, questo fa sì che il PMOS sia spento con la tensione di V_{be} di Q2.

Il segnale Disable fissa il gate di M2 a 0V questo fa sì che il MOS M2 sia sempre spento indifferentemente dal valore di Drive_in.

Il dimensionamento delle resistenze utilizzate è stato eseguito in questo modo:

- Sono state scelte due resistenze, R_{26} ed R_{25} , di valore $22k\Omega$, infatti con questo valore viene persa una corrente pari a $0,7mA$ quando il PMOS del convertitore Buck è acceso, mentre è $0A$ quando è spento.

Tenendo conto del guadagno dei transistor utilizzati, che è $h_{fe} \approx 200$, è possibile stimare il picco di corrente di gate utilizzata per accendere il MOS, che è pari a circa $300mA$.

E' quindi possibile calcolare il tempo di carica e scarica del gate del PMOS che vale circa $0,3ns$, che è un valore sufficientemente piccolo.

Perciò utilizzando resistenze di $22k\Omega$ si ha una piccola perdita di potenza ed un'elevata velocità di pilotaggio del PMOS.

- R_{23} ha un valore di $1,5k\Omega$ perché quando il driver è disabilitato non deve assorbire troppa corrente.
- R_{24} serve come protezione del MOS, ed ha un valore piccolo, pari a 100Ω , per annullare l'eventuale circuito LC parassita creato dalla pista.
- R_6 serve a tenere sempre abilitato il PMOS driver in caso il segnale disable fosse non collegato; ha un valore molto grande, di $100k\Omega$ per limitare la dissipazione di potenza.

5.4. 1B1121C_CompHyst

This is an hysteretic comparator that uses a MAX975_comparator. It is composed by a resistive divider to adapt the input voltage to the dynamics of the comparator, the comparator and a resistance to set the hysteresis.

The comparator output is used to control the P-mos drivers as a function of the PWM calculated by the CPU; in particular is not the real PWM but a new one correctly manipulated by the 1B1121C_PWMtoVref block.

5.4.1. Attributes

Signature	Documentation
-V_drive : float	

5.4.2. Operations

This is the voltage coming from 1B1121C_PWMtoVref.
It must be a value lower than 3.3V.
Signature: Vref()

This is the positive voltage input coming from solar cells.
It assumes a maximum value of 31V.
Signature: Vinput+()

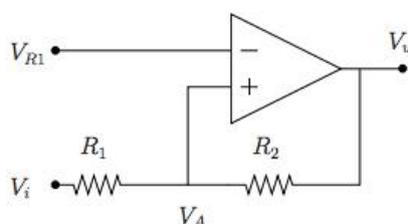
This is the output voltage generated by the comparator.
Can be 0V or 5V according to the comparison of Vinput+ and Vref.
Signature: Comp_out()

This is the supply voltage of the comparator.
Must be a 5V voltage.
Signature: Vss_5V()

5.4.3. Modello matematico del comparatore di soglia

Per realizzare un convertitore MPPT è necessario utilizzare un comparatore di soglia che sia in grado di confrontare la tensione proveniente dalle celle solari, con una di riferimento, in modo da realizzare un segnale di PWM in grado di comandare lo switch presente nel convertitore Buck. Inoltre, come vedremo in seguito, la tensione di riferimento del comparatore non sarà fissa, ma impostabile dal μ processore, in modo da far funzionare sempre il sistema nel punto nel quale si ha massima potenza.

Analizziamo ora brevemente il comportamento di un comparatore di soglia non invertente mostrato in figura, che sarà quello che andremo a progettare successivamente.



Ricordiamo come questo dispositivo sia utilissimo per fare delle scelte, infatti fissata una tensione di riferimento, al variare della tensione di ingresso, l'uscita del comparatore potrà assumere solo due valori di tensione, uno alto che chiameremo V_{OH} ed uno più basso, detto V_{OL} . In questo caso quando la tensione di ingresso diventa più piccola della tensione di riferimento, si passa da un livello alto ad uno basso della tensione in uscita.

Nell'immagine precedente, possiamo notare come sia stata realizzata una retroazione sul ramo positivo: questa serve a realizzare una doppia soglia (isteresi) ed a far sì che eventuali disturbi non facciano commutare in modo errato il comparatore.

Supponendo una resistenza d'ingresso elevatissima, e quindi una corrente quasi nulla entrante nel dispositivo, possiamo calcolare:

$$V_A = V_i \frac{R_2}{R_1 + R_2} + V_u \frac{R_1}{R_1 + R_2}$$

Supponendo che all'accensione si abbia $V_u = V_{OL}$ possiamo quindi ricavare i valori delle tensioni di soglia come:

$$V_{S1} = V_R \frac{R_1 + R_2}{R_2} - V_{OL} \frac{R_1}{R_2}$$

$$V_{S2} = V_R \frac{R_1 + R_2}{R_2} - V_{OH} \frac{R_1}{R_2}$$

Il valore dell'isteresi può essere calcolato come:

$$V_{S1} - V_{S2} = \frac{R_1}{R_2} (V_{OH} - V_{OL})$$

un altro valore interessante da conoscere è il punto medio tra le due soglie, che è facilmente ricavabile come:

$$\frac{V_{S1} + V_{S2}}{2} = V_R \frac{R_1 + R_2}{2} - (V_{OL} - V_{OH}) \frac{R_1}{2R_2}$$

5.4.4. Progetto comparatore di soglia

Dopo aver analizzato il modello matematico del convertitore di soglia, il passo successivo è stato quello di dimensionare e progettare correttamente il comparatore di soglia necessario a comandare adeguatamente lo switching utilizzato per la generazione della potenza.

Questo blocco è fondamentale per la realizzazione di un convertitore MPPT, infatti è grazie al comparatore MAX975 che si riesce a stabilizzare il buck sulla massima potenza. Il circuito realizzato è mostrato in figura.

Il comparatore scelto è il MAX975 che collegato in questo modo realizza un comparatore di soglia con isteresi. In particolare in ingresso sul ramo IN- ha la tensione di riferimento proveniente dal blocco PWM_to_Vref e quindi dalla CPU,

mentre sul ramo IN+ si ha la retroazione dell'uscita e la tensione proveniente dalla celle solari opportunamente ridotta mediante un partitore.

La tensione proveniente dalle celle solari di circa 32V, viene divisa per 11 in modo da lavorare con tensioni compatibili con il comparatore alimentato a 5V e la tensione Vref che può arrivare al massimo a circa 3V.

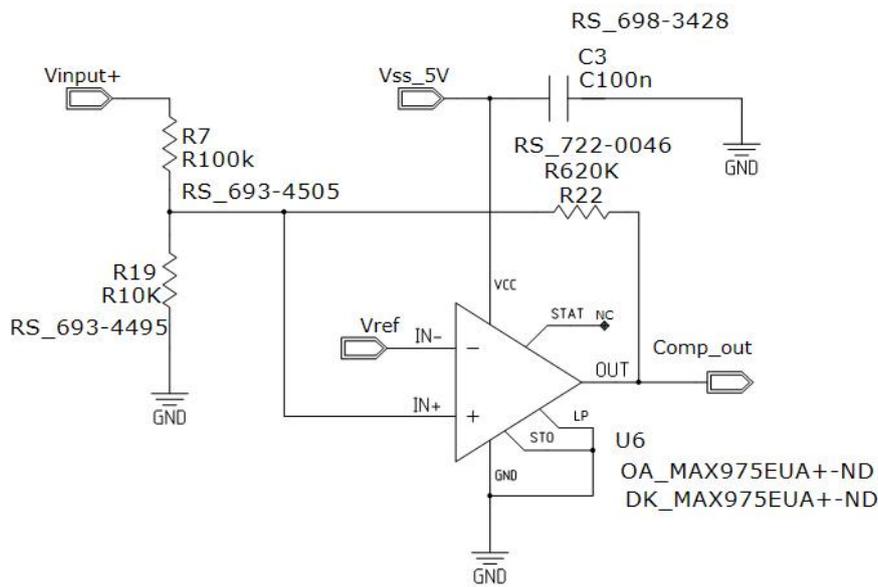


Figura 17: schema del Comparatore di soglia (Mentor Graphics)

Questa partizione è realizzata grazie a R7 ed R19, che hanno un valore grande per dissipare una piccola potenza, ossia perdere poca potenza. La resistenza R22 determina l'isteresi del comparatore, infatti si ha che:

$$V_{S1} - V_{S2} = \frac{R_{eq}}{R_{22}} (V_{OH} - V_{OL})$$

In questo caso il comparatore è di tipo rail-to-rail ed è alimentato con una tensione singola di 5V, perciò si hanno $V_{OH}=5V$ e $V_{OL}=0V$; inoltre la resistenza equivalente è calcolabile come:

$$R_{eq} = \frac{R_7 \cdot R_{19}}{R_7 + R_{19}} = 9,091k\Omega$$

Si ottiene quindi l'isteresi del comparatore come:

$$V_{S1} - V_{S2} = \frac{9,091K\Omega}{620K\Omega} \cdot 5V = 0,0733V.$$

Perciò si ha un'isteresi sulle celle solari di circa $(V_{S1} - V_{S2}) \cdot 11 = 0,8V$.

Questo significa che se si vuole far lavorare le celle solari con una tensione nominale di circa 30V, l'MPPT oscillerà tra una tensione massima di 30,4V e una minima di 29,6V. Questo valore nominale è impostabile tramite la tensione Vref: nel caso in cui una cella solare si guasti, con conseguente diminuzione di potenza e quindi della tensione massima, il microprocessore dovrà ri-elaborare la tensione Vref per creare il nuovo punto di lavoro.

Inoltre nel circuito è presente un condensatore C3 di bypass che serve al corretto funzionamento del componente MAX975.

5.5. Bk1B132E_LowSideCurrentSensor

5.5.1. Attributes

Signature	Documentation
<u>+I_MAX : float const = 0.4</u>	
<u>+SENS_CS_VOUT : float const = 16.4</u>	
<u>-INPUT_IMPEDANCE : float const = 0.1</u>	
+GBW : float const = 7.5	
<u>opamp : MAX4091_single_OPAMP</u>	

5.5.2. Operations

This signal is the negative current that has to be measured by the sensor.

Signature: I_in-()

Is the voltage supply for the MAX4091_single_OPAMP.

Nominally need s 5V voltage supply.

Signature: Val()

This is the current generated by the Bk1B132E_LowSideCurrentSensor.

Signature: I_out()

5.5.3. Progetto Sensore di Corrente Low-Side.

Per misurare la corrente proveniente dalle celle solari, sul ramo negativo, è stato progettato un sensore di corrente low-side, che tra le altre cose è più semplice ed economico. Il sensore progettato è mostrato in figura 18.

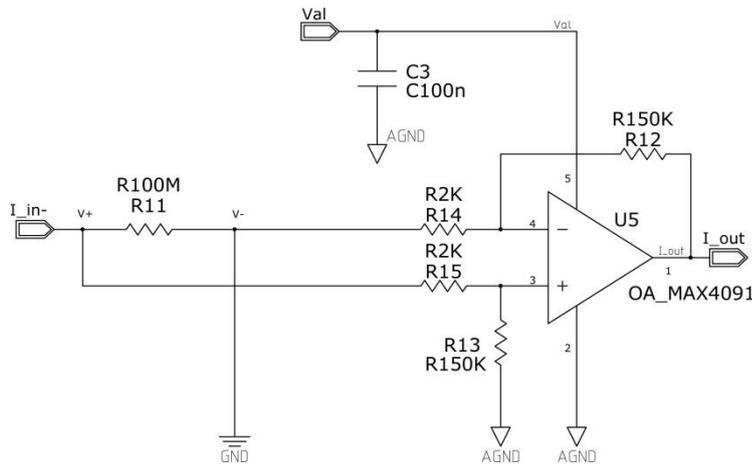


Figura 18: schematico sensore di corrente (Mentor Graphics)

L'amplificatore utilizzato è un MAX4091, un particolare comparatore rail-to-rail low power . Studiando il comportamento matematico di questa struttura, secondo la teoria degli amplificatori operazionali invertenti, si può scrivere che:

$$V_u = V_+ \frac{R_{13}}{R_{13} + R_{15}} \cdot \left(1 + \frac{R_{12}}{R_{14}}\right) - V_- \frac{R_{12}}{R_{14}}$$

perciò per fare in modo che la tensione in uscita sia pari alla differenza delle tensioni d'ingresso, bisogna fare in modo che:

$$\frac{R_{13}}{R_{13} + R_{15}} \cdot \left(1 + \frac{R_{12}}{R_{14}}\right) = \frac{R_{12}}{R_{14}}$$

quindi si può dire che:

$$\frac{R_{13}}{R_{13} + R_{15}} \cdot (1 + K) = K$$

Scegliendo $R_{12} = 150k\Omega$ e $R_{14} = 2k\Omega$ otteniamo $K = 75$. Perciò:

$$\frac{R_{13}}{R_{13} + R_{15}} \cdot 76 = 75 \quad \rightarrow \quad \frac{R_{13}}{R_{13} + R_{15}} = \frac{75}{76}$$

Quindi:

$$V_u = V_+ \cdot 75 - V_- \cdot 75$$

ossia in uscita dall'operazionale si ha esattamente la differenza tra le due tensioni di ingresso.

5.6. Bk1B131D_Voltage_Sensor

A specific implementation of a Bk1B131_Voltage_Sensor with INPUT_RANGE=40V.
The block multiplies input voltage by factor SENS_VOUT

5.6.1. Attributes

Signature	Documentation
<u>+SENS_VOUT : float</u> <u>const = 0.0643</u>	

5.6.2. Operations

Voltage input coming from solar cells (up to 40V).

Signature: Vin()

Output voltage with respect to GND

Signature: Vout()

5.7. 1B1121C_PWMtoVref

This block convert the PWM signal from the CPU to the reference voltage for the hysteretic block 1B1121C_CompHyst.

If the CPU's PWM signal goes to 0 or to his maximum value there is a protection that fix voltage reference to 2,7V.

5.7.1. Attributes

Signature	Documentation
-V_REF_OUT : float	Tensione generata in uscita in funzione del segnale PWM proveniente dalla CPU.

5.7.2. Operations

This signal must be connected to the PWM generated by the CPU.

Signature: PWM_in()

This is the voltage generated by the 1B1121C_PWMtoVref to adapt the PWM coming from the CPU for the 1B1121C_CompHyst block.

If PWM_in assumes the value of 0% or 100% the generated voltage is fixed to 2.6V.

Signature: Vref_Out()

This signal must be connected to a 3V3 voltage, in order to obtain the correct Vref_Out

Signature: Val()

5.7.3. Progetto PVM to Vref

Questo blocco serve ad adattare il PWM generato dal microcontrollore ad una tensione compatibile con il comparatore di soglia. La struttura realizzata è mostrata nella figura 19 in pagina seguente.

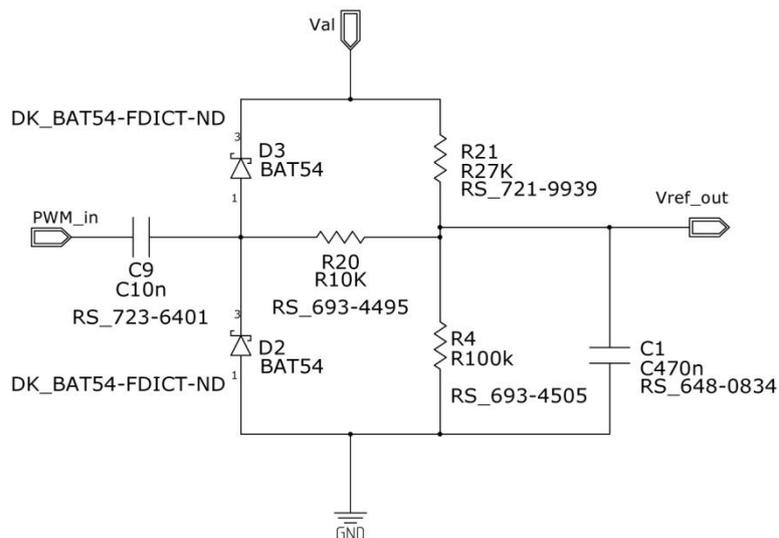


Figura 19: Schematico PWM to Vref (Mentor Graphics)

R18 ed R6 realizzano un partitore statico, che nel caso in cui manchi il segnale di PWM_in, o assuma il valore 0% o 100%, la tensione di riferimento venga fissata ad un dato valore, calcolabile come:

$$V_{ref} = \frac{R_6}{R_{18} + R_6} \cdot V_{al} = 2,6V$$

Nel normale funzionamento i due diodi ed R17 servono a modificare il valore di Vref in funzione del segnale di PWM proveniente dalla CPU.

5.8. 1B1121C_LowPowerReg

This regulator supplies only the buck converter, and it is made by the component LM2936M-5 Voltage Regulator that is a low quiescent linear regulator. This regulator is needed to make the MMPT autonomous, that means power the buck converter by the solar cells.

5.8.1. Attributes

Signature	Documentation
+I_QUIESCENT : float const = 500e-6	
-OUTPUT_VOLTAGE : float const = 5	This is the 5V generated from the LM2936M-5 Voltage Regulator.

5.8.2. Operations

This signal must be connected to the voltage coming from solar cells, in order to allow the MPPT to work independently from other regulators.

It support voltage up to 40V according to LM2936M-5 Voltage Regulator.

Signature: Vin()

This is the 5V voltage output generated by the regulator.

Signature: V_out()

5.9. Bk1B132C_Current_Sensor

A specific implementation of a Bk1B132_Current_Sensor with INPUT_RANGE=2.5A.

The block multiplies input current by factor SENS_CS_VOUT.

5.9.1. Attributes

Signature	Documentation
-----------	---------------

<u>+SENS_CS_VOUT</u> : float const = 1	
---	--

5.10. 1B1121C_3VRef

This block is a 3V voltage reference generator that uses the LM4128AMF.

It is used in 1B1121C_Primary_Switching_Buck to generate the voltage reference for the satellite.

5.10.1. Attributes

Signature	Documentation
-----------	---------------

<u>+REF_TOLERANCE</u> : float const = 0.001	This is the tolerance of the reference voltage created by LM4128AMF
---	---

<u>+I_OUT_MAX</u> : float const = 10e-3	This is the maximum output current generated by LM4128AMF, measured in ampere.
--	--

<u>+PPM_DRIFT</u> : float const = 20	
---	--

5.10.2. Operations

This is the input voltage of the 3V reference. It must be equal or lower than 5V.

Signature: V_in()

This is the 3V voltage reference generated by the LM4128AMF component.

Signature: Vref_out()

5.11. 1B1121C_3V3Power

This block is a 3,3V voltage generator that use the component LT1963AEQ-3V3 Voltage Regulator.

It is used in 1B1121C_Primary_Switching_Buck to generate the 3V3 voltage for the satellite.

5.11.1. Attributes

Signature	Documentation
<u>+V OUT : float</u> <u>const = 3.3</u>	3,3V voltage output generated by the regulator
<u>+I OUT MAX : float</u> <u>const = 1.6</u>	This is the maximum current generated by the regulator

5.11.2. Operations

This is the 3V3 power switching input. It must be at least a 5V voltage.
Signature: V_in()

This is the 3,3V voltage generated by the 1B1121C_3V3Power regulator.
Signature: V_out()

5.12. 1B1121C_5VPower

This block is a 5V voltage regulator that uses the component LT3695 Stepdown Switching.

It is used in 1B1121C_Primary_Switching_Buck to generate the 5V voltage for the satellite.

This is a particular type of regulator that can use an input up to 60V to generate 5V, so it is particular suitable for this project where the solar cells generate voltage around 30V.

5.12.1. Attributes

Signature	Documentation
<u>+V_OUT_MAX</u> : float const = 5	This is the voltage generated by the LT3695 Stepdown Switching regulator
<u>+I_OUT_MAX</u> : float const = 1	This is the maximum current generated by LT1963AEQ-3V3 Voltage Regulator, measured in ampere,

5.12.2. Operations

This is the voltage input of the 5V regulator.

It must be a voltage lower than 16V.

Signature: V_in()

This is the 5V voltage output of the regulator.

Signature: V_out()

5.13. Bk1B131C_Voltage_Sensor

A specific implementation of a Bk1B131_Voltage_Sensor with INPUT_RANGE=20V.

The block multiplies input voltage by factor SENS_VOUT.

5.13.1. Attributes

Signature	Documentation
<u>+SENS_VOUT</u> : float const = 0.1277	

6.1. Test_Bk1B81S_Power_Management_Honeycomb

6.1.1. Attributes

Signature

Documentation

-temp_buck : Bk1B133S_Temperature_Sensor
-temp_CPU : Bk1B133S_Temperature_Sensor
-temp_sat : Bk1B133S_Temperature_Sensor
-sun : Bk1B235S_Simple_Sun_Sensor
-switching : Bk1B1121CS_Primary_Switching_Buck
-hk : HK
-cpu : MSP_430F5438

6.1.2. Operations

Signature: main() : void

6.2. Testmain_Bk1B81S_Power_Management_Honeycomb

6.2.1. Attributes

Signature

Documentation

-test : Test_Bk1B81S_Power_Management_Honeycomb

6.2.2. Operations

Signature: main() : int, Code Body: test.main();

6.3. Bk1B1121CS_Primary_Switching_Buc k

6.3.1. Attributes

Signature	Documentation
<u>+ATTACHED : ushort const = 0x8000</u>	
<u>+ATTACH BIT : ushort const = 0x8</u>	
<u>+ENABLED : ushort const = 0x4000</u>	
<u>+step MIN : ushort const = 100</u>	It is the finest step range usable by the MPPT to adjust his working point. That means that if present_power is similar to old_power, the working point can be adjusted in a precise way toward the maximum power.
<u>+step MED : ushort const = 1000</u>	It is the medium step range usable by the MPPT to adjust his working point. That means that if the difference between present_power and old_power is higher than this value, but lower than step_MAX, the system will use this range to adjust the working point.
<u>+step MAX : ushort const = 3000</u>	It is the maximum step range usable by the MPPT to adjust his working point. That means that if the difference between present_power and old_power is high, the system will use this range to adjust the working point.
-ERR : signed short	This variable stores the difference between old power and present power and it's used by the function evaluate_PWM to adjust the step that the MPPT uses to reach maximum working point.
<u>+err MAX : ushort const = 2</u>	This is the value used to select the correct step to add or subtract to the old PWM to generate the new PWM using the evaluate_PWM function. In particular there are different cases based on the ERR variable that stores the module of the difference between present_power and old_power: <ul style="list-style-type: none"> • ERR > err_MAX the used step will be step_MAX. • ERR > err_MED the used step will be step_MED. • Otherwise the used step will be step_MIN that is the most accurate.

<p><u>+err_MED : ushort</u> <u>const = 1</u></p>	<p>This is the value used to select the correct step to adds or subtract to the old PWM to generate the new PWM using the evaluate_PWM function.</p> <p>In particular there are different cases based on the ERR variable that stores the module of the difference between present_power and old_power:</p> <ul style="list-style-type: none"> • ERR>err_MAX the used step will be step_MAX. • ERR>err_MED the used step will be step_MED. • otherwise the used step will be step_MIN that is the most accurate.
<p>-present_power : ushort</p>	<p>It stores the present power generated by the MPPT, calculated in function evaluate_PWM.</p>
<p>-old_power : ushort</p>	<p>It stores the power calculated in the previous cycle. It is compared to present_power in function evaluate_PWM. to decide in what direction move the system.</p>
<p>-step : ushort</p>	<p>This variable contains the increment or decrement to apply to MPPT to reach next working point. Can assume the value of step_MAX, step_MED or step_MIN according to the ERR value.</p> <p>The possible values are:</p> <ul style="list-style-type: none"> • step_MAX if ERR>err_MAX • step_MED if ERR>err_MED but less than err_MAX • step_MIN if lesser than other cases
<p>-step_PWM : signed short</p>	<p>This is the signed value that will be added to the PWM to generate the next_PWM. The sign depends on the difference between present_power and old_power and it represent how the system has to move.</p> <p>In particular if present_power>old_power the sign will be plus, otherwise minus.</p>
<p>-PWM : ushort</p>	<p>This is the PWM value generated by the function evaluate_PWM that is used by the CPU to generate the actual PWM for the drive of the MOS inside the 1B1121C_MPPT block.</p>
<p>-next_PWM : ushort</p>	<p>This variable contains the new PVM calculated and returned by the function evaluate_PWM.</p>
<p>- MPPT_tracking_enable : bool</p>	<p>This variable allows the MPPT to calculate his next working point.</p> <ul style="list-style-type: none"> • If MPPT_tracking_enable=1 the MPPT works independently and in each cycle adjusts

	<p>himself to reach the maximum power point (use case MPPT Set Auto)</p> <ul style="list-style-type: none"> • MPPT_tracking_enable =0 means that there is a fixed working point and the MPPT is stuck at that point by a command (use case MPPT Set fixed point)
-fixed_PWM : ushort	This is the value fixed by the command MPPT Set fixed point.
-upper_MPPT_limit : ushort	<p>This is the upper limit fixed by the command MPPT Set parameters.</p> <p>If this variable is set, the MPPT cannot rise the PWM over this limit.</p>
-lower_MPPT_limit : ushort	This is the low MPPT limit fixed by the command MPPT Set parameters. If this point is fixed, the MPPT cannot work under this limit.

6.3.2. Operations

Signature: boot() : void,

Code Body:

```
moduleA::D2.output();
present_power = 0;
old_power =0;
PWM =0;
next_PWM=0;
```

Signature: housekeeping(index : ushort) : void,

Code Body:

```
if (HK::ConfigRegister[CONFIG_STATUS_WORD] & (ATTACHED || ENABLED)) {
    switch (index) {
        case V_SOLAR:
            moduleB::A0.acquire(HK::housekeeping[V_SOLAR]);
            //physically connected to B1_D6_A0 - P6.0/A0
            //Capture the solar voltage and put the value in housekeeping vector at
            //V_SOLAR address
            break;
```

```

case I_SOLAR:
    moduleA::A0.acquire(HK::housekeeping[I_SOLAR]);
//physically connected to A1_D6_A0 - P6.2/A2
//Capture the solar current and put the value in housekeeping vector at //I_SOLAR
address
    break;

case V_PDB:
    moduleB::A1.acquire(HK::housekeeping[V_PDB]);
//physically connected to B1_D7_A1 - P6.1/A1
//Capture the generated PDB voltage and put the value in housekeeping vector
//at V_PDB address
    break;

case I_PDB:
    moduleA::A1.acquire(HK::housekeeping[I_PDB]);
//physically connected to A1_D7_A1 - P6.3/A3
//Capture the generated PDB current and put the value in housekeeping vector
//at I_PDB address
    break;

case MPPT_PWM:
    HK::housekeeping[MPPT_PWM]= evaluate_PWM();
    if (MPPT_tracking_enable == true) {
        moduleA::PWM.SetDutyCycle(HK::housekeeping[MPPT_PWM]);
    } else {
        moduleA::PWM.SetDutyCycle(fixed_PWM);
    }
    break;
}
}

```

This function decodes command received by the system and executes the correct operation for each type of command. In particular the commands that can be used are:

- `CMD_SET_CONFIGURATION` enables the MPPT (use case Attach solar panel&Enable solar panel)
- `CMD_RESET_CONFIGURATION` disables the MPPT (use case Detach solar panel&Disable solar panel)
- `CMD_MPPT_RESET` that actuate the use case MPPT Reset
- `CMD_MPPT_SET_PARAMETERS` that actuate the use case MPPT Set parameters
- `CMD_MPPT_SET_FIXED_POINT` that actuate the use case MPPT Set fixed point
- `CMD_MPPT_SET_AUTO` that actuate the use case MPPT Set Auto

Signature: `interpret(command : t_Commands, error : ushort) : void,`

Code Body:

```
unsigned short *limits;
switch (command) {
    case Bk1A1_Common_Codes::CMD_SET_CONFIGURATION:
        if (HK::ConfigRegister[CONFIG_STATUS_WORD].test(ATTACHED &
ENABLED))
            enable_MPPT();
        else
            disable_MPPT();
        break;

    case Bk1A1_Common_Codes::CMD_RESET_CONFIGURATION:
        if (HK::ConfigRegister[CONFIG_STATUS_WORD].test(ATTACHED &
ENABLED))
            disable_MPPT();
        break;
```

```

case Bk1A1_Common_Codes::CMD_MPPT_RESET:
//use case "reset MPPT"
    present_power = 0;
    old_power =0;
    ERR = 0;
    fixed_PWM =0;
    lower_MPPT_limit = 0;
    upper_MPPT_limit =0;
    MPPT_tracking_enable = true;
break;

case Bk1A1_Common_Codes::CMD_MPPT_SET_PARAMETERS:
//use case "set working parameters"
    limits = (unsigned short *) HK::buffers.get(command);
    // rende un puntatore a byte
    upper_MPPT_limit = limits[0];
    lower_MPPT_limit = limits[1];
break;

case Bk1A1_Common_Codes::CMD_MPPT_SET_FIXED_POINT: //use case
"set fixed point"
    limits = (unsigned short *) HK::buffers.get(command);
    // rende un puntatore a byte
    fixed_PWM = limits[0];
    MPPT_tracking_enable = false;
break;

case Bk1A1_Common_Codes::CMD_MPPT_SET_AUTO:
    //use case "set auto-tracking mode"
    MPPT_tracking_enable = true;
break;
}

```

Signature: supervise() : void

This is the function used each cycle to calculate actual PWM.

It checks the present_power and the old_power and put the difference between them in the ERR variable, then it uses this value to select the correct step range between step_MIN, step_MED and step_MAX. After that, the PWM is added or subtracted to the previous selected step_PWM and this value is returned by this function as actual PWM.

If some working limit are fixed by the commands MPPT Set parameters or MPPT Set fixed point, this function makes sure that they are respected.

Signature: evaluate_PWM() : ushort,

Code Body:

```
//evaluate the new present power
present_power = HK::housekeeping[I_PDB]*(HK::housekeeping[V_PDB]);

//check the difference between present power and old power
ERR = old_power - present_power;

if (ERR < 0) ERR = - ERR; //if negative turn it in positive

if (ERR > err_MAX) {           //far away from optimal working point
    step = step_MAX;
} else if (ERR > err_MED) {
    step = step_MED;
} else {                       //if near maximum power point
    step = step_MIN;
}

if (present_power > old_power) {
    step_PWM = step;
} else {
```

```

    step_PWM = -step;
}
old_power = present_power;

next_PWM = PWM + step_PWM;
if ( lower_MPPT_limit != 0) {
    if (next_PWM < lower_MPPT_limit) {
        next_PWM = lower_MPPT_limit;
    }
}

if (upper_MPPT_limit != 0) {
    if (next_PWM > upper_MPPT_limit) {
        next_PWM = upper_MPPT_limit;
    }
}

return next_PWM;

```

This function enables the MPPT converter using the disable line of the 1B1121C_PmosDriver block.

Until the MPPT is enabled, the PDB voltage is generated.

Signature: enable_MPPT() : void,

Code Body:

```

moduleA::D9.set();
// the physical disable pin is connected to bus line A1_D9_EN_PWM2

```

This function disables the MPPT converter, using the disable line of the 1B1121C_PmosDriver.

If the MPPT is disabled, it will not generate the PDB voltage.

Signature: disable_MPPT() : void,

Code Body: moduleA::D9.reset();

```

// the physical disable pin is connected to bus line A1_D9_EN_PWM2

```

6.3.3. Template parameters

Signature	Documentation
HK	
V_SOLAR	Memory location (for housekeeping and other purposes) where to store the measured voltage generated by the1B112 Solar Panel Subsystem.
I_SOLAR	Memory location (for housekeeping and other purposes) where to store the measured current generated by the1B112 Solar Panel Subsystem.
V_PDB	Memory location (for housekeeping and other purposes) where to store the measured voltage generated by the1B1121C_Primary_Switching_Buck.
I_PDB	Memory location (for housekeeping and other purposes) where to store the measured current generated by the1B1121C_Primary_Switching_Buck.
MPPT_PWM	Memory location (for housekeeping and other purposes) where to store the generated PWM that has to be used by the CPU to allow the 1B1121C_Primary_Switching_Buck to work at his MPP or in an assigned point.
moduleA	CPU interface module used
moduleB	CPU interface module used
CONFIG_STATUS_WORD	

6.4. Bk1B235S_Simple_Sun_Sensor

6.4.1. Operations

Signature: housekeeping(index : ushort) : void,

Code Body:

```
switch (index) {  
case SUNSENSOR:  
switch (CHANNEL) {  
case 0:  
module::A0.acquire(HK::housekeeping[SUNSENSOR]);  
break;  
case 1:  
module::A1.acquire(HK::housekeeping[SUNSENSOR]);  
break;  
}  
}  
}
```

6.5. Bk1B133S_Temperature_Sensor

6.5.1. Operations

Signature: housekeeping(index : ushort) : void,

Code Body:

```
switch (index) {  
case TEMPERATURE:  
switch (CHANNEL) {  
case 0:  
module::A0.acquire(HK::housekeeping[TEMPERATURE]);  
break;  
case 1:  
module::A1.acquire(HK::housekeeping[TEMPERATURE]);  
break;  
}  
}  
}
```

6.6. HK_fields_1B8_Honeycomb

Codes for housekeeping of 1B8 Power management Tile.

Each code identifies the location of the corresponding value in the 1B8 housekeeping vector

6.6.1. Attributes

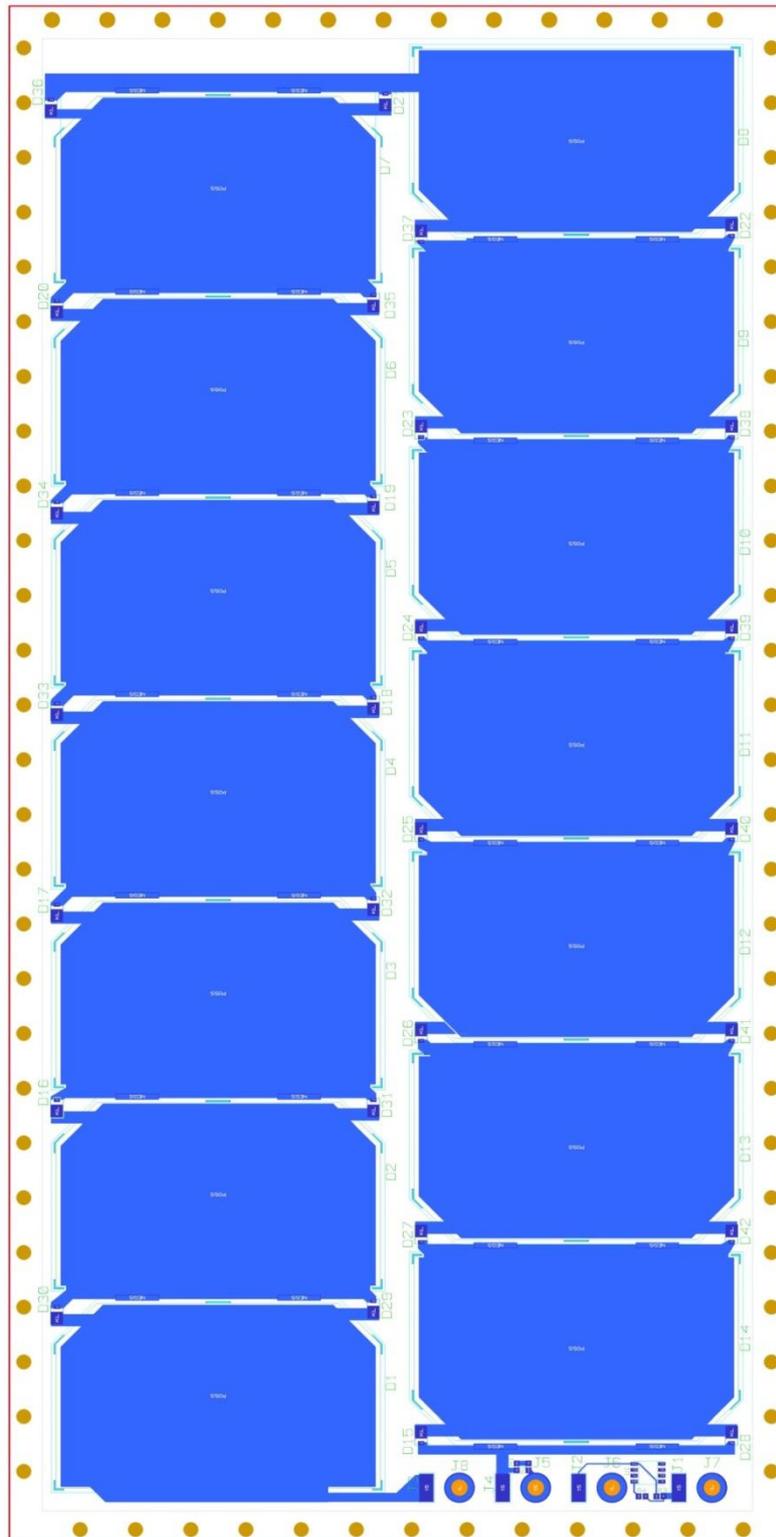
Signature	Documentation
-SUN_SENSOR	Housekeeping code for measured sun level. It is associated with use case Get sun.
-SOLAR_VOLTAGE	Housekeeping code for measured voltage across solar panel. It is associated with use case Get solar panel voltage(see 1B112C project).
-SOLAR_CURRENT	HK code for measured average current from solar panel. Associated with use case Get solar panel current (see 1B112C project)
-PRIMARY_VOLTAGE	Housekeeping code for measured voltage at the output of primary switching buck. It is associated with use case Get PDB voltage (see 1b112C project)
-PRIMARY_CURRENT	HK code for measured average current generated by primary switching. Associated with use case Get solar panel current(see 1B112C project).
-MPPT_DUTYCYCLE	This is the index of the duty cycle calculated by the software of the MPPT. It is used by the CPU to generate the PWM for the MPPT. It is associated to use case Get PWM (see 1b112C project).
-TEMPERATURE_BUCK	HK code for measured temperature close to the buck converter on the internal PCB. Associated with use case Get buck temperature(see 1B81 project).
-TEMPERATURE_CPU	HK code for measured temperature close to the CPUs. Associated with use case Get CPU temperature(see 1B81 project).
-TEMPERATURE_SAT	HK code for measured temperature inside the satellite, far away from buck converter and CPUs. Associated with use case Get SAT temperature(see 1B81 project).

+LENGTH_HOUSEKEEPING	The length of the housekeeping vector, in number of t_sensor elements. It also indicates the number of measurements in the housekeeping vector.
----------------------	--

APPENDICI

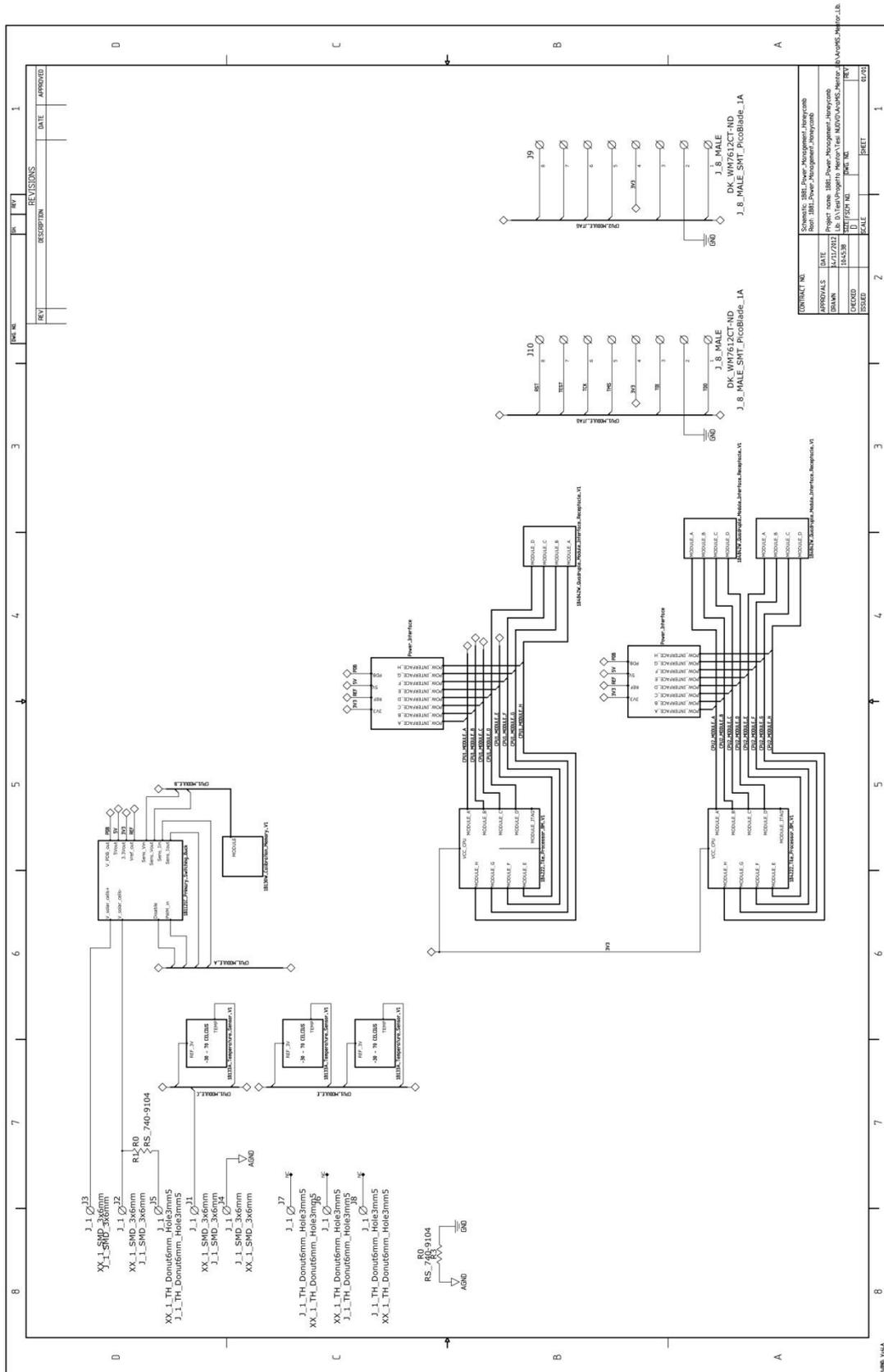
A. Layout 1B81_Solar_panel_honeycomb

1. PCB esterno

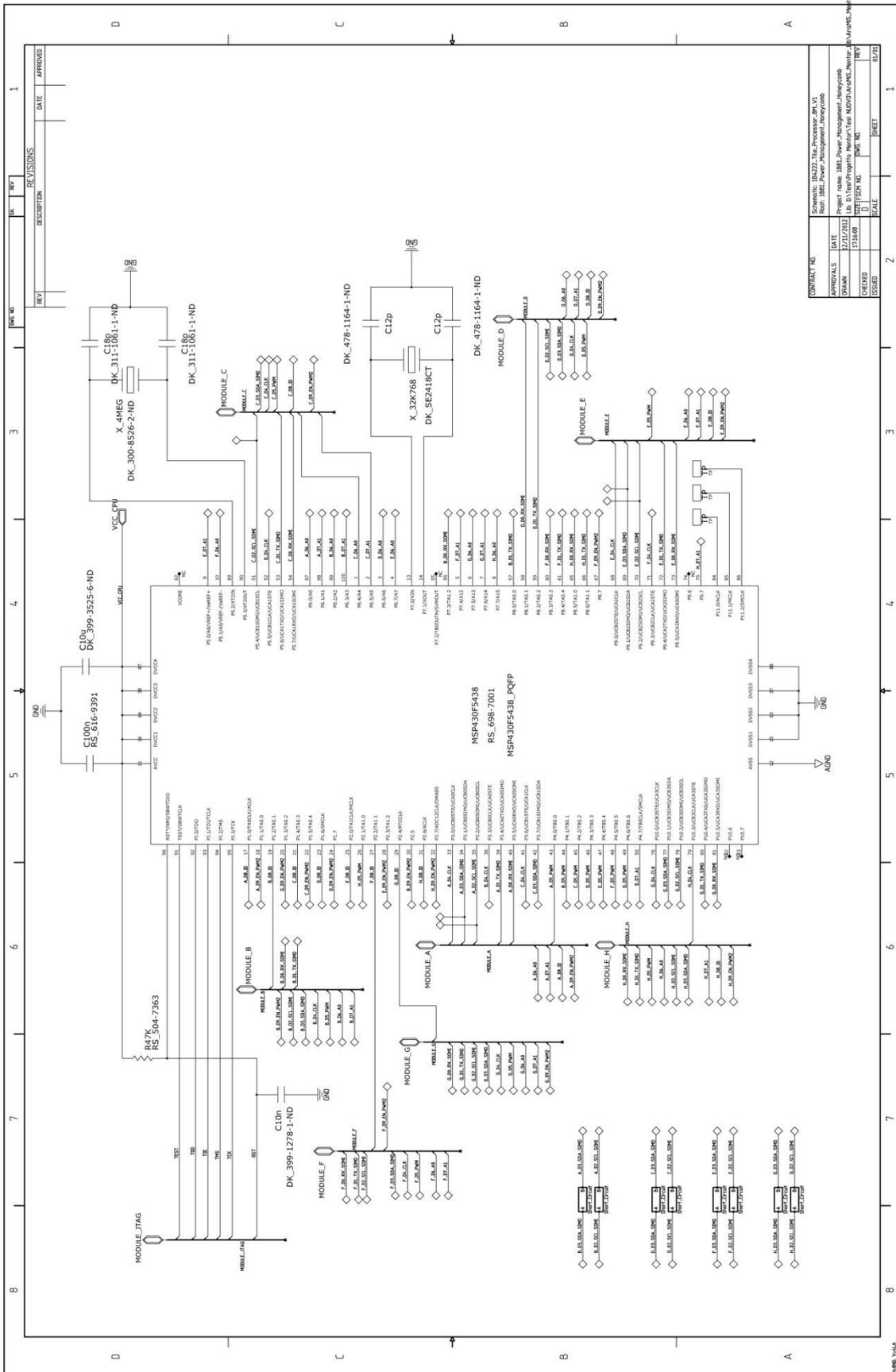


B. Schemi elettrici

1. 1B81_power_management_honeycomb



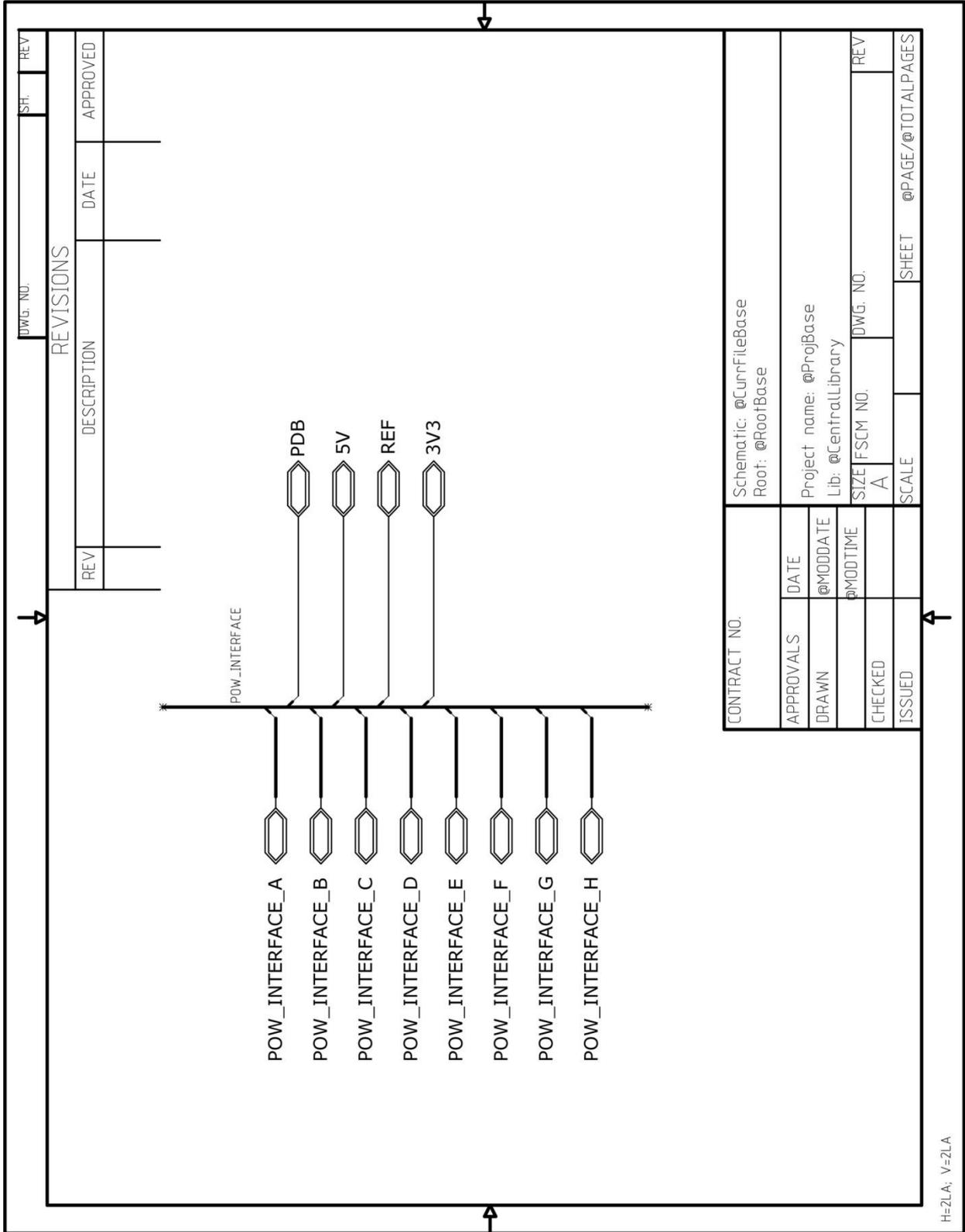
2. 1B4222_Tile_Processor_8M_V1.



REV	DESCRIPTION	DATE	APPROVED

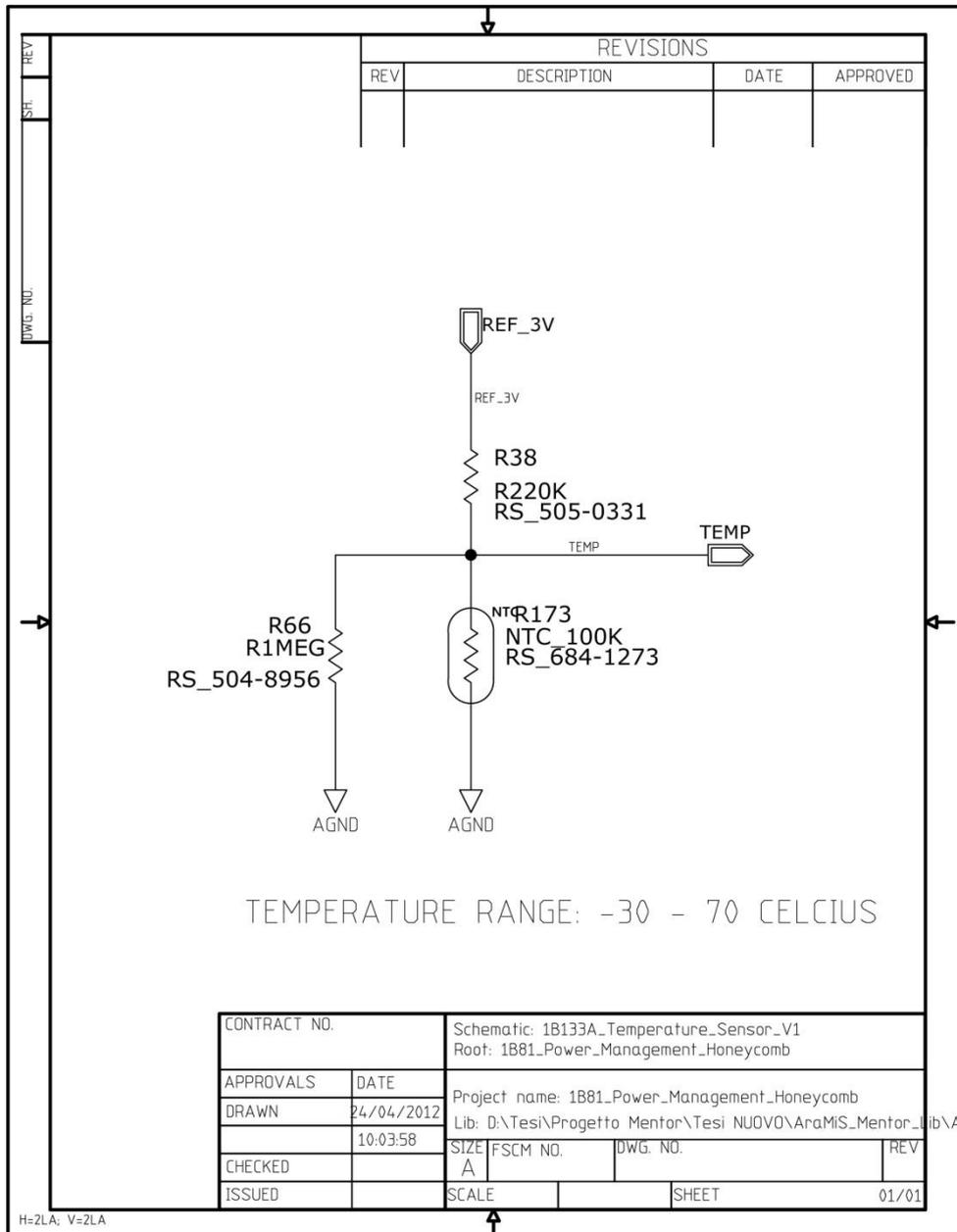
CONTINUTI N°	Schematic: 1B4222_Tile_Processor_8M_V1
REVISIONI	Rev: 1B4222_Power_Management_2009y08
DATA	17/11/2002
PROGETTA	Lib: 13_VariProgetti_Memorie_Vel: MD070_Archives_Memorie_8M_V1_0805_0809_0810_0811_0812_0813_0814_0815_0816_0817_0818_0819_0820_0821_0822_0823_0824_0825_0826_0827_0828_0829_0830_0831_0832_0833_0834_0835_0836_0837_0838_0839_0840_0841_0842_0843_0844_0845_0846_0847_0848_0849_0850_0851_0852_0853_0854_0855_0856_0857_0858_0859_0860_0861_0862_0863_0864_0865_0866_0867_0868_0869_0870_0871_0872_0873_0874_0875_0876_0877_0878_0879_0880_0881_0882_0883_0884_0885_0886_0887_0888_0889_0890_0891_0892_0893_0894_0895_0896_0897_0898_0899_0900
VERIFICATA	17/11/02
PROGETTA	
ISSUE	
SCALE	
SHEET	1
TOT	1/20

3. Power_interface.

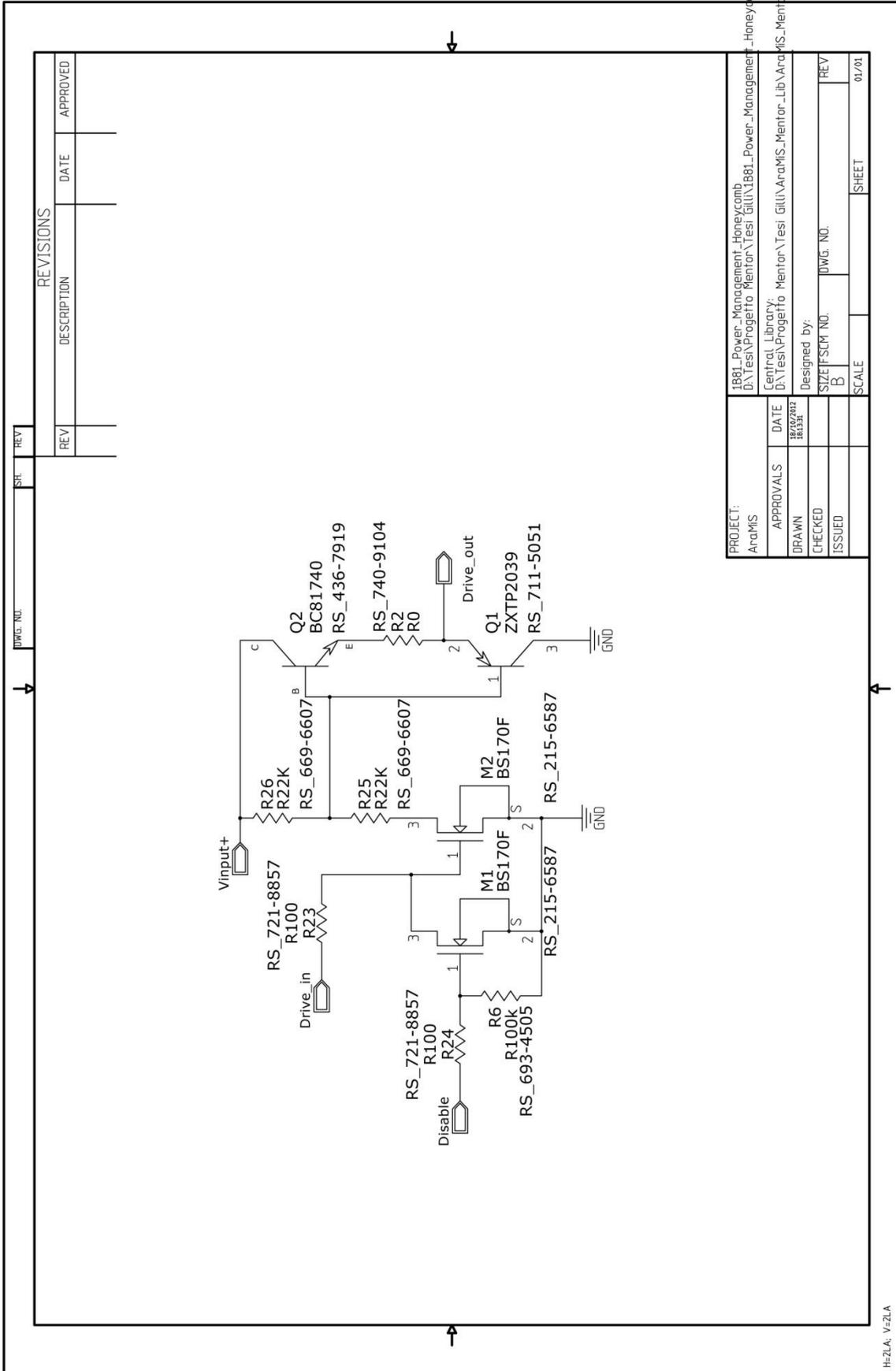


H=2LA; V=2LA

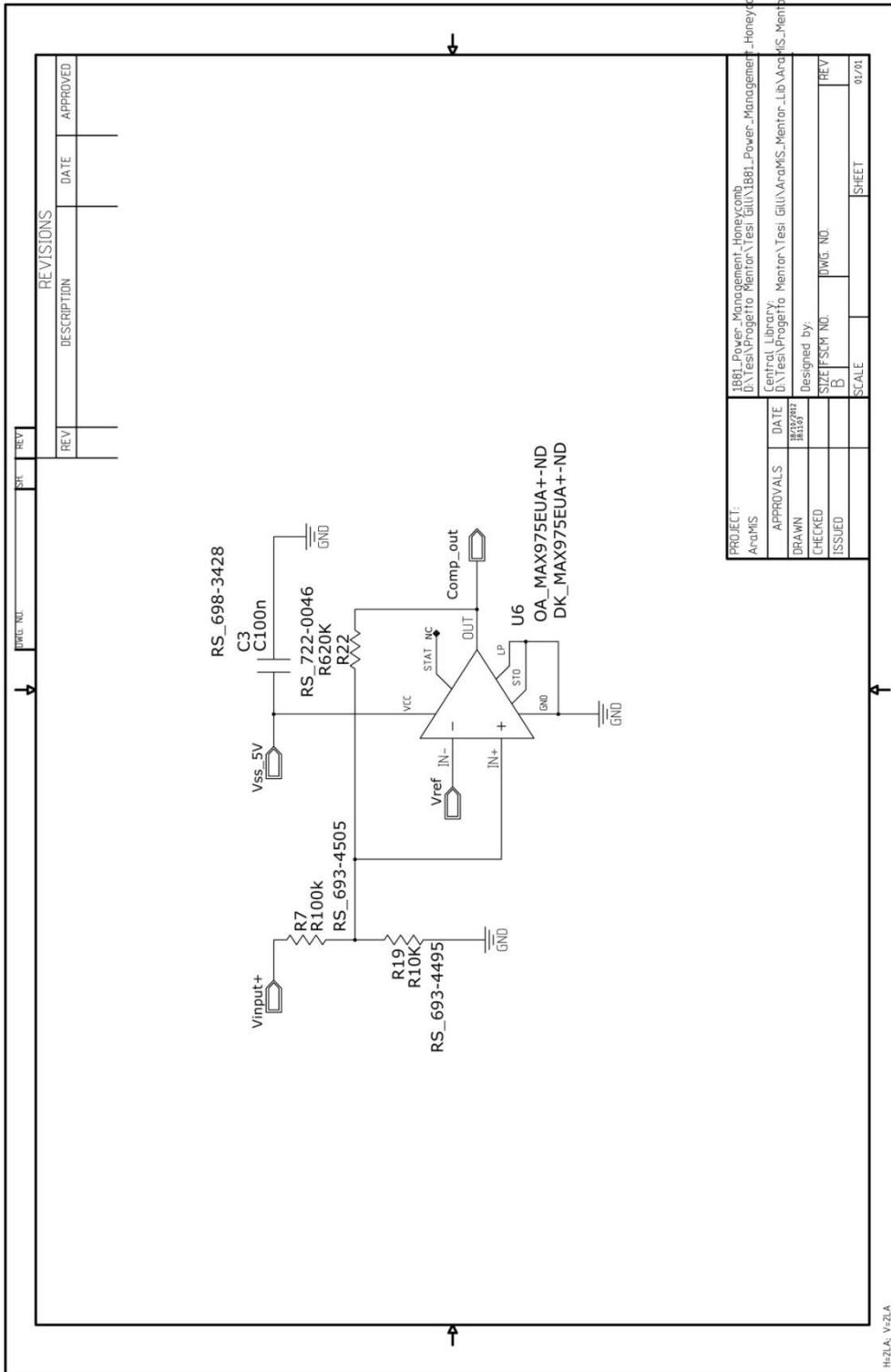
4. 1B133A_Temperature_Sensor_V1.



7. Pmos_Driver_Block.



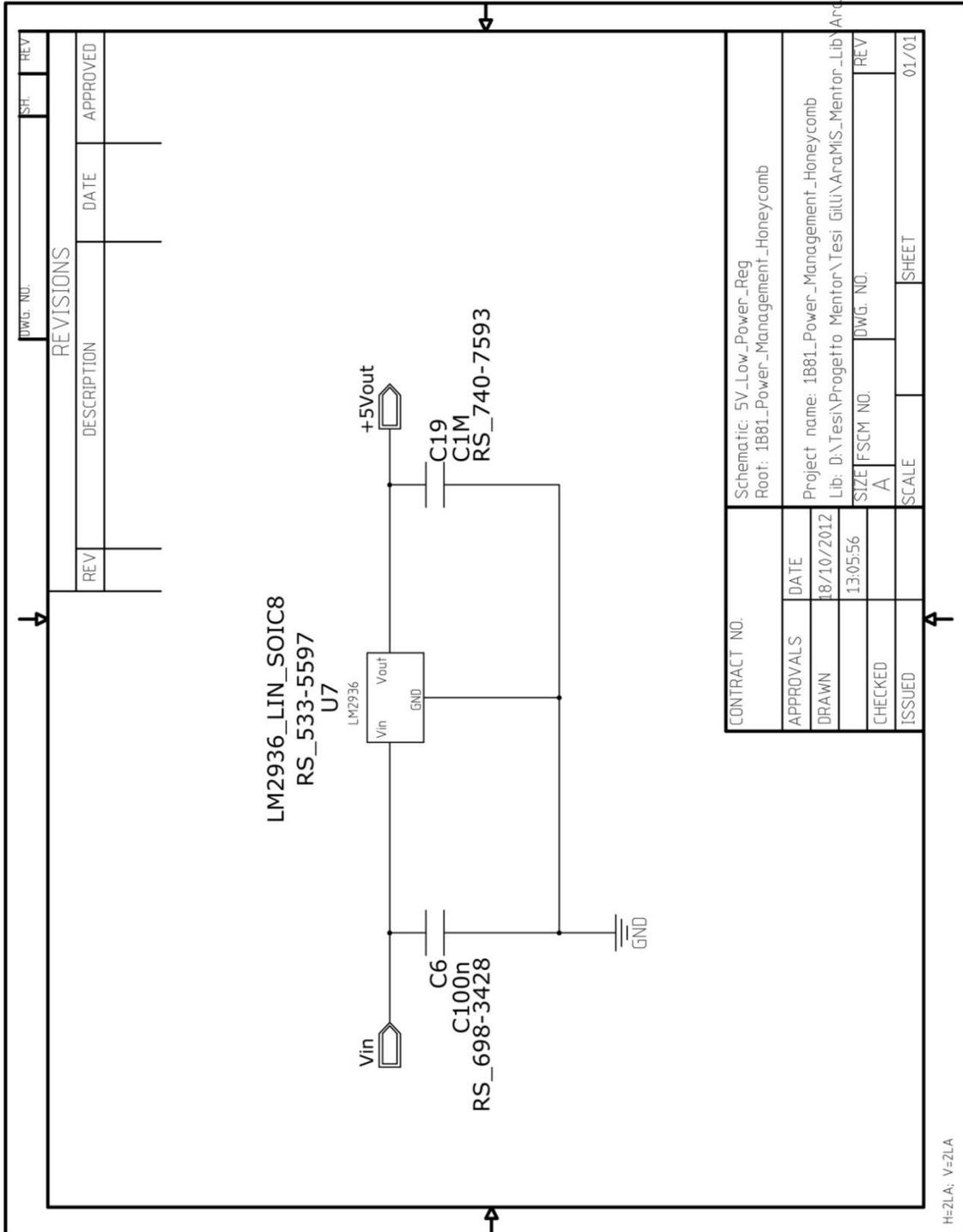
8. Hysteretic_Comparator.



REV. NO.			REV.		
REV	DESCRIPTION	DATE	REV	DESCRIPTION	DATE

PROJECT:	1881_Power_Management_Honeycomb				
APPROVALS	DATE	DA\Tes\Progetto_Mentor\Tes_Gil\1881_Power_Management_Honeycomb			
DRAWN	18/07/2012	Gentrel_Liberty			
CHECKED		DA\Tes\Progetto_Mentor\Tes_Gil\1881_Power_Management_Honeycomb			
ISSUED		DA\Tes\Progetto_Mentor\Tes_Gil\1881_Power_Management_Honeycomb			
		DESIGNED BY:			
		SIZE	FSM	NO.	UWG
		B			
		SCALE		SHEET	02/01

9. Low_power_regulator.

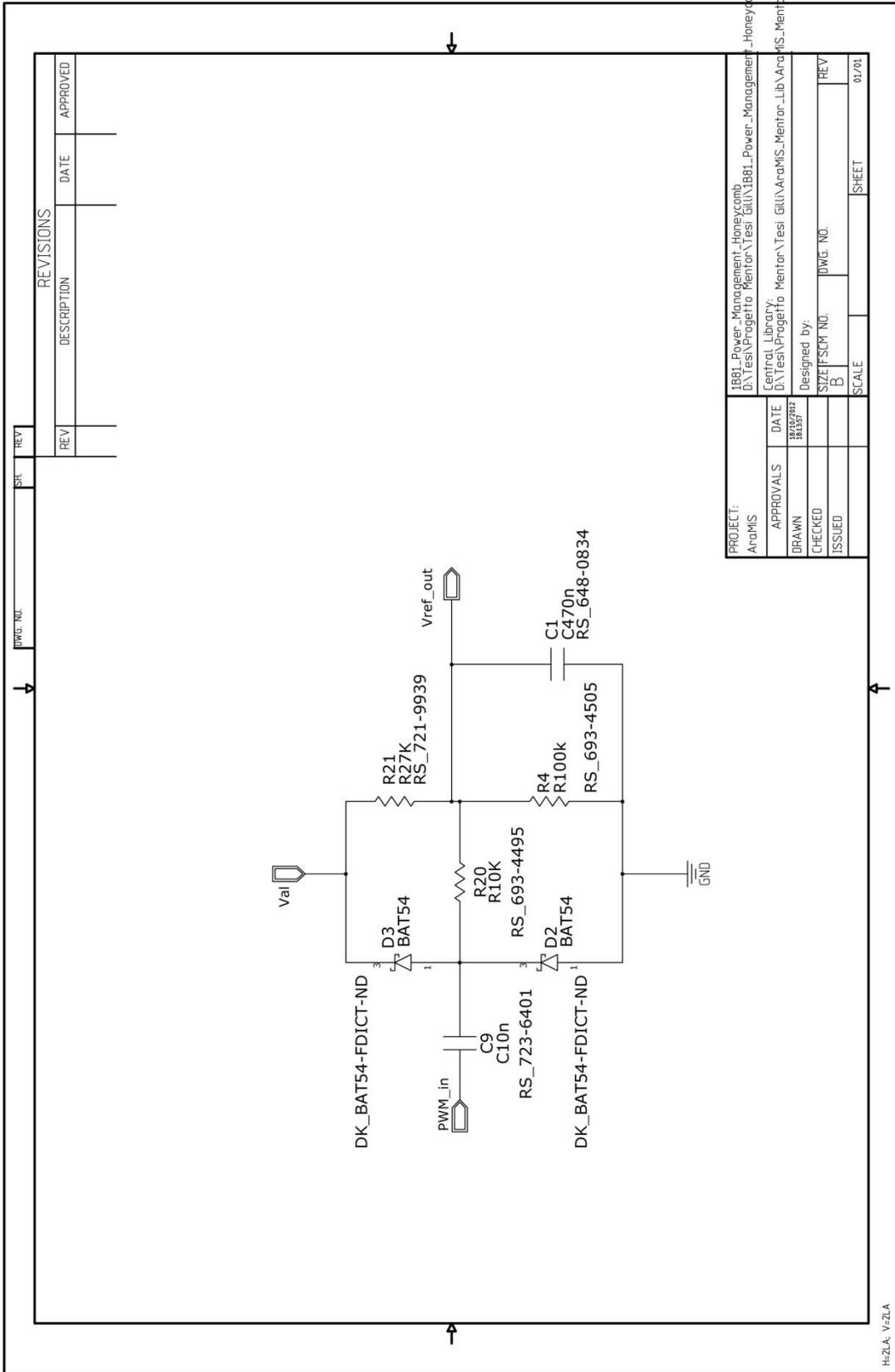


DWG. NO.		SH.	REV.
REVISIONS		DATE	APPROVED
REV	DESCRIPTION	DATE	APPROVED

CONTRACT NO.		Schematic: 5V_Low_Power_Reg Root: 1B81_Power_Management_Honeycomb	
APPROVALS	DATE	Project name: 1B81_Power_Management_Honeycomb	
DRAWN	18/10/2012	Lib: D:\Tesi\Progetto Mentor\Tesi Gil\AraMIS_Mentor_Lib\AraMIS_Mentor_Lib.lmc	
CHECKED	13:05:56	SIZE	FSCM NO.
ISSUED		A	DWG. NO.
		SCALE	
		SHEET	01/01

H=2LA; V=2LA

10.PWM_to_Vref.

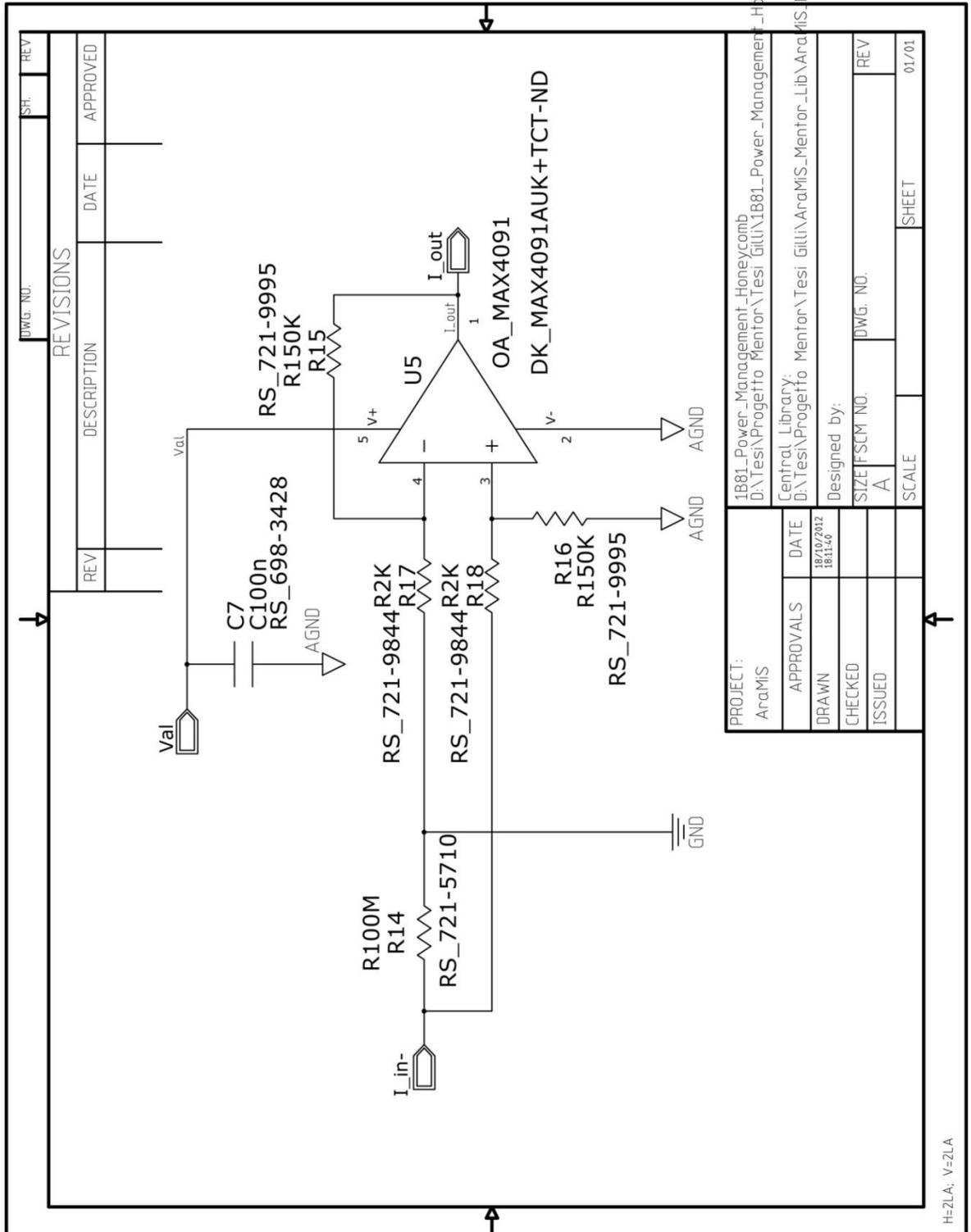


REVISIONS			
REV	DESCRIPTION	DATE	APPROVED

PROJECT: ArqMS		1881_Power_Management_Honeycomb D:\Tesi\Progetto_Mentor\Tesi_GliU\1881_Power_Management_Honeycomb	
APPROVALS	DATE	Central Library: D:\Tesi\Progetto_Mentor\Tesi_GliU\ArqMS_Mentor_Lib\ArqMS_Mentor_Lib.linc	
DRAWN	18/03/2017	DESIGNED BY:	
CHECKED		SIZE/SCM NO.	DWG. NO.
ISSUED		SCALE	SHEET
			01/01

H=2LA_V=2LA

11.Low_side_current_sensor.

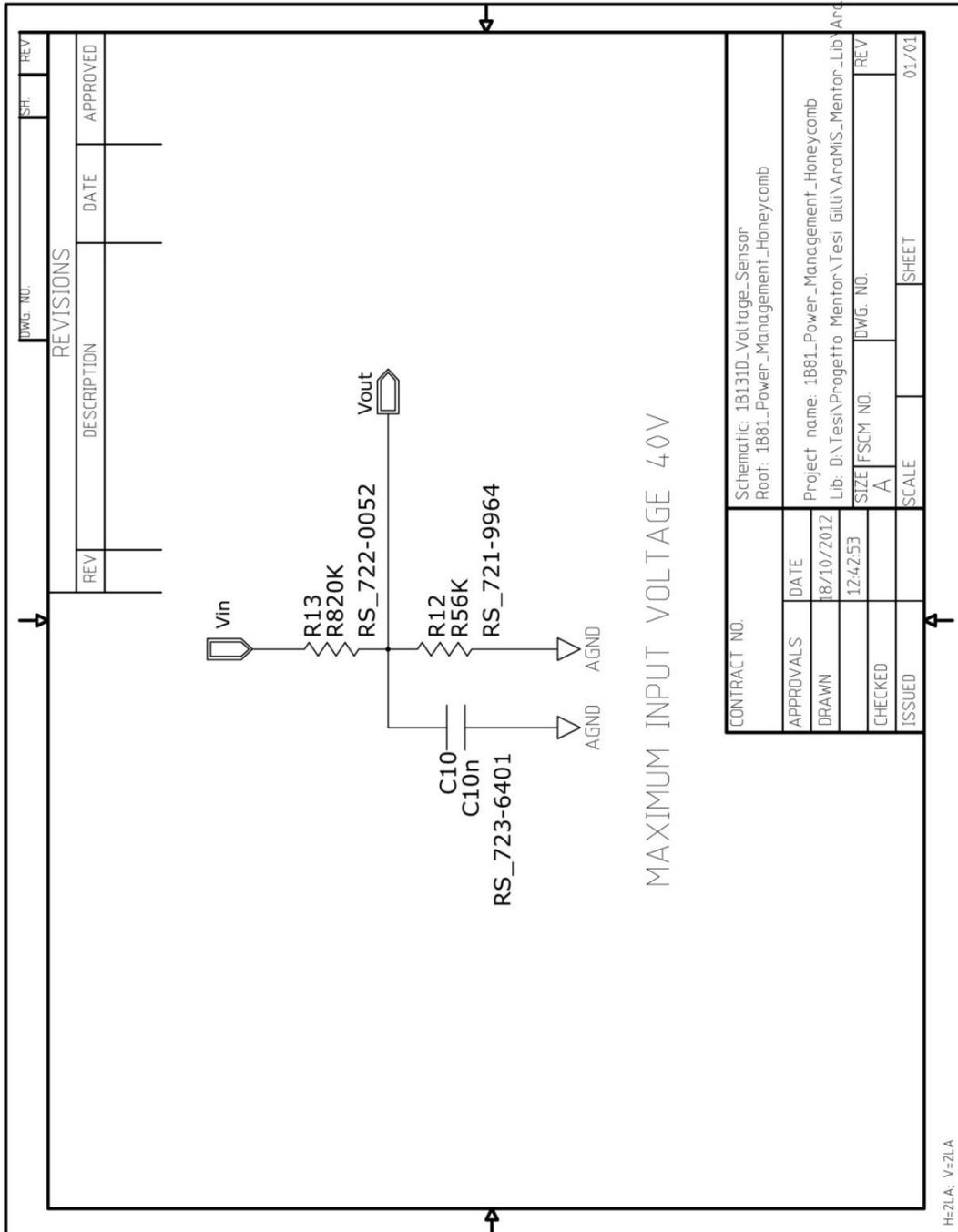


REVISIONS		DATE	APPROVED
REV	DESCRIPTION		

PROJECT:	1B81_Power_Management-Honeycomb		
	D:\Tesi\Progetto_Mentor\Tesi_Gilli\1B81_Power_Management-Honeycomb\		
	Central Library:		
	D:\Tesi\Progetto_Mentor\Tesi_Gilli\AraMIS_Mentor_Lib\AraMIS_Mentor_Lib.Lmc		
DESIGNED BY:			
SIZE:	FSCM NO.	DWG. NO.	REV
A			
	SCALE	SHEET	01/01

H=2LA; V=2LA

12.1B131D_voltage_sensor.

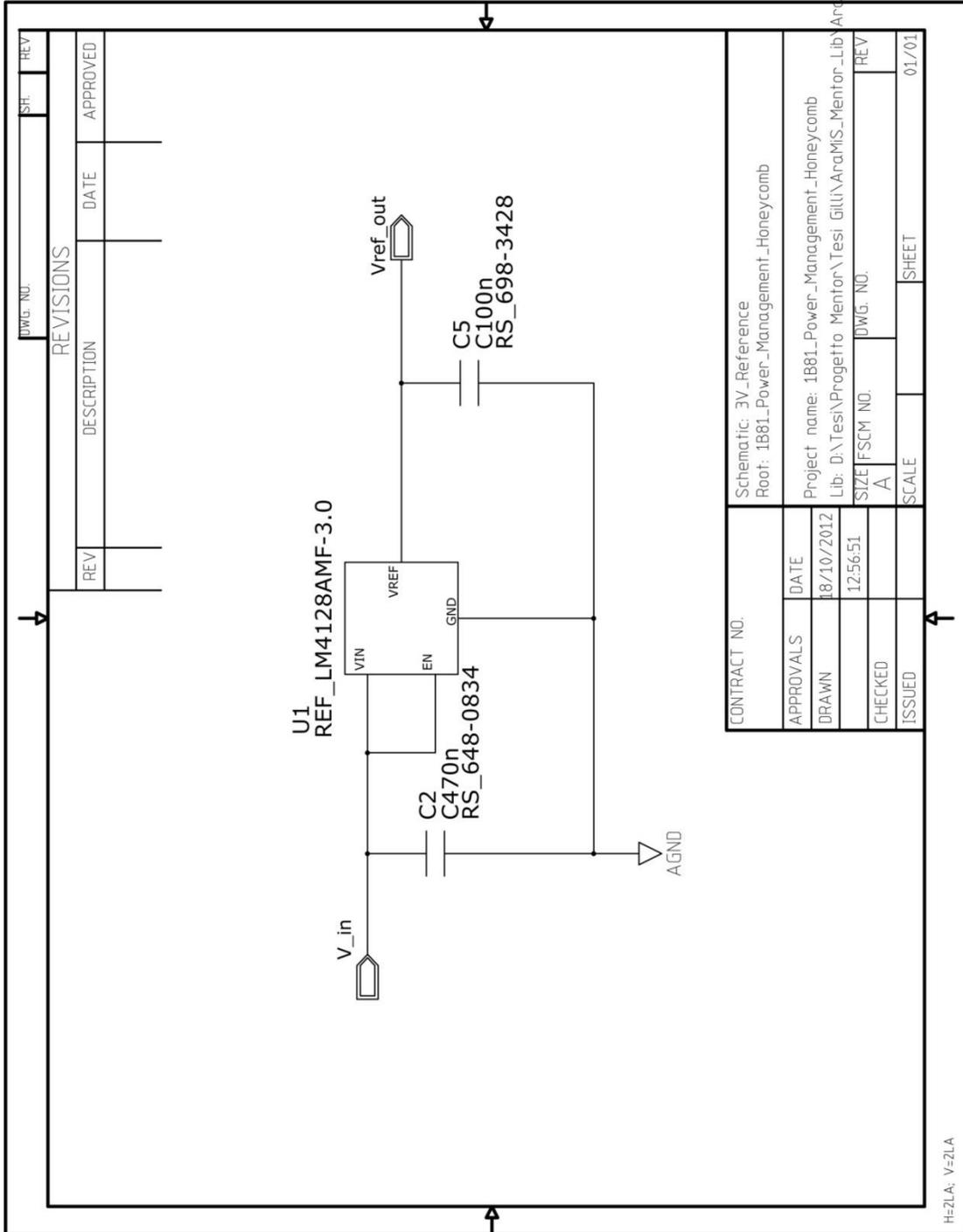


DWG. NO.		REV	SH.	REV	
REVISIONS		REV	DESCRIPTION	DATE	APPROVED

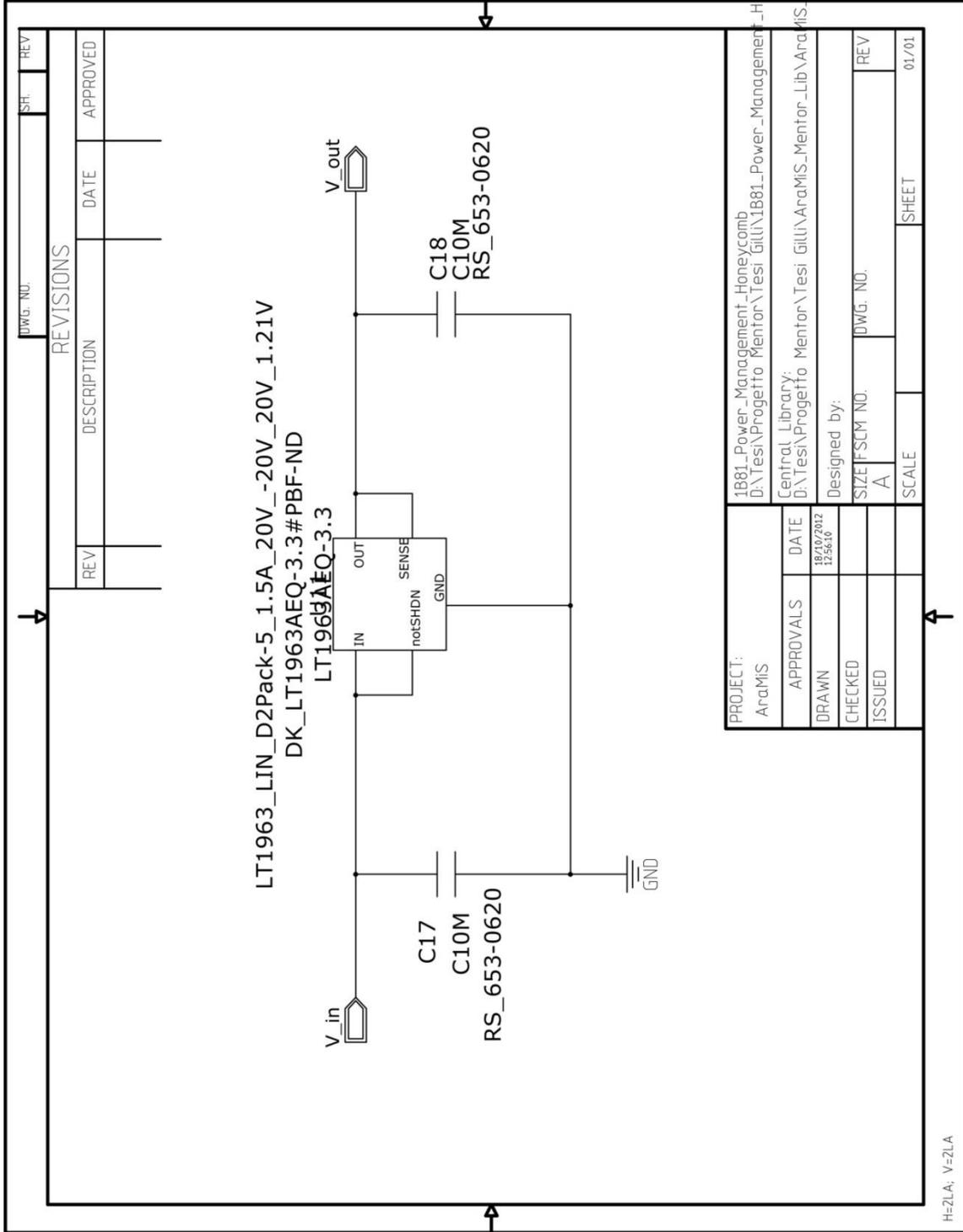
CONTRACT NO.		Schematic: 1B131D_Voltage_Sensor Root: 1B81_Power_Management_Honeycomb		
APPROVALS	DATE	Project name: 1B81_Power_Management_Honeycomb		
DRAWN	18/10/2012	Lib: D:\Tesi\Progetto_Mentor\Tesi_Gilva\AraMIS_Mentor_Lib		
CHECKED	12/4/53	SIZE	FSCM NO.	REV
ISSUED		A		
		DWG. NO.	SCALE	SHEET
				01/01

H=2LA, V=2LA

14.3V_Reference.



15.3V3_Power.



REV		DATE		APPROVED	

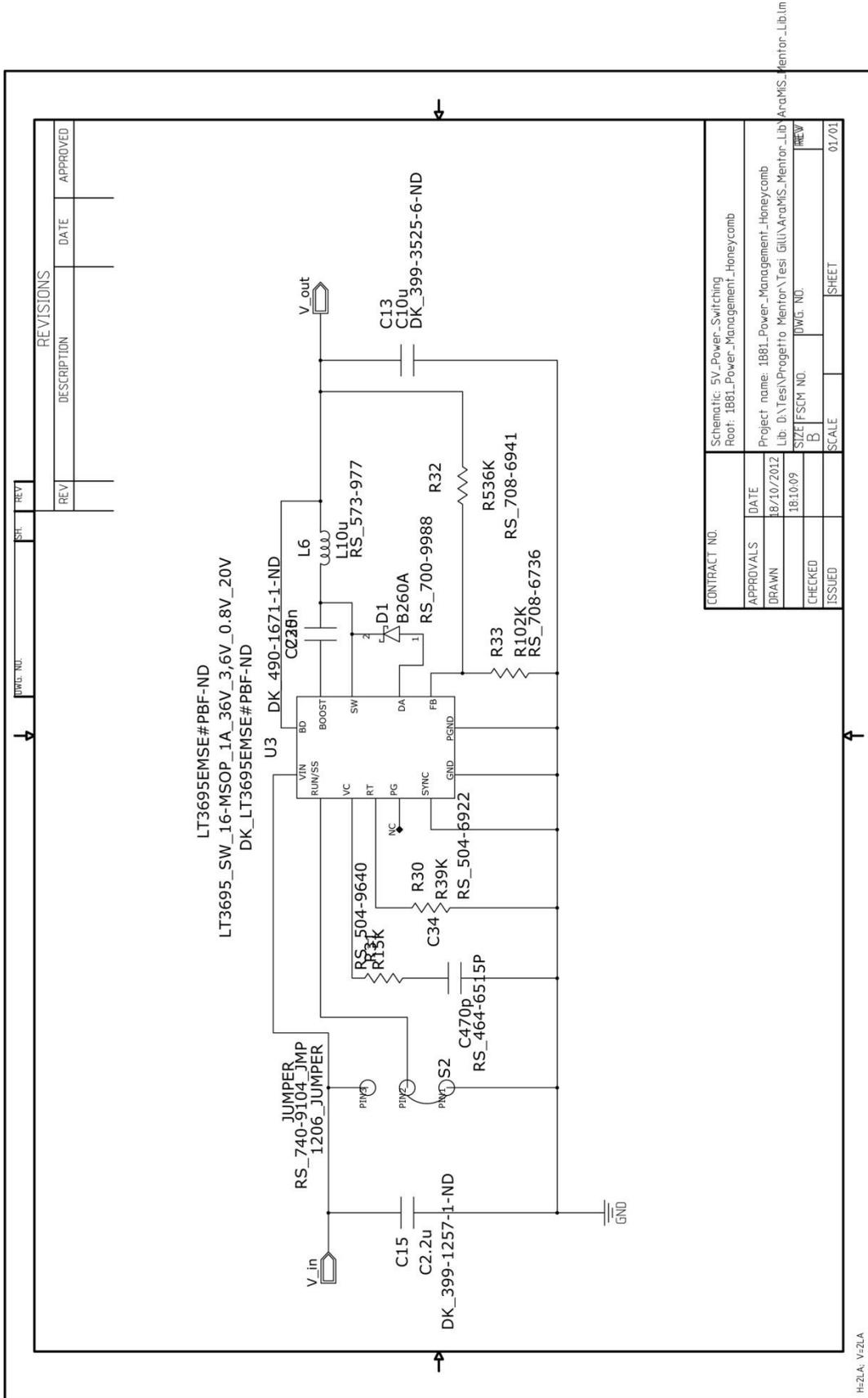
DESCRIPTION		DATE		APPROVED	

DWG. NO.	SH.	REV.
----------	-----	------

PROJECT:	1B81_Power_Management_Honeycomb				
AraMIS	D:\Tesi\Progetto Mentor\Tesi Gil\1B81_Power_Management_Honeycomb\				
APPROVALS	DATE	Central Library:			
DRAWN	19/07/2012	D:\Tesi\Progetto Mentor\Tesi Gil\AraMIS_Mentor_Lib\AraMIS_Mentor_Lib.lmc			
CHECKED	12.2010	Designed by:			
ISSUED		SIZE	FSCM NO.	DWG. NO.	REV.
		A			
		SCALE		SHEET	01/01

HE-2LA; V=2LA

16.5V_Power_Switching.

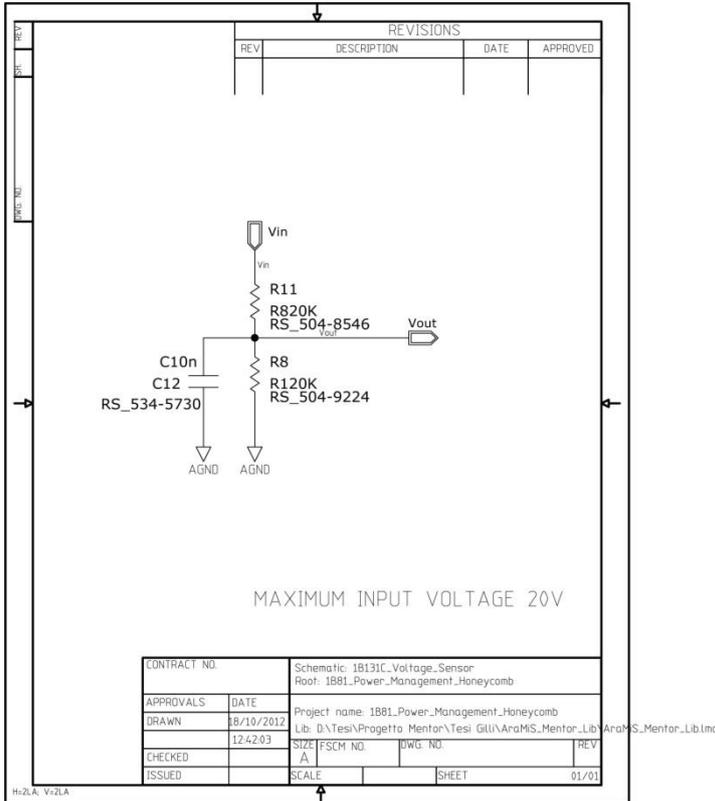


REVISIONS			
REV	DESCRIPTION	DATE	APPROVED

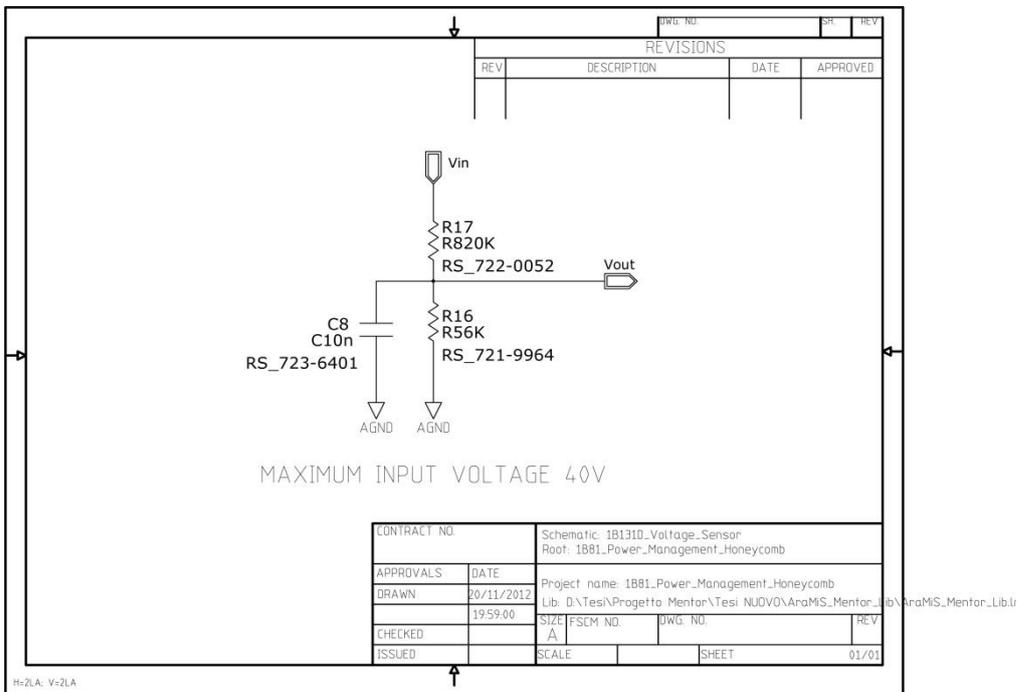
CONTRACT NO.		Schematic: 5V_Power_Switching	
APPROVALS		Root: 1B81_Power_Management_Honeycomb	
DRAWN	DATE	Project name: 1B81_Power_Management_Honeycomb	
	18/10/2012	Lib: D:\Tesi\Progetto_Mentor\Tesi_Giliu\AraMS_Mentor_Lib_AraMS_Mentor_Lib.lib	
CHECKED	18/10/09	SIZE	FSCM NO.
		B	
ISSUED		SCALE	SHEET
			01/01

HzLA V=ZLA

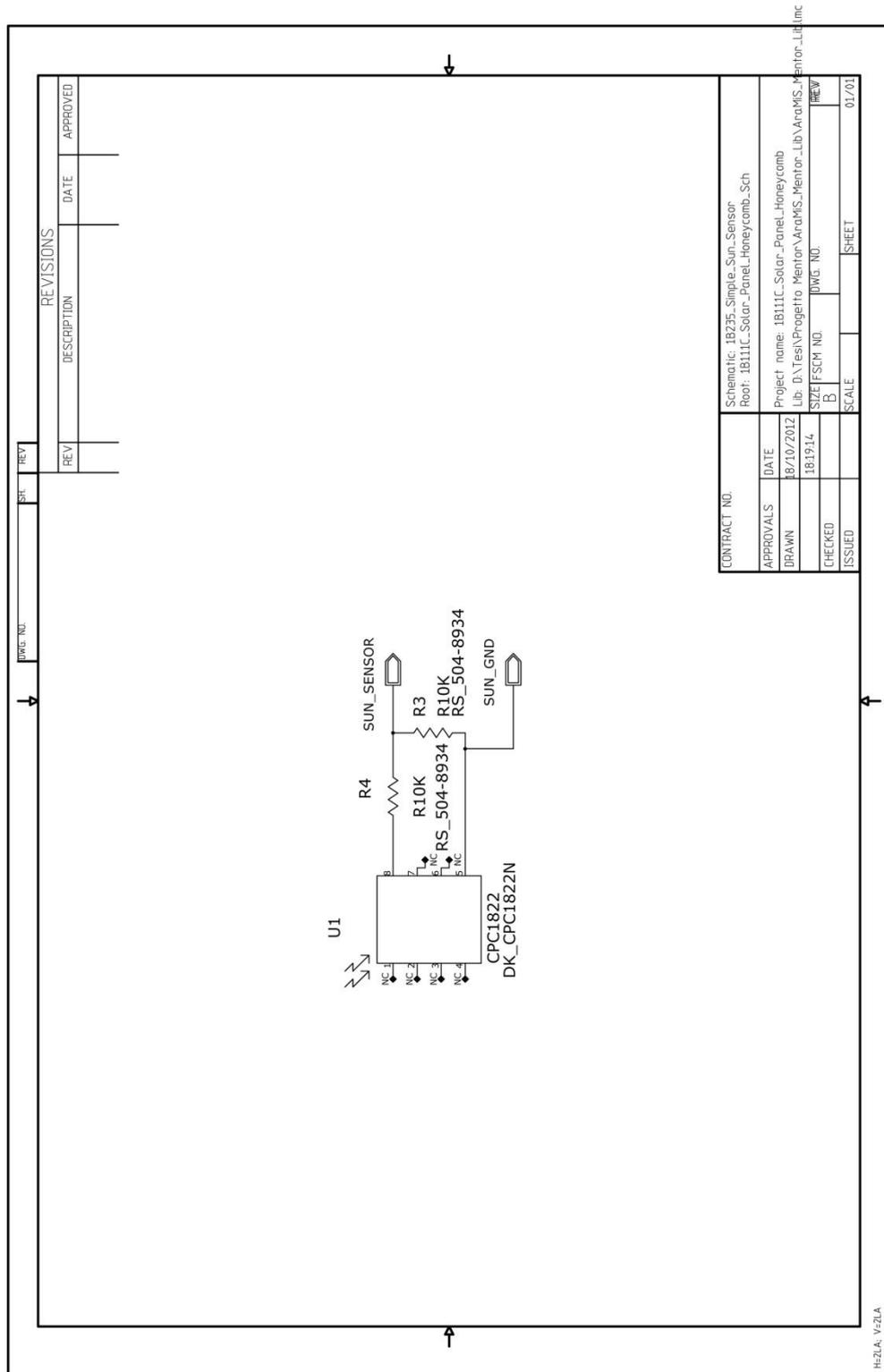
17.1B131C_Voltage_Sensor.



18.1B131D_Voltage_Sensor.



20.Bk1B235_Simple_Sun_sensor



H=ZLA_V=ZLA

REV	DESCRIPTION	DATE	APPROVED

CONTRACT NO.	Schematic: IB235_Simple_Sun_Sensor Floor: IB111C_Solar_Panel_Honeycomb_Sch		
APPROVALS	DATE		
DRAWN	18/10/2012	Project name: IB111C_Solar_Panel_Honeycomb Lib: D:\Tesi\Progetto_Mentor\ArchiMS_Mentor_Lib\ArchiMS_Mentor_Lib.mc	
CHECKED	18/10/14	SIZE	FSCM NO.
ISSUED		B	
		SCALE	SHEET
			01/01

Bibliografia

- [1] L.M. Reyneri, S. Speretta, J. Rotteveel, E. Petri, S. Woldendrop *Integrated harness technology (Proposal in response to ESA invitation to tender A06445 TEC-MMA/2010/40)*, settembre 2010.
- [2] M. Borri, L.M. Reyneri *Mechanical Subsystem of ARAMIS – V1.0.11*, maggio 2011.
- [3] Martin Flower *UML Distilled*, third edition, Pearson Education, 2004.
- [4] Visual paradigm: sito ufficiale. <http://www.visual-paradigm.com/>
- [5] Luis Castaner, Santiago Silvestre *Modelling Photovoltaic System using Pspice*, Wiley.
- [6] H. Abdullah, Z. Demir, N. Sazak, M. Yildiz *A study of silicon solar cells and modules using Pspice reflection on power supply design*, IEEEExplore.
- [7] S. Speretta *1B111_Solar_Panel_Fault_Tolerance_Analysis_V1*.
- [8] E. Ruo Rui, V. Gilli *1B111_Solar_Panel_Fault_Tolerance_Analysis_V2*, febbraio 2012.
- [9] M. Caldaralo *Manuale utente per realizzare circuiti stampati con il software Mentor Graphics*, ottobre 2004.
- [10] *Expedition PCB User's Guide*, agosto 2005.
- [11] *Design Capture for Expedition PCB training manual*, agosto 2001.
- [12] D. Schelle, J. Castorena *Buck-Converter Design Demystified*, IEEEExplore, giugno 2006.
- [13] K.S. Phani Kiranmai, M. Veerachary *Maximum Power Point Tracking: A PSPICE Circuit Simulator Approach*, IEEEExplore, 2005.
- [14] De los Rios, Rizwan, Reyneri *Board to Board Connector ICD of ARAMIS – V1.0.11*, gennaio 2012.