Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative

SUPSI

Smart meter con disaggregazione dei carichi

Studente/i

Thomas Di Blasi

Relatore

Vasco Medici

Correlatore

Davide Rivola
Giuseppe Ippolito

Committente

SUPSI

Corso di laurea

Ingegneria elettronica

Modulo

Progetto di diploma

Anno

2018

Data

3 settembre 2018

STUDENTSUPSI

Indice

1 Abstract **1**

2 Introduzione **3**

3 Premessa **5**
3.1 Specifiche di progetto 6
3.2 Approfondimento delle specifiche ed organizzazione 7

4 Approfondimenti teorici **11**
4.1 Nonintrusive load monitoring (NILM) 11
4.2 Raspberry Pi CM3 13
4.3 Energy metering chip 15
4.4 ADE9000 16
4.4.1 Analog-to-digital converter (ADC) 17
4.4.2 Sequenza di alimentazione e avvio dell'ADE9000 22
4.4.3 Correzioni e calcoli effettuati dall'ADE9000 24
4.4.4 Calibrazione 25
4.5 SPI, comunicazione tra Raspberry e ADE9000 26

5 Lavoro pratico **31**
5.1 PCB 31
5.1.1 Alimentazione 32
5.1.2 ADE9000 su PCB 36
5.1.3 Raspberry su PCB 38
5.1.3.1 Real Time Clock 39
5.1.3.2 Relay 40
5.1.3.3 Risultato finale 41
5.2 Software 45
5.2.1 Sistema operativo e attivazione SPI Raspberry 45
5.2.2 Python 47
5.2.3 Programma 48

STUDENTSUPSI

6 Consuntivo finale **51**

7 Conclusioni **53**

8 Sitografia **55**

STUDENTSUPSI

Elenco delle figure

3.1 Diagramma di Gantt 8
3.2 Quadro elettrico 9
3.3 Schema a blocchi dello smart meter 10

4.1 Processo base NILM 11
4.2 Distinzione tra potenza reattiva e reale 12
4.3 Raspberry Compute Module 3 13
4.4 Diagramma a blocchi CM3 14
4.5 Schema a blocchi ADE9000, Analog Devices 16
4.6 ADE9000 chip 16
4.7 Segnale input massimo 17
4.8 Circuito applicato con sensore di corrente 18
4.9 Circuito applicato per la misura della tensione tramite partitore resistivo 18
4.10 Datapath dopo gli ADC 19
4.11 Canali di misurazione della corrente e frequenza di campionamento relativa . 20
4.12 Modulatore ADC_REDIRECT a digital datapath multiplexing 21
4.13 Stato ADC in PSMx Power Mode 22
4.14 Sequenza di alimentazione per modalità PSM0 22
4.15 Collegamento base del protocollo SPI 26
4.16 CMD_HDR [15:0] 27
4.17 Collegamenti SPI, con più slave (sinistra), a catena (destra) 29

5.1 Layer PCB 32
5.2 Schema a blocchi PCB 33
5.3 Layer PCB 34
5.4 ADE9000 su PCB 36
5.5 Raspberry pi CM3 su PCB 38
5.6 Adafruit PCF8523 Real Time Clock 39
5.7 Relay, schema elettrico e su PCB 40
5.8 Visualizzazione 3D del PCB 41
5.9 Top layer e top overlay 42

STUDENTSUPSI

5.10 Bottom layer e bottom overlay 43
5.11 PCB finale 44
5.12 Raspberry CM3 con l'Evaluation board 45
5.13 Python logo 47
5.14 Codice di Setup 48

STUDENTSUPSI

Capitolo 1

Abstract

In questo documento, si vuole mostrare tutto il lavoro fatto, riguardo al progetto di diploma *Smart meter con disaggregazione dei carichi*.

Tra queste pagine, vengono spiegati diversi aspetti riguardo all'importanza del progetto. Si trovano gli approfondimenti teorici sulle tecnologie utilizzate.

Si vedono nel dettaglio le principali periferiche. Vengono descritti i principali passaggi per la realizzazione del PCB e le idee che si sono avute.

Si parla delle problematiche incontrate e di come si sono affrontate. In generale si mostra la crescita, fino alla nascita, di un prototipo, osservandone le parti più interessanti.

Il progetto viene valutato sotto ogni aspetto, tecnico e finanziario, confrontandone il risultato con le aspettative iniziali.

STUDENTSUPSI

STUDENTSUPSI

Capitolo 2

Introduzione

Il progetto si prefigge come scopo quello di sviluppare uno smart meter capace di monitorare la rete trifase di un'abitazione, riuscendo a detettare le apparecchiature che vengono azionate senza l'uso di sensori applicati sugli stessi apparecchi, monitorandone i consumi legati al tempo d'uso. Questa tecnica di misura è chiamata monitoraggio non intrusivo dei carichi (NILM). Questa tecnica è considerata un'alternativa a basso costo invece che il collegamento di singoli apparecchi di misura su ciascun carico. Tuttavia, presenta problemi di privacy, in quanto i dati di monitoraggio spesso vengono inviati ad un server, il quale si occupa di effettuare la disaggregazione[1]. In questo progetto si è voluto progettare uno smart meter dalle capacità computazionale sufficienti ad effettuare la disaggregazione localmente.

Il monitoraggio non intrusivo dei carichi è un processo che, tramite l'analisi dei cambiamenti di tensione e corrente permette di dedurre quali elettrodomestici vengono utilizzati e di isolarne il consumo energetico individuale.

Più ampiamente uno smart meter consente di ottimizzare la produzione e la distribuzione di energia elettrica, in quanto permette di fare un analisi territoriale dei consumi di energia elettrica, consentendo alle aziende che si occupano della produzione e del trasporto di effettuare ottimizzazioni volti all'utente finale oppure di pianificare la sostituzione di elettrodomestici con altri più efficienti. Una volta installato lo smart meter le misure di un dato profilo di consumo di energia elettrica avviene in modo automatico, inoltre, grazie ad una pagina web, l'utente avrà la possibilità di prendere visione dei consumi elettrici della propria abitazione.

[1] v.tr. [sogg-v-arg-prep.arg] Separare qlco. da un insieme; in statistica, ripartire un dato globale in dati più elementari o più omogenei

STUDENTSUPSI

STUDENTSUPSI

Capitolo 3

Premessa

Il presupposto di questo progetto di diploma, è quello di mettere lo studente alla prova, in un ambiente che rispecchia quello lavorativo, dove c'è un obbiettivo da raggiungere per un cliente e consentire allo studente di mettere in pratica la teoria appresa.

Il progetto di diploma si compone di una fase iniziale, la presentazione dei progetti disponibili agli studenti e proposti dai professori o da aziende esterne, qui lo studente sceglie il progetto che più lo attira. Poi una fase di lavoro ed infine la fase di presentazione e difesa del progetto, dove lo studente mostra il suo percorso ad un pubblico composto anche di esperti.

Nella prima fase prima descritta, vi è un documento per ogni progetto. In questo documento vengono riportate diverse informazioni, tra cui gli obbiettivi che devono essere raggiunti. Questi obbiettivi sono spesso chiamati specifiche di progetto.

3.1 Specifiche di progetto

- Design elettronico su altium completato;
- Sviluppo, produzione e montaggio di un prototipo su PCB;
- Test di un set di funzionalità di base:
  - Misura corrente;
  - Misuare tensione;
  - Misura Potenze e cos(ϕ);
  - Misura frequenza;
- Calibrazione;
- Sviluppo e test codice per NILM.

I compiti assegnati allo studente sono i seguenti:

- Definizione delle funzionalità e scelta dei componenti in base alle specifiche richieste dagli algoritmi NILM (risoluzione, precisione, frequenza di campionamento, ecc.)
- Progettazione, configurazione e realizzazione della scheda di misura
- Programmazione firmware per l'acquisizione delle misure
- Analisi della performance di algoritmi NILM forniti

Si farà uso delle seguenti tecnologie:

- Rasberry compute module;
- Altium;
- Python;

Una volta scelto il progetto, è compito dello studente approfondire gli obbiettivi per avere una chiara idea di come affrontare il progetto. Come descritto nei compiti, la prima fase è stata quella di definire le funzionalità del prototipo, per cui, dopo un analisi degli obbiettivi ed una valutazione delle tecnologie, si è arrivati a definire degli obbiettivi più approfonditi. Le specifiche approfondite sono riportate di seguito

STUDENTSUPSI

3.2 Approfondimento delle specifiche ed organizzazione

Obbiettivi approfonditi del progetto:

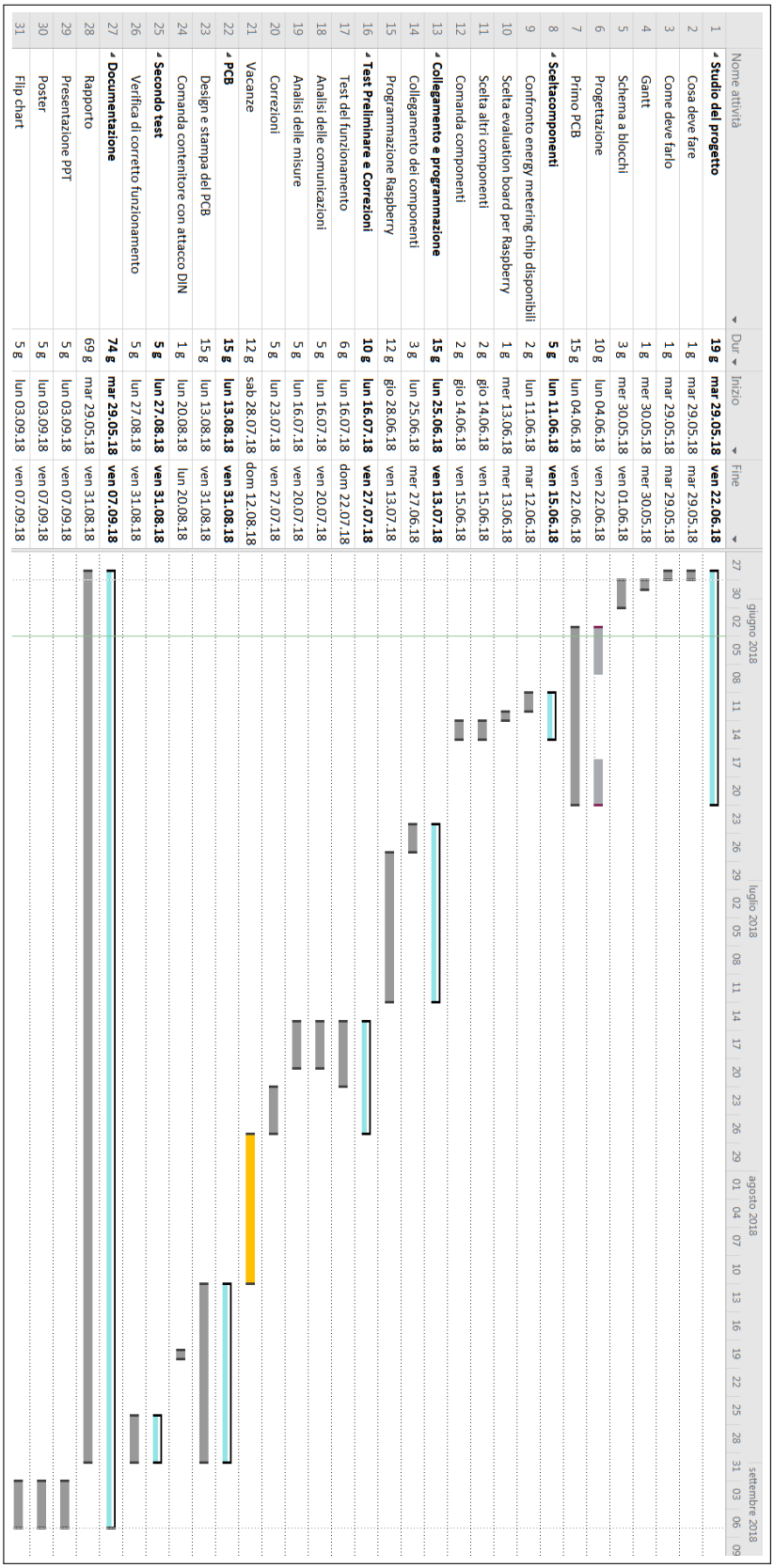
- Lo smart meter deve utilizzare Raspberry Pi Compute Module 3 per poter effettuare la disaggregazione dei carichi a bordo;
- Esso deve poter monitorare una rete trifase o una monofase, compreso il neutro;
- Deve avere la capacità di scollegare un carico;
- Si deve alimentare dalla fase 1;
- Deve mostrare i dati ricavati su una pagina Web.

Principalmente lo smart meter sarà composto da:

- Un chip di misura;
- Raspberry Pi CM3;
- Elettronica di alimentazione;
- Relay per gestire un contattore esterno;
- Presa USB.

Nell'immagine 3.1 viene mostrato il diagramma di Gantt, uno strumento di supporto alla gestione dei progetti. In questo diagramma viene riportato il progetto suddiviso in compiti e disposti secondo il tempo necessario e a disposizione. Si prevedeva inizialmente di creare due PCB, il secondo sarebbe stato un ottimizzazione del primo.

STUDENTSUPSI



	Nome attività	Dur	Inizio	Fine
1	**Studio del progetto**	**19 g**	**mar 29.05.18**	**ven 22.06.18**
2	Cosa deve fare	1 g	mar 29.05.18	mar 29.05.18
3	Come deve farlo	1 g	mar 29.05.18	mar 29.05.18
4	Gantt	1 g	mer 30.05.18	mer 30.05.18
5	Schema a blocchi	3 g	mer 30.05.18	ven 01.06.18
6	Progettazione	10 g	lun 04.06.18	ven 22.06.18
7	Primo PCB	15 g	lun 04.06.18	ven 22.06.18
8	**Scelta componenti**	**5 g**	**lun 11.06.18**	**ven 15.06.18**
9	Confronto energy metering chip disponibili	2 g	lun 11.06.18	mar 12.06.18
10	Scelta evaluation board per Raspberry	1 g	mer 13.06.18	mer 13.06.18
11	Scelta altri componenti	2 g	gio 14.06.18	ven 15.06.18
12	Comanda componenti	2 g	gio 14.06.18	ven 15.06.18
13	**Collegamento e programmazione**	**15 g**	**lun 25.06.18**	**ven 13.07.18**
14	Collegamento dei componenti	3 g	lun 25.06.18	mer 27.06.18
15	Programmazione Raspberry	12 g	gio 28.06.18	ven 13.07.18
16	**Test Preliminare e Correzioni**	**10 g**	**lun 16.07.18**	**ven 27.07.18**
17	Test del funzionamento	6 g	lun 16.07.18	dom 22.07.18
18	Analisi delle comunicazioni	5 g	lun 16.07.18	ven 20.07.18
19	Analisi delle misure	5 g	lun 16.07.18	ven 20.07.18
20	Correzioni	5 g	lun 23.07.18	ven 27.07.18
21	Vacanze	12 g	sab 28.07.18	dom 12.08.18
22	**PCB**	**15 g**	**lun 13.08.18**	**ven 31.08.18**
23	Design e stampa del PCB	15 g	lun 13.08.18	ven 31.08.18
24	Comanda contenitore con attacco DIN	1 g	lun 20.08.18	lun 20.08.18
25	**Secondo test**	**5 g**	**lun 27.08.18**	**ven 31.08.18**
26	Verifica di corretto funzionamento	5 g	lun 27.08.18	ven 31.08.18
27	**Documentazione**	**74 g**	**mar 29.05.18**	**ven 07.09.18**
28	Rapporto	69 g	mar 29.05.18	ven 31.08.18
29	Presentazione PPT	5 g	lun 03.09.18	ven 07.09.18
30	Poster	5 g	lun 03.09.18	ven 07.09.18
31	Flip chart	5 g	lun 03.09.18	ven 07.09.18

Figura 3.1: Diagramma di Gantt

STUDENTSUPSI

Lo smart meter viene collocato solitamente all'interno del quadro elettrico all'ingreso di ogni abitazione. Nell'immagine 3.2 viene mostrato in modo chiaro come si collocherà lo smart meter quando si alimentano due abitazioni allacciate allo stesso punto della rete.

Si può vedere inoltre, nell'immagine 3.3, le periferiche principale che sono connesse tra di loro e alla rete da misurare. Si può notare che vi è la presenza di un teleruttore, questo da la possibilità allo smart meter di scollegare uno dei vari carichi all'interno dell'abitazione (ad esempio il boiler dell'acqua calda) qualora si volesse limitare il consumo eccessivo di energia elettrica.

Come altri particolari che possono essere aggiunti qual'ora gli obbiettivi principale fossero raggiunti, si è pensato di includere un M-Bus per la misurazione di Gas e Acqua e la possibilità di gestire lo smart meter tramite la pagine web che esso crea, per poter specificare la rete (monofase o trifase) da misurare, o altri parametri.

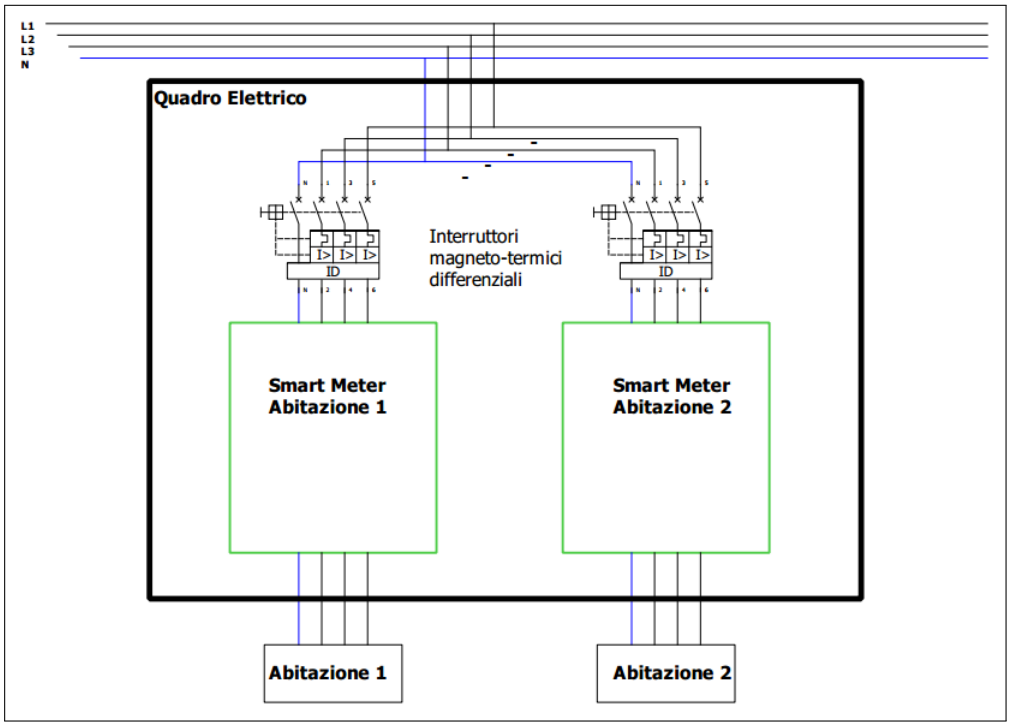


Figura 3.2: Quadro elettrico

STUDENTSUPSI

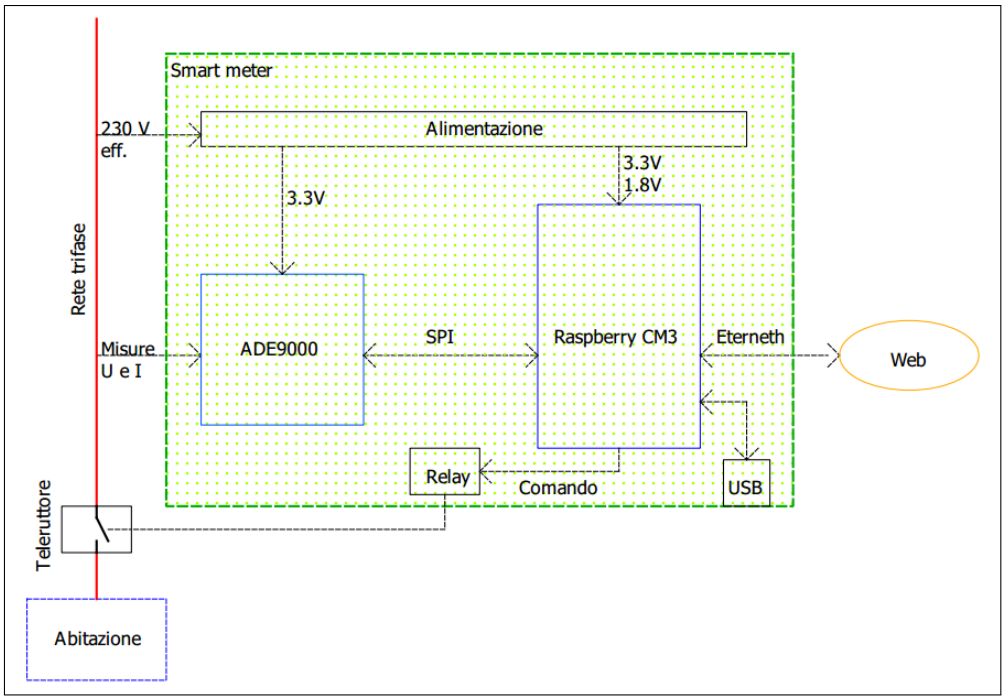


Figura 3.3: Schema a blocchi dello smart meter

STUDENTSUPSI

Capitolo 4

Approfondimenti teorici

4.1 Nonintrusive load monitoring (NILM)

Questa tecnologia fu inventata da George W. Hart, Ed Kern e Fred Schweppe, tre ricercatori presso la Massachusetts Institute of Technology nel 1980, dove venne fondata la Eletric Power Research Institute. Vengono misurati i cambiamenti di tensione e corrente delle fasi prese in analisi tramite un misuratore AC che converte i valori analogici in digitale.

Questi vengono normalizzati, ovvero portati a dei valori compatibili per essere registrati e successivamente questo insieme di dati raccolti vengono analizzati, per collegare l'accensione allo spegnimento dello stesso apparecchio e calcolare l'energia consumata dal medesimo.

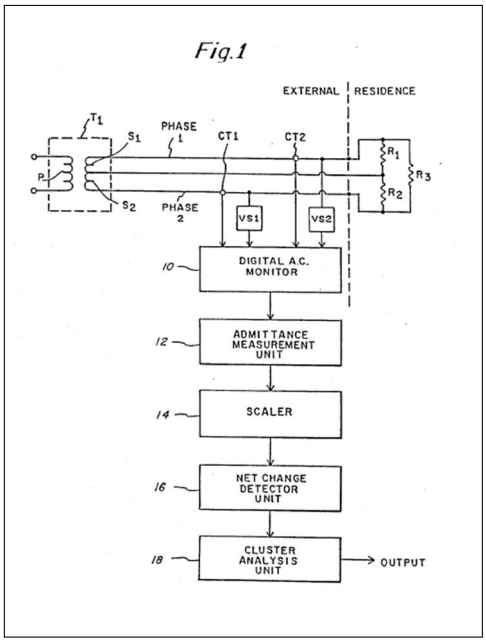


Figura 4.1: Processo base NILM

STUDENTSUPSI

Se per esempio venissero accesi in sequenza una lampadina da 60 W e una da 100 W, e successivamente questi venissero spenti in sequenza, l'algoritmo metterebbe in relazione i segnali di accensione e spegnimento della lampadina da 60 W e quella da 100 W e quindi determinare quanta potenza è stata consumata dalle due lampadine e quando, e quindi determina l'energia consumata.

Inoltre il sistema è capace di misurare la potenza reattiva e quella reale, così da poter distinguere ulteriormente gli apparecchi che vengono azionati. Nella figura 4.2 viene riportato come esempio la potenza consumata da un riscaldatore elettrico (Heater) ed un frigorifero (Refrigerator).

Si può notare che la potenza consumata dal frigorifero si compone di 200 VAR (potenza reattiva) e 200 W (potenza attiva), invece il riscaldatore consuma solo potenza attiva. Grazie a questa differenza costruttiva degli apparecchi elettrici utilizzati in un'economia domestica, l'algoritmo NILM ricava un profilo di consumo giornaliero di ogni apparecchio in modo distinto. Così facendo si ha un maggiore controllo sui propri consumi in quanto si può prendere in esame i consumi singolarmente e valutare la sostituzione degli apparecchi che consumano maggiormente.

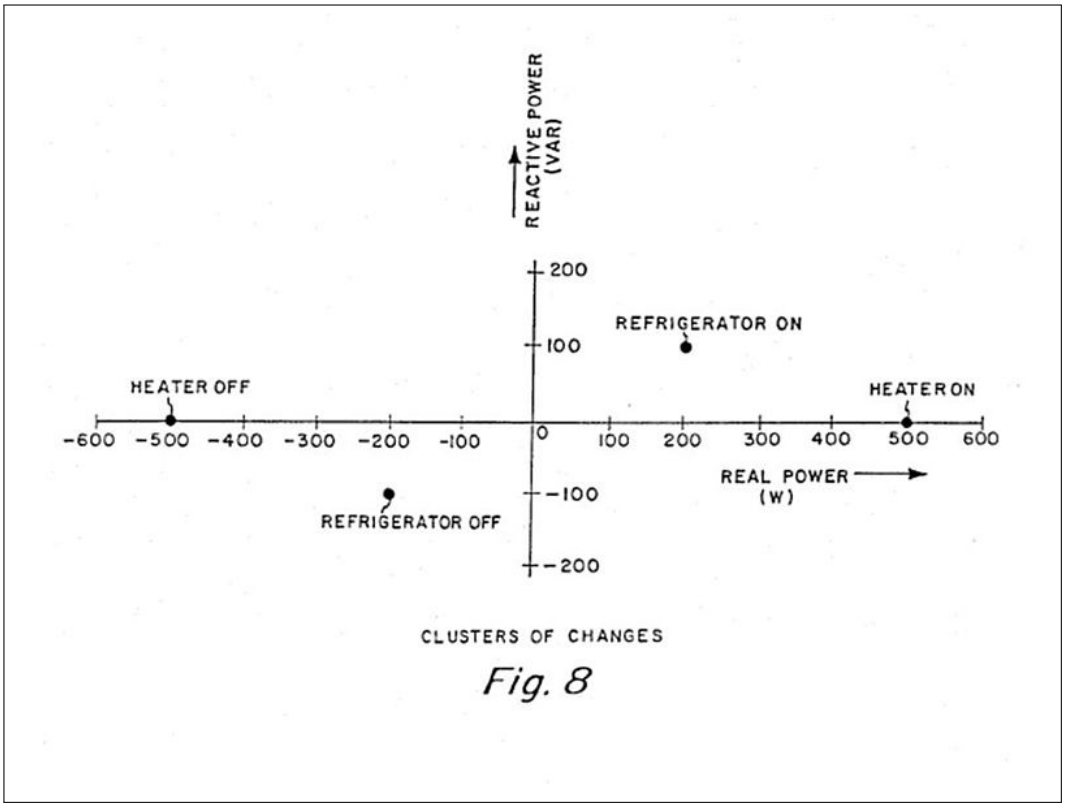


Figura 4.2: Distinzione tra potenza reattiva e reale

STUDENTSUPSI

4.2 Raspberry Pi CM3

Il Raspberry Pi Compute Module 3 è un sistema su modulo (System on Modules, SoMs) che contiente un processore, una memoria e una Flash eMMC, oltre che una circuiteria di supporto per l'alimentazione. Questo modulo venne sviluppato con lo scopo di poter essere implementato in un sistema custom, quindi offre le sue potenzialità di sistema pronto da montare. Il CM3 contiene il processore Quad core Cortex A53 di nome BCM2837, 1 Gbyte di RAM LPDDR2 e 4 Gbyte di memoria Flash eMMC, inoltre supporta l'aggiunta di due memorie SD. Il CM3 possiede molte periferiche che possono essere viste sul suo datasheet, in questo progetto si fa uso di alcune delle sue 48 GPIO[1], della porta USB e di una delle due porte SPI. Le GPIO hanno un minimo di due fino a un massimo di sei funzionalità alternative,

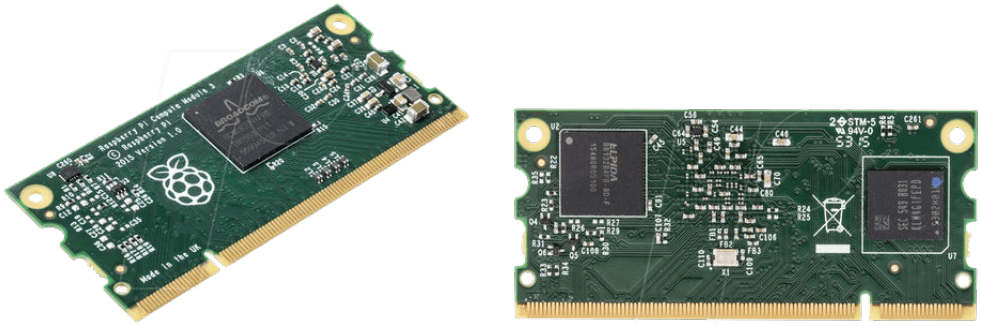


Figura 4.3: Raspberry Compute Module 3

oppure possono essere usati come input (opzionalmente come interrupt) o come output. Sono suddivisi in banchi, con la possibilità quindi di non alimentarne uno se non venissero utilizzati le GPIO di quel banco. Raspberry supporta l'ultima versione del Kernel di Linux, e come ambiente operativo il software principale è Raspbian, un sistema operativo basato sui rilasci ufficiali di Debian per l'architettura ARM adattato per l'utilizzo su Raspberry Pi, scaricabile dal sito ufficiale della Raspberry. Esso permette poi l'installazione di software di terze parti quali Ubuntu mate e altri, sempre reperibili dal sito ufficiale. Si tratta quindi di un computer molto versatile, che si presta a tutti gli utilizzi, per questo motivo è stato scelto, perchè dà la possibilità di adempiere alle esigenze e poter quindi gestire i dati del chip di misura e connettersi ad internet, per poter visualizzare i risultati delle misure.

[1] General Purpose Input/Output

STUDENTSUPSI

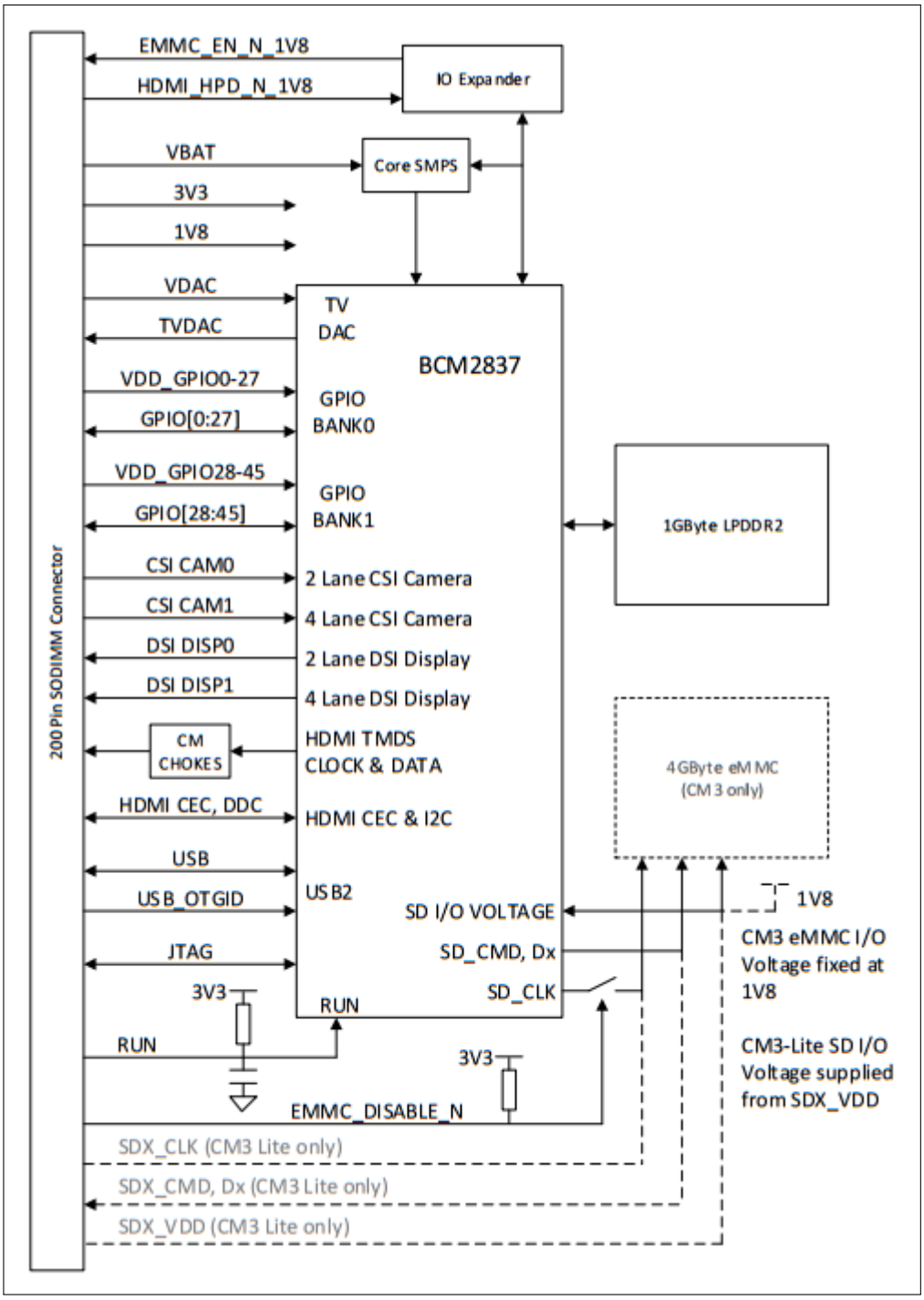


Figura 4.4: Diagramma a blocchi CM3

STUDENTSUPSI

4.3 Energy metering chip

Il chip di misura deve rispondere a determinati criteri, cioè deve misurare o ricavare corrente, tensione, potenza (attiva e reattiva), frequenza e cos(φ) di ogni fase incluso il neutro. inoltre deve poter comunicare con il Raspberry, possibilmente tramite protocollo SPI, I2C o UART, quindi compatibile con i protocolli di comunicazione presenti su Raspberry Pi CM3.

Per la ricerca di un chip di misura adatto alle esigenze del progetto si ha dedicato una fase iniziale di ricerca su internet. Tra i chip trovati in rete sono stati presi in considerazione tre di questi:

- Analog Devices ADE9000
- Microchip ATM90E32AS
- Texas Instruments MSP430F6775

Il chip di misura della Microchip possiede tutte le caratteristiche eccetto la misura della corrente di neutro. Questo chip al posto che misurare il neutro lo ricava sommando le correnti misurate delle tre fasi.

Il chip della Texas Instruments è un chip molto performante, possiede la capacità di gestire vari tipi di segnali, cioè non è dedicato principalmente alla misura di alimentazioni di bassa tensione, non è un chip ottimizzato per lo scopo di questo progetto.

Tra questi, andando per esclusione, è stato scelto il chip della Analog Devices. Esso è un chip orientato sulla misurazione e monitoraggio di reti elettrica in bassa tensione. In osservazione a queste poche ma importanti caratteristiche si è giunto alla conclusione che questo chip risponde completamente alle richieste del progetto.

Dopo una ricerca su internet, è emerso che le evaluation board per l'ADE9000 (ve ne sono due presenti sul mercato), quella per Arduino non era disponibile in breve tempo, si stimava un periodo di attesa di 3 mesi. Invece l'altra evaluation board, che permette di collegare qualsiasi tipo di processore, ha un costo di 458.- CHF. In base a queste informazioni, si è deciso di assumersi dei rischi e di progettare direttamente il PCB. Solo dopo aver ricevuto il PCB si sarebbe potuto iniziare con la fase di programmazioni. Questo è uno svantaggio che ha reso il progetto più rischioso.

STUDENTSUPSI

4.4 ADE9000

Il chip della Analog Devices ADE9000 possiede tre entrate per la misura della tensione e quattro entrate per la misura delle correnti avendo in totale sette convertitori da analogico a digitale (ADC). Possiede un canale di comunicazione SPI ed un sensore di temperatura.

Nell'immagine successiva si vuole mostrare lo schema a blocchi dell'ADE9000. Successivamente si andranno a vedere alcuni dettagli che riguardano da vicino il progetto.

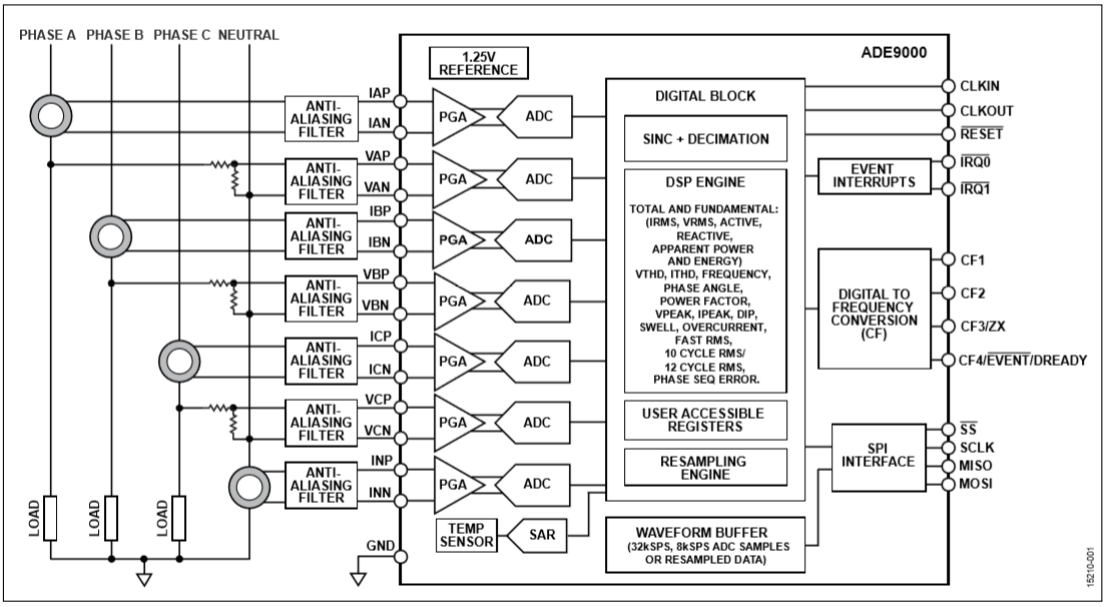


Figura 4.5: Schema a blocchi ADE9000, Analog Devices

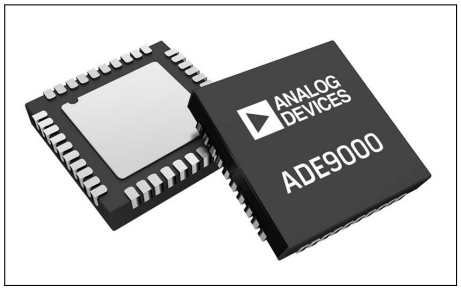


Figura 4.6: ADE9000 chip

STUDENTSUPSI

4.4.1 Analog-to-digital converter (ADC)

Un ADC è un circuito elettronico in grado di convertire un segnale analogico con andamento continuo (ad es. una tensione) in una serie di valori discreti. L'ADE9000 incorpora sette ADC $\Sigma - \Delta$ indipendenti di secondo ordine. Questo tipo di ADC, rispetto a quelli normali, è caratterizzato da un'alta risoluzione, infatti ogni ADC è di 24 bit, corrispondente a 2^{24} = 16'777'216 livelli di quantizzazione della misura e ogni ADC ha collegato al suo ingresso ad un amplificatore programmabile (PGA) che da la possibilità di applicare un guadagno di 1, 2 o 4 al valore misurato. La massima tensione che ogni ADC ammette è di 0.6V come mostrato in figura, per cui all'ingresso di ogni ADC per la misura di tensione viene applicato un partitore resistivo (vedi figura 4.9), che scala la tensione al valore consentito, mentre per gli ADC adibiti alla lettura della corrente viene usato, in questo progetto, un trasformatore di corrente chiamata anche bobina di Rogowski (vedi figura 4.8).

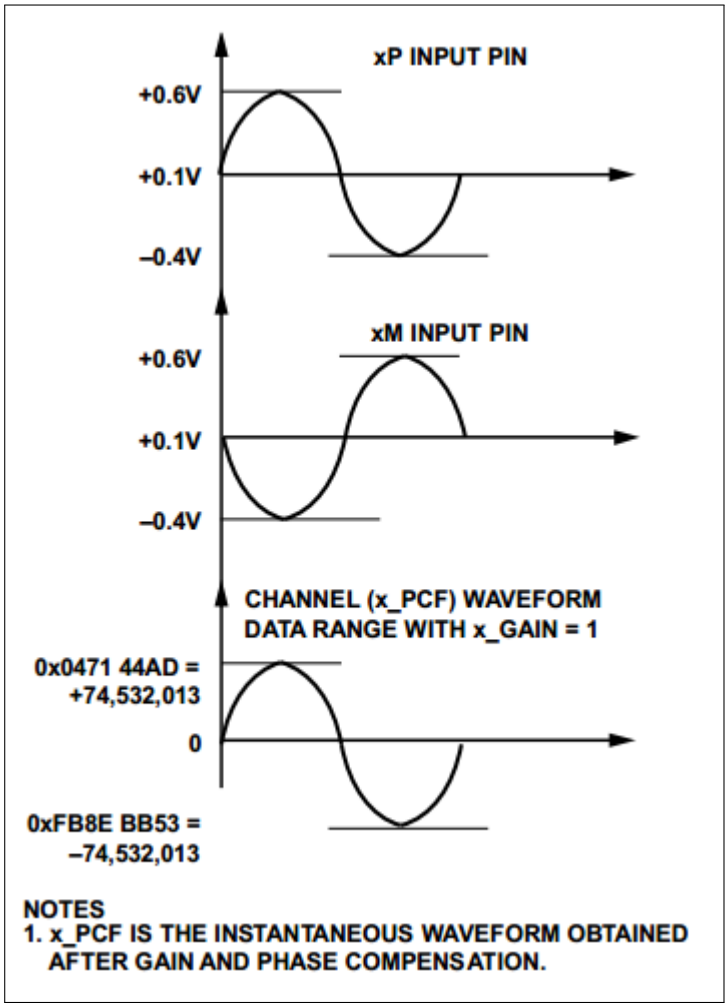


Figura 4.7: Segnale input massimo

L'interfacciamento con i sensori di corrente e tensione che supporto l'ADE9000 necessitano

STUDENTSUPSI

di un filtro antialiasing[2], dove nello specifico viene scelto un filtro che attenua fuori dalla banda dei 7kHz. si tratta di un filtro RC usato sia per il sensore di corrente che per l'ingresso della tensione, per evitare errori di fase tra tensione e corrente. In queste due immagini

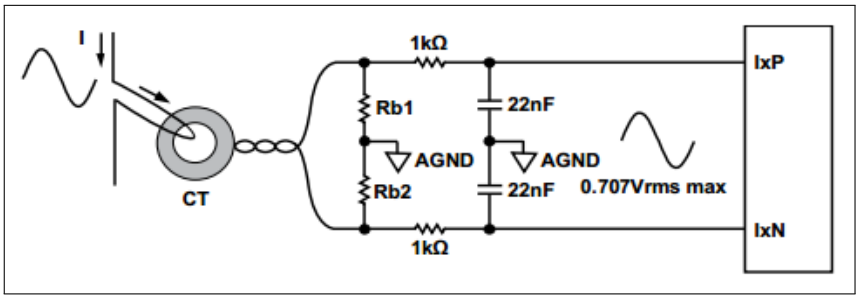


Figura 4.8: Circuito applicato con sensore di corrente

vengono riportati le interfacce tra il punto di misura e l'ADC del chip di misura. Nella prima immagine, cioè il sensore di corrente, si può vedere il filtro antialiasing RC usato, con i relativi valori per ottenere un filtraggio passa banda a 7.23kHz. Nella seconda immagine si può vedere il partitore resistivo con filtro RC.

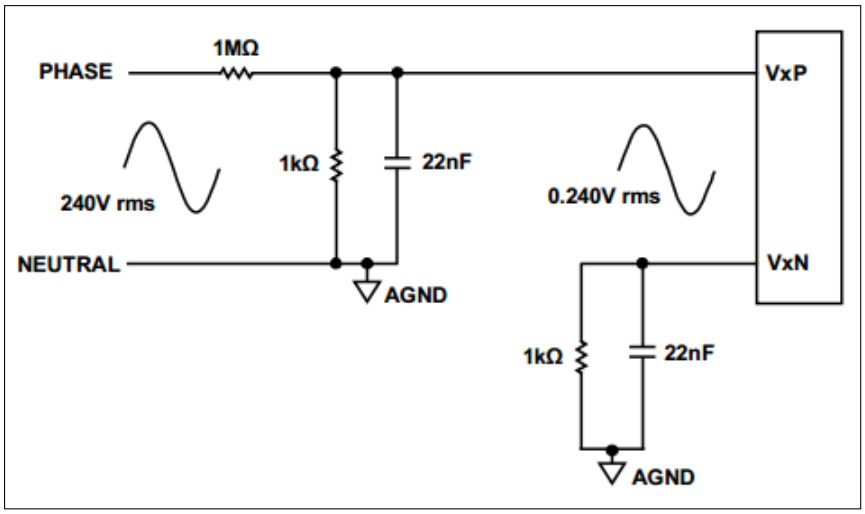


Figura 4.9: Circuito applicato per la misura della tensione tramite partitore resistivo

[2]fenomeno per il quale due segnali analogici diversi possono diventare indistinguibili una volta campionati

STUDENTSUPSI

In figura 4.10 viene mostrato il datapath che segue ogni ADC, questo è un unità di calcolo col quale vengono ricavate le misure per costruire una forma d'onda salvata in un buffer interno al chip e tutte le altre misure che possono essere ad esempio il valore istantaneo o efficace della tensione misurata. In questo processo vi è il modulatore ADC che campiona continuamente, ad una freqeunza di 2.048 MHz, il punto di misura fornendo in uscita uno strea di 1 bit dove il loro valore binario corrisponde al valore della misura. Il filtro digitale SINC4 processa il bit stream proveniente dall'ADC ottenendo la risoluzione di 24 bit.

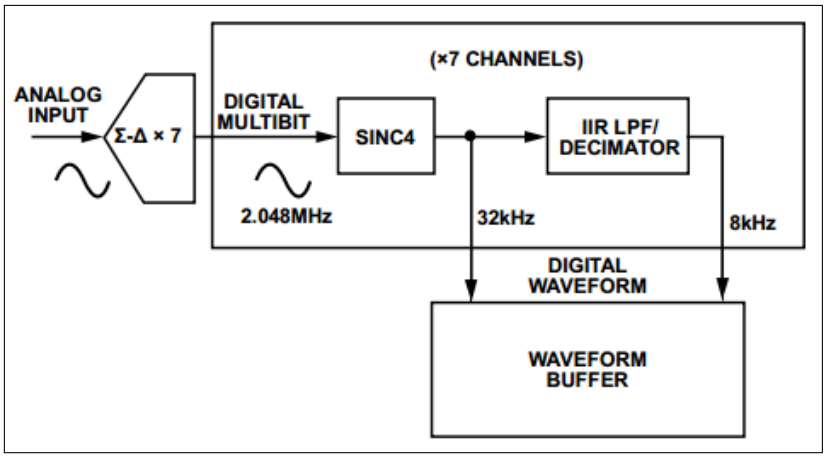


Figura 4.10: Datapath dopo gli ADC

Il IIR LPF / DECIMATOR è un filtro passa basso digitale, il suo comportamento è uguale ad un filtro passa basso analogico RC. Questo filtro ha la funzione di antialiasing. Quindi entrambi i filtri descritti sono collegati ad un buffer[3] che detiene la forma d'onda costruita grazie ai dati raccolti dai due filtri. Da questo buffer vengono ricavate le misura che il chip offre all'utente.

[3]Dispositivo di memoria usato per la conservazione provvisoria di dati destinati ad essere rielaborati, cancellati o trasferiti in un'altra unità di memoria, allo scopo di compensare le differenti velocità di trattamento delle due unità

STUDENTSUPSI

Nell'immagine che segue si possono vedere diversi registri che detengono i risultati del processo prima descritto. Si può notare che si hanno a disposizione vari tipi di misura della corrente delle tre fasi più neutro. Accedendo ai vari registri indicati si può leggere il valore desiderato.

Register Name	Description	Update Rate
AI_SINC_DAT	IA sinc4 filter output	32 ksps
BI_SINC_DAT	IB sinc4 filter output	32 ksps
CI_SINC_DAT	IC sinc4 filter output	32 ksps
NI_SINC_DAT	IN sinc4 filter output	32 ksps
AI_LPF_DAT	IA sinc4 + IIR LPF filter output	f_{DSP} = 8 ksps
BI_LPF_DAT	IB sinc4 + IIR LPF filter output	f_{DSP} = 8 ksps
CI_LPF_DAT	IC sinc4 + IIR LPF filter output	f_{DSP} = 8 ksps
NI_LPF_DAT	IN sinc4 + IIR LPF filter output	f_{DSP} = 8 ksps
AI_PCF	Instantaneous current on IA	f_{DSP} = 8 ksps
BI_PCF	Instantaneous current on IB	f_{DSP} = 8 ksps
CI_PCF	Instantaneous current on IC	f_{DSP} = 8 ksps
NI_PCF	Instantaneous current on IN	f_{DSP} = 8 ksps
AIRMS	Filtered-based total rms of IA	f_{DSP} = 8 ksps
BIRMS	Filtered-based total rms of IB	f_{DSP} = 8 ksps
CIRMS	Filtered-based total rms of IC	f_{DSP} = 8 ksps
NIRMS	Filtered-based total rms of IN	f_{DSP} = 8 ksps
ISUMRMS	Filtered rms of vector sum (AI_PCF + BI_PCF + CI_PCF ± NI_PCF); see the Neutral Current RMS, Vector Current Sum section	f_{DSP} = 8 ksps
IPEAK	Peak current channel sample; see the Peak Detection section	f_{DSP} = 8 ksps
ANGLx_xxx	Voltage to current or current to current phase angle; see the Angle Measurement section	CLKIN/24 = 1024 ksps

Figura 4.11: Canali di misurazione della corrente e frequenza di campionamento relativa

STUDENTSUPSI

Il buffer prima descritto, in cui vengono salvati i dati della forma d'onda, è grande 32 bit. I dati vengono salvati shiftando verso sinistra di 4 bit ad ogni misura. L'ADE9000 dispone di una tensione di riferimento interna di 1.25 V per poter effettuare misure e calibrazione, in alternativa è possibile applicare un riferimento esterno tramite i pin REFIN e REFGND e settando il bit EXT_REF a 1 nel registro CONFIG1, così facendo si disabilita il riferimento interno.

Per calibrare la fase della misura del neutro, siccome questo punto di misura non offre alcun dato sullo sfasamento, è necessario indirizzare l'uscita dell'ADC del neutro ad un datapath di un'altra fase e controllare quale angolo ha il neutro in confronto alla fase uno. Questo reindirizzamento è possibile grazie a dei multiplexer posti tra ogni ADC e datapath di ogni canale. L'impostazione predefinita di questi multiplexer è che ogni ADC è collegato al proprio datapath.

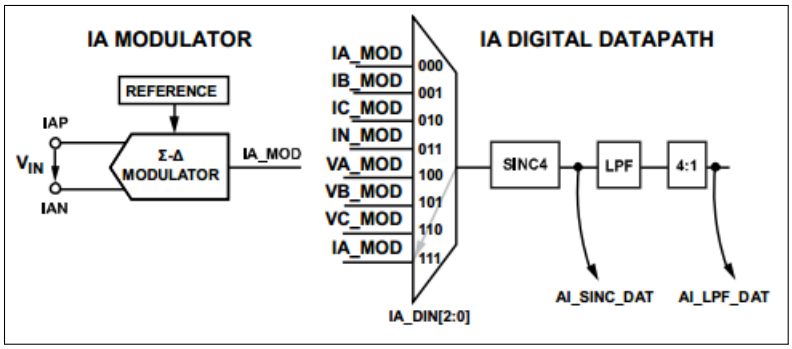


Figura 4.12: Modulatore ADC_REDIRECT a digital datapath multiplexing

STUDENTSUPSI

4.4.2 Sequenza di alimentazione e avvio dell'ADE9000

Il chip ADE9000 dispone di due ingressi, chiamati PM0 e PM1, col quale si può impostare due modalità, PSM0 e PMS3. Impostando la modalità PSM0 si attiva il chip ADE9000, quindi si attivano gli ADC, la comunicazione SPI e tutte le funzionalità. Al contrario impostando PSM3 si imposta uno stato di riposo del chip, dove tutto viene disabilitato. Osservando la tabella in figura 4.13 si può vedere che comandando i pin PM0 e PM1 si sceglie quale delle due modalità impostare. Inoltre è possibile disattivare singolarmente gli ADC scrivendo su un registro.

PSMx Power Mode	Description	PM1 Pin	PM0 Pin	Functions Available	SPI Available?
PSM0	Normal mode	0	0 or 1	All functions	Yes
PSM3	Idle	1	1	None	No

Figura 4.13: Stato ADC in PSMx Power Mode

L'alimentazione principale del chip è di 3.3 V, con limite massimo ammesso di 3.96 V, oltre questo limite si rischia di arrecare danno al chip. Se si passa in modalità PSM0 due regolatori di tensione interni al chip, AVDD e DVDD, vengono attivati per fornire un'alimentazione di 1.7 V (AVDD) e 1.9 V (DVDD). Queste tensioni sono accessibili da chip e possono essere misurati per controllare il corretto funzionamento del chip. Questi stabilizzatori di tensione inoltre servono per limitare la corrente che scorre all'interno del chip, quindi viene mantenuta una corrente inferiore ai 17 mA.

Una volta alimentanto il chip con VDD e impostata la modalità PSM0, una sequenza di alimentazione interna al chip viene seguita. Questa sequenza viene descritta nell'immagine che segue ed è importante per sincronizzare le varie periferiche del chip e ottenere un corretto funzionamento.

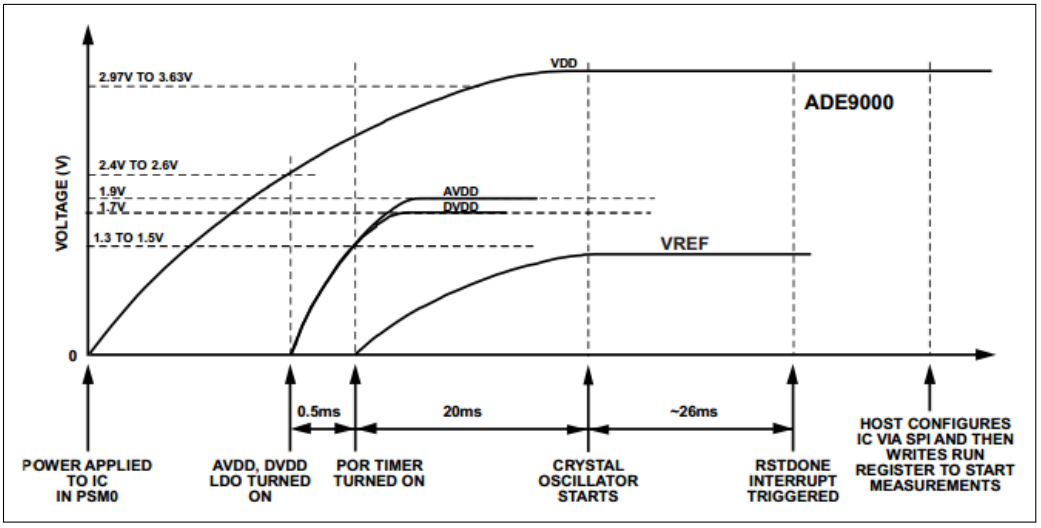


Figura 4.14: Sequenza di alimentazione per modalità PSM0

STUDENTSUPSI

Al termine di questa sequenza è possibile comunicare con il chip tramite la connessione SPI, in questa fase si deve rispettare un setup[4], questo per assicurarsi delle impostazioni corrette ad esempio sul guadagno dei canali ma non solo, infatti il chip ADE9000 necessita un settaggio software iniziale per poter iniziare ad operare. Questo processo di setup richiede:

1. Inizialmente si attende l'interrupt[5] RSTDONE, quindi che i pin IRQ0 e IRQ1 vadano a 0;
2. L'imposizione del pin PM1 e PM0 a livello logico basso (0) per entrare in modalità PSM0;
3. L'imposizione del canale di reset al valore logico alto (1);
4. In questo momento è possibile applicare tutte le impostazioni per i guadagni e altro, come descritto nel manuale del chip;
5. Infine, per avviare il DSP[6], si scrive il comando RUN, ovvero 1 nel registro RUN.

È inoltre importante che chip di misura sia immune alle interferenze radio esterne, per poter dare valori di misura esatti che non siano influenzati da disturbi esterni secondo la norma IEC 61000-4-3. Per questo il chip dispone di filtri antialising e filtri passa basso (LPF) interni.

[4]Predisposizione / impostazioni inizali del sistema per seguire le operazioni volute.

[5]Segnale asincrono che indica il "bisogno di attenzione" da parte di una periferica

[6]Il processore di segnale digitale o digital signal processor (DSP) è un processore dedicato e ottimizzato per eseguire in maniera estremamente efficiente sequenze di istruzioni ricorrenti (come ad esempio somme, moltiplicazioni e traslazioni)

STUDENTSUPSI

4.4.3 Correzioni e calcoli effettuati dall'ADE9000

Per avere una migliore accuratezza nella misura del valore rms e di energia, un filtro passa alto rimuove la componente continua nelle misure. Nel caso vi fosse un errore nella lettura della fase di una delle tre fasi tra corrente e tensione, l'ADE9000 da la possibilità di compensare questo errore tramite un filtro digitale che migliora la risoluzione nella lettura dell'energia attiva e reattiva a basse potenze. Questa correzione può essere da $-15°$ a $2.25°$ a 50 Hz con un accuratezza di $\pm 0.0001°$.

Un filtro passa basso ricava il valore in rms della misura, attenuando di 64 dB l'armonica fondamentale alla frequenza di 50 Hz così da diminuire le variazioni nel calcolo del valore rms, ottenendo un errore massimo di $\pm$0.064%.

Il chip caclola inoltre la somma delle correnti delle tre fasi più la somma e la sottrazione della corrente di neutro $IA + IB + IC \pm IN$ salvandone il risultato nel registro ISUMRMS.

La potenza attiva viene calcolata moltiplicando tra di loro le misurazioni in tempo reale di ogni canale di corrente e tensione della medesima fase, successivamente vengono applicate le correzioni di guadagno e di potenza che vegnono specificate dal programmatore $[(Ix \cdot xIGAIN) \cdot (Ux \cdot xUGAIN)] \cdot xWATT$

La potenza reattiva totale include la potenza reattiva della fondamentale e delle armoniche. Per ricavare questa potenza il chip sfasa di $90°$ la corrente di ogni canale sia della fondamenta che delle armoniche, questa misura viene moltiplicata per la tensione della medesima fase ed il risultato viene filtrato in un filtro passa basso. Dopo aver applicato le correzioni di guadagno a corrente e tensione e di sfasamento alla corrente si ottiene la potenza reattiva di quella fase. È possibile ricavare la potenza reattiva e apparente unicamente della fondamentale. L'energia che viene misurata istantaneamente viene salvata e quindi viene misurata l'energia cumulata durante tutto il periodo di misurazione.

STUDENTSUPSI

4.4.4 Calibrazione

La calibrazione di uno strumento di misura è necessario quando ciò che viene misurato non corrisponde esattamente, o in misura di quanto si vuole accurato lo strumento, al valore reale misurato. L'ADE9000, essendo un chip di misura è necessario che esso sia accurato, a maggior ragione quando esso esegue i calcoli per ricavare l'energia, la potenza etc...

L'accuratezza della calibrazione è uguale o minore all'accuratezza della sorgente che viene usata per la calibrazione. In questo processo bisogna quindi tenerne conto, otlre a considerare i sensori che vengono usati per la misura.

Il processo di calibrazione avviene semplicemente facendo una misura su dei valori di tensione e corrente conosciuti. La differenza tra il valore atteso ed il valore misurato sarà il valore che bisogna fornire al chip come correzione.

STUDENTSUPSI

4.5 SPI, comunicazione tra Raspberry e ADE9000

Il Serial Peripheral Interface (SPI) è un bus di comunicazione sincrono, full duplex, con architettura master/slave, originariamente sviluppato da Motorola. Esso di basa su dei piccoli registri, generalmente da 8 bit, posizionati uno sul dispositivo Master e uno per ogni dispositivo slave. La comunicazione avviene attraverso lo "shift" (spostamento) dei bit di questi registri, dove questi registri vengono visti come contigui l'uno all'altro. Il master può leggere e scrivere dati solo nei primi 8 bit del registro, lo slave lo può fare solo negli ultimi 8. Grazie a una linea riservata al segale di clock che sincronizza la comunicazione, lo shift può avvenire tra i 2 registri. SPI (Serial Peripheral interface) presenta una interfaccia hardware composta

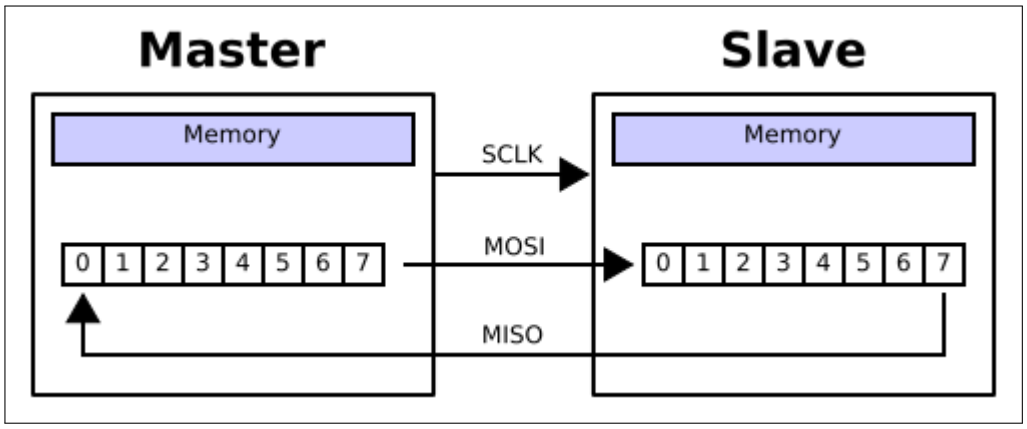


Figura 4.15: Collegamento base del protocollo SPI

da 4 pin principali:

- **SCLK** (Searial CLocK): In questo canale passa il segnale di clock, è generato della scheda e serve per la sincronizzazione dei canali usati per lo scambio di dati
- **MISO** (Master Input Slave Output): Serve per rendere il primo bit del registro Master contiguo all'ultimo bit del registro slave
- **MOSI** (Master Output Slave Input): Serve per rendere l'ultimo bit del registro Master contiguo al primo bit del registro slave
- **SS** (Slave Select): Questo pin viene utilizzato per selezionare lo slave desiderato

Per accedere ai valori misurati si fa uso del protocollo SPI, dove il chip ADE9000 è lo slave e quindi non inizia la comunicazione, ma aspetta di essere stimolato dal master. Il protocollo SPI si divide in tre parti:

1. Un comando di 16 bit viene inviato per determinare se si effettua un operazione di lettura o di scrittura e a quale registro si accede

STUDENTSUPSI

2. Seguono, nel caso dell'operazione di scrittura, i dati di 16 o 32 bit (a dipendenza del registro coinvolto), o i dati letti dal registro.

3. Nel caso della lettura vi è questa terza fase che consiste nell'assicurarsi che i dati letti siano corretti, per cui vengono inviati dei bit ridondanti di controllo (Cyclic redundancy check CRC).

Il pin SS, che viene usato per far partire la comunicazione SPI (portando da alto a basso il suo stato) e per scegliere lo slave col quale comunicare, deve rimanere in stato basso durante la comunicazione, previo l'aborto della comunicazione. La massima frequenza di clock supportata dal chip è di 20 MHz. Il primo comando di 16 bit chiamato CMQ_HDR conteine:

- CMD_HDR[15:4], i 12 bit più importante che contengono l'indirizzo del registro che si dovrà leggere o scrivere (ADDR[11:0])
- CMD_HDR[3] è il bit che specifica se l'operazione è lettura o scrittura. Se questo bit è 1 l'operazione è la lettura, altrimenti 0 scrittura.
- CMD_HDR[2:0] sono bit richiesti dal chip per la sua frequenza interna, e possono essere 000 o 111.

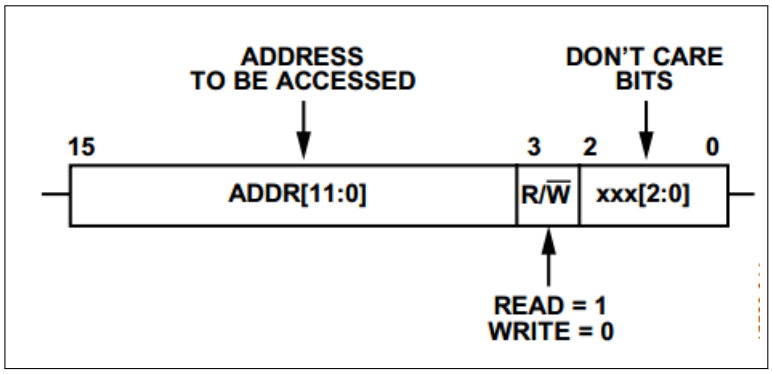


Figura 4.16: CMD_HDR [15:0]

Operazione di scrittura

La procedura per scrivere un dato all'interno di un registro del ADE9000 è:

1. pin SS passa da 1 a 0, si iniza la comunicazione SPI
2. vengono inviati i 16 bit CMD_HDR che contengono indirizzo del registro e operazione di scrittura
3. vengono inviati i dati di 16 o 32 bit da scrivere
4. pin SS passa da 0 a 1

STUDENTSUPSI

I dati venono letti o scritti nel formato Big Endian.

Operazione di lettura

In modo analogo funziona la lettura, con la particolarità che l'ADE9000 consente la lettura burst, ovvero la possibilità di lettura di registri multipli, ovvero quando si inizia inviando il primo CMD_HDR, si continua incrementando ad ogni lettura l'indirizzo del registro. Questa funzione permette di legger più registri che contengono, tutti insieme, il dato che si vuole trattare, come ad esempio il buffer della forma d'onda.

STUDENTSUPSI

Vi sono due tipi di collegamento, il collegamento con più slave select, dove il canale SS è presente per ogni dispositivo slave separatamente, mentre gli altri canali sono condivisi. In questa modalità è possibile selezionare lo slave col quale comunicare e ha il vantaggio di un elevata velocità di trasmissione dei dati, il limite invece è dato dal numero di pin che il controllore può dedicare per la selezione degli slave.

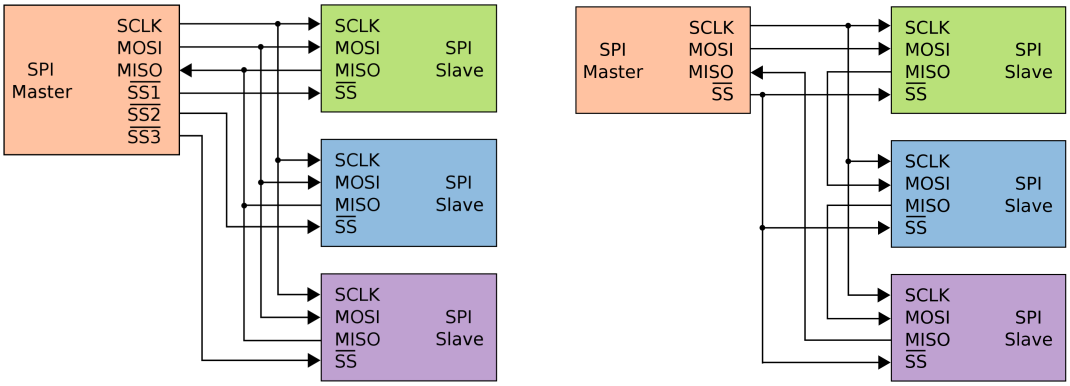


Figura 4.17: Collegamenti SPI, con più slave (sinistra), a catena (destra)

Invece il collegamento a catena, o daisy chain, ha la particolarità che tutti gli slave sono connessi uno dopo l'altro, formando una catena. In questa modalità si comunica simultanieamente con tutti gli slave e ha una velocità di trasferimento minore rispetto alla modalità con più slave.

STUDENTSUPSI

STUDENTSUPSI

Capitolo 5

Lavoro pratico

5.1 PCB

In questo capitolo si vuole fornire qualche dettaglio riguardo alla progettazione e costruzione del PCB[1], ovvero il prototipo sviluppato durante il progetto. A tale scopo si suddivide il prototipo in sezioni:

- Alimentazione;
- ADE9000;
- Raspberry.

Attualmente il PCB è assemblato e si alimenta correttamente. Si ha un programma demo che imposta i parametri iniziali dal Raspberry all'ADE9000 e scrive il comando RUN. Una volta chiarito questo capitolo sarà possibile vedere il procedimento di lavoro che ha portato allo stato attuale del progetto.

[1] Un circuito stampato (anche conosciuto come PCB ovvero "printed circuit board" in lingua inglese), nell'elettronica, è un supporto utilizzato per interconnettere tra di loro i vari componenti elettronici di un circuito tramite piste conduttive incise su di un materiale non conduttivo.

STUDENTSUPSI

5.1.1 Alimentazione

Innanzi tutto è importante ricordarsi che si sta progettando un dispositivo che sarà a contatto con tensioni di 230 V efficaci, corrispondenti a 325 V di picco. Per questo motivo la scheda è divisa in due parti, con alimentazioni separate, questo argomento verrà trattato più avanti.

Partendo dal principio, il PCB è stato progettato formato da quattro layer, ovvero quattro strati, separati da un dielettrico[2], come mostrato in figura 5.1. Si può notare che i layer 3V3 e GND sono due piani, il primo di alimentazione e il secondo di massa, mentre il top ed il bottom layer sono dei piani dove vengono depositate le piste che connettono le periferiche ed i componenti.

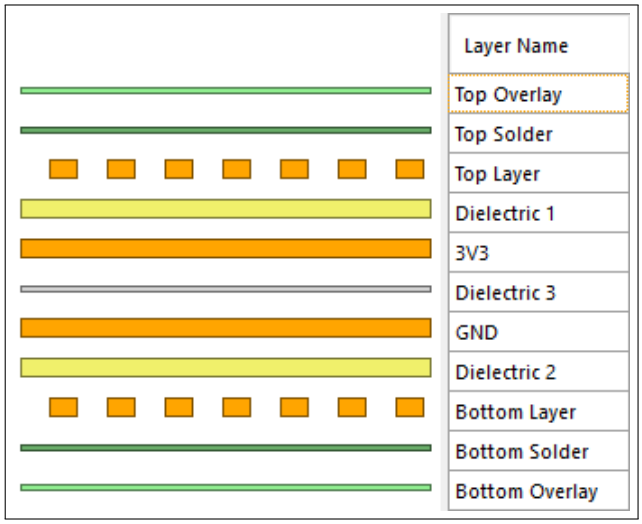


Figura 5.1: Layer PCB

Le alimentazioni richieste dai vari componenti sono tre:

- 5 V;
- 3.3 V;
- 1.8 V.

Inoltre bisogna considerare il consumo di corrente da parte delle principali periferiche, ovvero il chip di misura ADE9000 e il Raspberry. Secondo il datasheet del Raspberry, la potenza minima che deve essere garantita è di 3.5 W, mentre l'ADE9000 ha un consumo massimo di 17 mA con 3.3 V. L'alimentazione dello smart meter viene presa dalla prima fase tramite un convertitore AC/DC. In base a queste indicazioni e considerazioni si è scelto un convertitore AC/DC capace di fornire 5 W.

[2] agg. elettr. Che è caratterizzato da una bassa conduttività elettrica

STUDENTSUPSI

Come anticipato, il PCB sarà separato galvanicamente[3] in due parti, come mostrato in figura. Dove i pallini (vias) viola e arancione sono connessioni ai piani di massa (GND) e alimentatione (3.3 V). Da questa figura si può capire che c'è una netta separazione tra le due zone, anche i piani sono separati a loro volta.

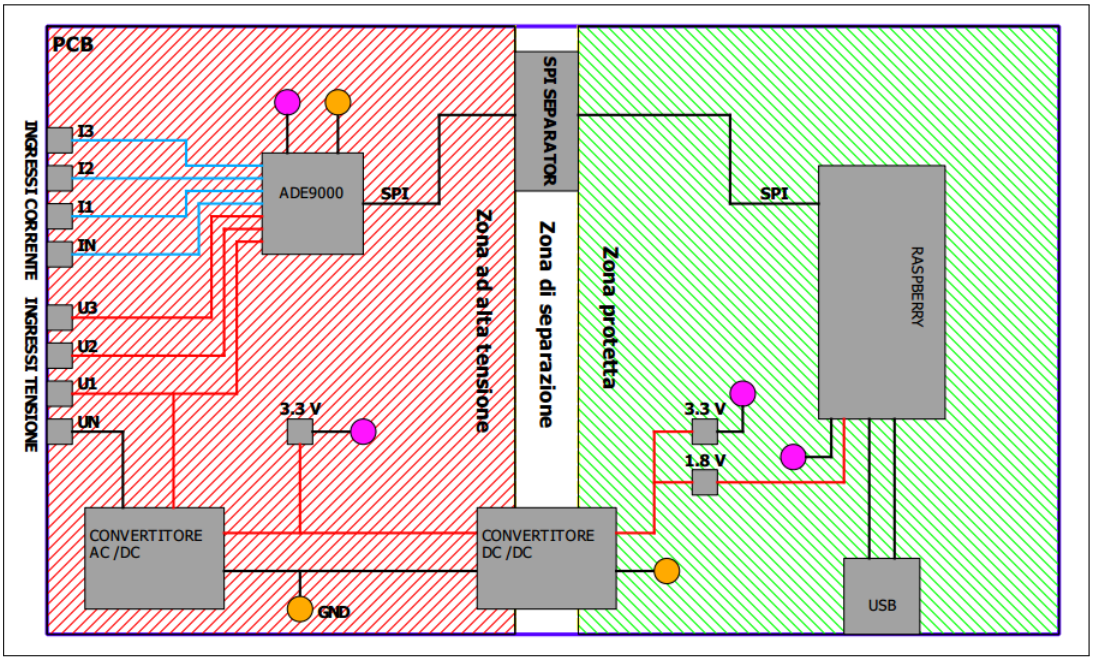


Figura 5.2: Schema a blocchi PCB

Questa separazione veine fatta per proteggere il Raspberry e le altre periferiche, proteggendo inoltre qualsiasi dispositivo sia collegato alla porta USB. Per applicare questa separazione si fa uso di un convertitore DC/DC che utilizza l'alimentazione fornita dal convertitore AC/DC e fornisce la stessa alimentazione di 5 V alla zona protetta, la separazione in questo componente è garantita perchè la trasmissione di energia avviene per effetto magnetico.

Per mantenere la comunicazione tra ADE9000 e Raspberry, si fa uso di chip chiamati SPI-separator, questi funzionano grazie a degli optocopler che trasmettono i segnali in modo ottico.

In questo modo otteniamo che la zona ad alta tensione viene alimentata dall'AC/DC che eleva il potenziale a 5 V, da questi uno stabilizzatore di tensione lineare crea l'alimentazione per l'ADE9000 di 3.3 V. Nella zona protetta il DC/DC è collegato a due stabilizzatori che forniscono 1.8 V e 3.3 V.

[3]Due circuiti si dicono galvanicamente separati (o anche elettricamente separati) se tra i due circuiti non può circolare alcuna corrente continua (detta anche galvanica)

STUDENTSUPSI

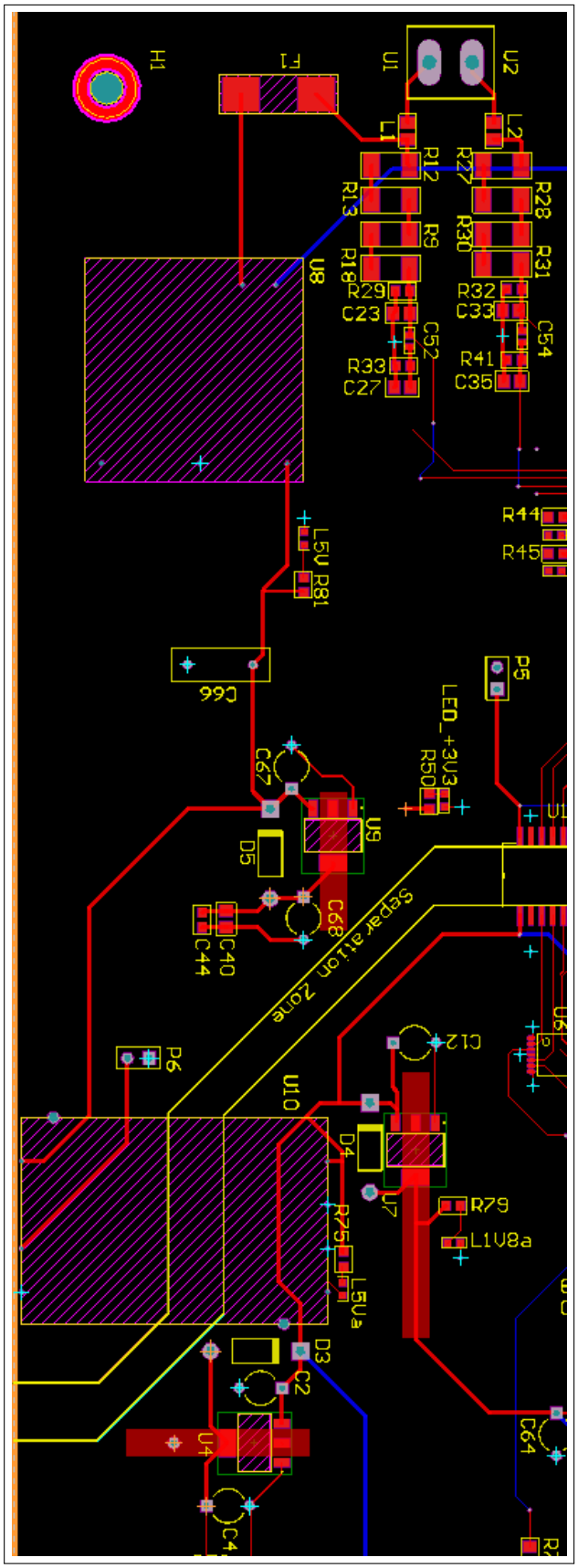


Figura 5.3: Layer PCB

STUDENTSUPSI

Nell'immagine 5.3 è possibile vedere la sezione che si occupa di alimentare tutti i circuiti presenti sul PCB con tutti i collegamenti. Le piste rosse sono depositate sul layer superiore, mentre le piste blu sono depositate sul layer inferiore. Si osservano:

- U8: Convertitore AC/DC RAC05-05SK della Racom;
- U10: Convertitore DC/DC CC10-0505SF-E della TDK-Lambda;
- U9 e U5: Stabilizzatori di tensione lineari a 3.3 V TLV1117-33CDCYR;
- U7: Sabilizzatore di tensione lineare a 1.8 V TLV1117-18CDCYR.

In appoggio, per filtrare correttamente l'alimentazione ed ottenere una tensione fissa, vi sono alcune capacità che filtrano evenutali ondulazioni. Si vuole informare il lettore che lo schema qui riportato è uno schema modificato ed ottimizzato, fatta eccezione per le distanze e spazi dei componenti, in quanto è ancora possibile ottimizzare questi fattori.

Il motivo di quanto scritto è l'aver osservato alcuni errori e alcune ottimizzazioni:

- I componenti U8 e U10 erano di dimensioni errate, per cui sul prototipo è stato eseguito un adattamento con l'ausilio di schede mille fori.
- I componenti U5, U7 e U9 erano previsti di tipo regolabile, con l'ausilio di resistenze calcolate per ottenere le tensioni volute. A causa di un errore sono stati comandati degli stabilizzatori con uscita di 1.8 V. Questo inconveniente ha causato un probabile guasto al chip di misura, siccome in questa configurazione lo stabilizzatore che doveva fornire 3.3 V avendo una tensione di riferimento di 1.25 V, forniva 4.2 V, non essendo il componente giusto per l'utilizzo che ne è stato fatto.

Il problema sopra descritto verrà trattato in seguito, nello specifico si vedrà in quale modo ci si è accorti del reale malfunzionamento del chip di misura e quali soluzioni si sono adottate.

STUDENTSUPSI

5.1.2 ADE9000 su PCB

Nell'immagine 5.4 viene mostrato una zona del PCB, più precisamente la parte contenente l'ADE9000 (chip U1) con l'interfacciamento ai punti di misura IN (corrente di neutro) e U3 (tensione terza fase). Il PCB, come detto prima è costruito con due piani di alimentazione e massa posti nel mezzo del PCB, tra il layer superiore e quello inferiore, per cui qualsiasi componente abbia necessità di essere collegato ad uno di questi due punti è sufficiente applicare un pad (foro che collega i punti interessati) come si può notare dall'immagine, dove si vedono i pallini con le croci blu (massa) e arancioni (3.3 V).

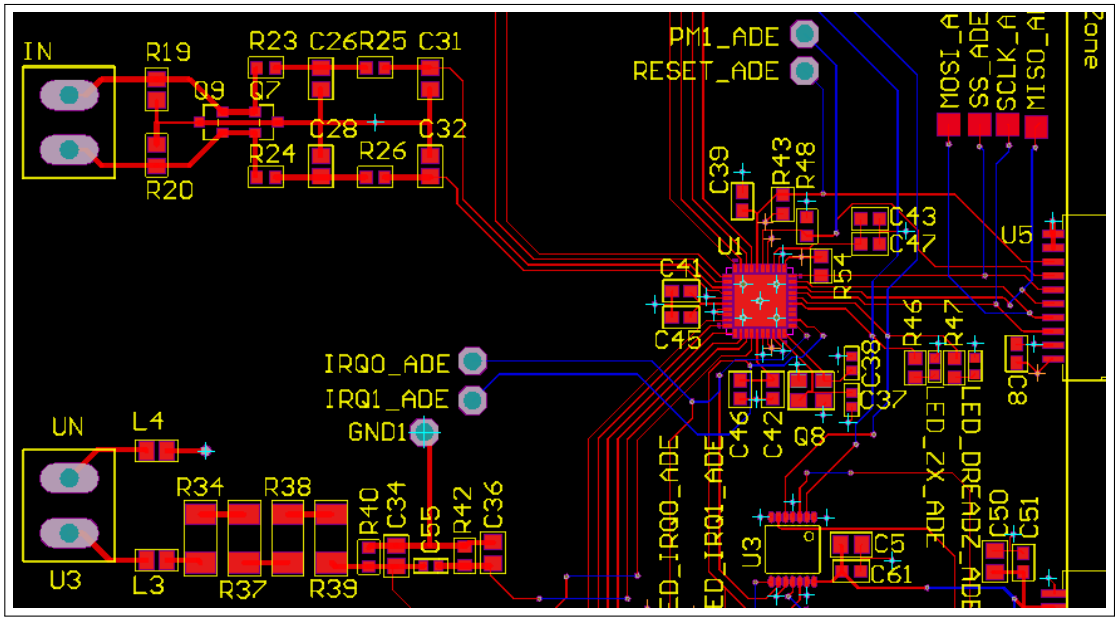


Figura 5.4: ADE9000 su PCB

Di contorno all'ADE9000 si possono vedere dei componenti tra resistenze e condensatori, adibiti a filtrare l'alimentazione e la tensione di riferimento di 1.25 V generata dal chip stesso. Si può vedere anche il quarzo (Q8), un componente piezoelettrico che ricevendo una tensione produce degli impulsi a frequenza di 24.576 MHz, ovvero un impolso ogni 40.7 nS. Questo componente fornisce la frequenza di clock col quale lavora internamente il chip di misura, quindi interessa anche la parte di campionamento del segnale, come tutti i filtri digitali spiegati precedentemente.

A destra dell'ADE9000 vi è un integrato che si occupa di tramsettere i dati che riceve senza che ci sia un contatto elettrico. Si tratta di un separatore SPI, quindi che riceve i quattro canali SPI e li riporta dall'altra parte. I dati possono essere trasmessi bidirezionalmente.

L'integrato U3 ha lo scopo di traslare il livello di tensione dei segnali che riceve. Più precisamente si occupa dei segnali di interrupt che manda o riceve l'ADE9000, siccome que-

STUDENTSUPSI

st'ultimo non è progettato per erogare o ricevere alte tensioni. Il chip U3 che si occupa di fare questo si chiama *Buffer translator logic level shifter*.

Per facilitare una possibile fase di correzione e ricerca di eventuali errori, il PCB è dotato di punti di misura dai quali poter verificare, con l'ausilio di un oscilloscopio il corretto funzionamento del circuito.

STUDENTSUPSI

5.1.3 Raspberry su PCB

Per connettere il compute module 3 sul PCB è stato installato un connettore SO-DIMM[4] a 200 contatti. Dall'immagine il componente X1 è il connettori SO-DIMM, da questo sono state fatte molteplici connessioni alle GPIO, alla connessione SPI e alle alimentazioni del Raspberry che vanno a due connettori a pettine con 60 contatti ogniuno. In questo modo è più semplice effettuare misure e controlli, è possibile testare ad esempio la comunicazione SPI, per verificare sel dal pin MISO vi è una trasmissione di dati, questo è molto utile in fase di debugging[5]. Nell'immagine si possono vedere alcuni punti di misura, dove PM1_ISO ser-

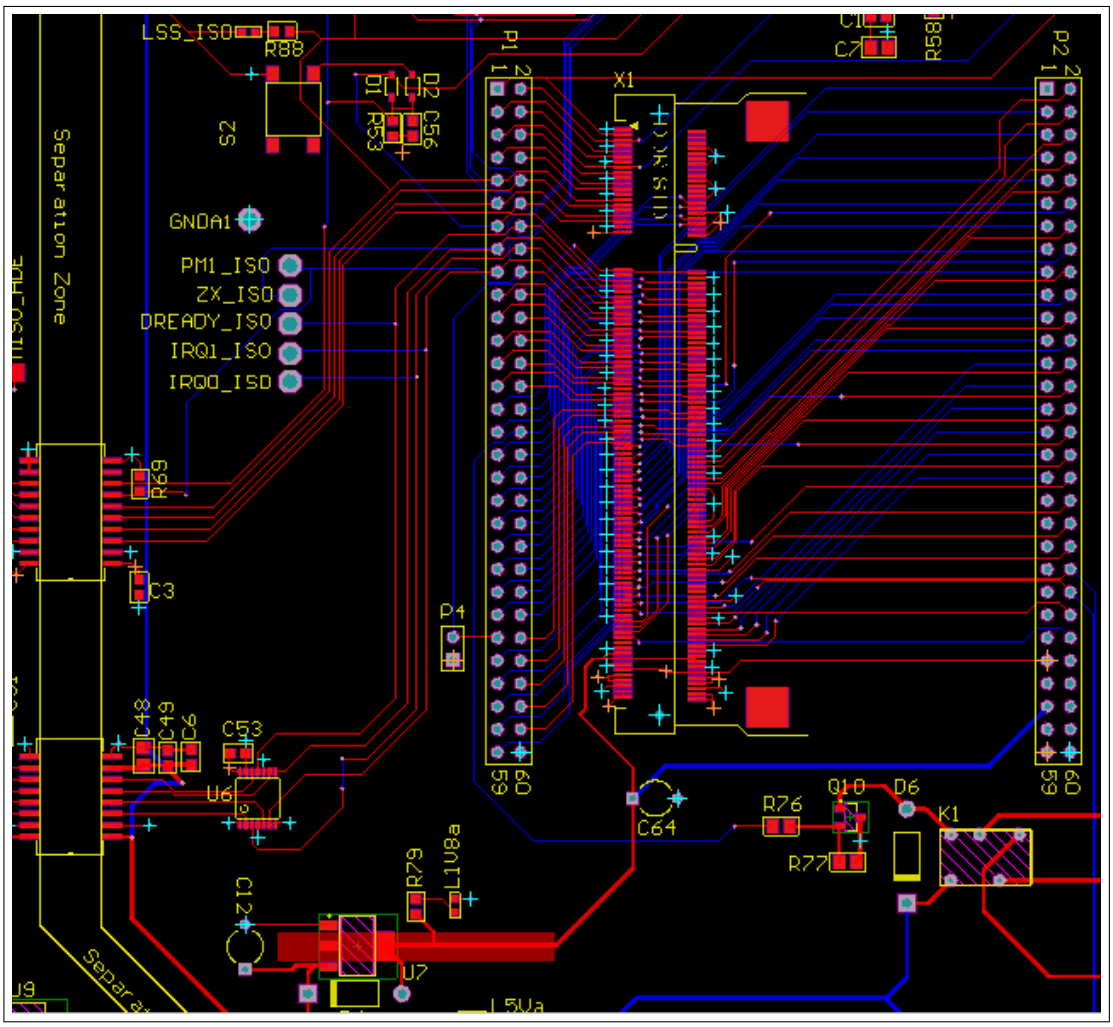


Figura 5.5: Raspberry pi CM3 su PCB

ve per controllare la modalità operativa dell'ADE9000, mentre gli altri quattro punti di misura sono interrupt dell'ADE9000, e informano se ci sono dati pronti da leggere. Inoltre si può

[4]Small Outline Dual In-line Memory Module

[5]Con debugging si intende la ricerca e la correzione degli errori di funzionamento di un sistema o di un programma.

STUDENTSUPSI

vedere il tasto di reset S2, dove qualora fosse necessario è possibile resettare l'ADE9000. In aggiunta al Raspberry, il PCB è stato progettato per poter essere testato con un Arduino zero. Oltre ad essere connesso all'ADE9000 tramite SPI ed all'alimentazione, il Raspberry controlla un Relay ed è connesso ad una RTC (Real Time Clock).

5.1.3.1 Real Time Clock

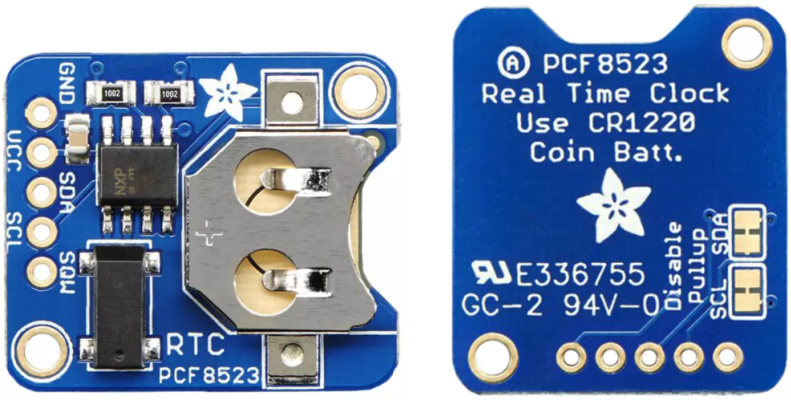


Figura 5.6: Adafruit PCF8523 Real Time Clock

Per fare in modo che il Raspberry abbia un regolare flusso del tempo e sappia collegare ogni misura all'orario esatto, si fa uso di un Real time clock, ovvero un circuito che comunica al Raspberry un conteggio del tempo.

Questo accorgimento viene fatto nell'eventualità che venga meno l'alimentazione, difatti questo RTC dispone di un alimentazione separata, ovvero da una batteria CR1220. Da specifiche la RTC tiene il tempo per 5 anni, tempo in cui la batteria si scarica.

È molto importante per le misure, sapere a che ora sono state fatte. Sopratutto per poter calcolare correttamente l'energia consumata.

STUDENTSUPSI

5.1.3.2 Relay

Il Relay ha lo scopo di controllare un teleruttore o contattore esterno. È possibile grazie a questo accorgimento, scrivere un programma che in base alle misure sul consumo, comanda questo Relay, che a sua volta comanda un teleruttore che inserisce o disinserisce un carico.

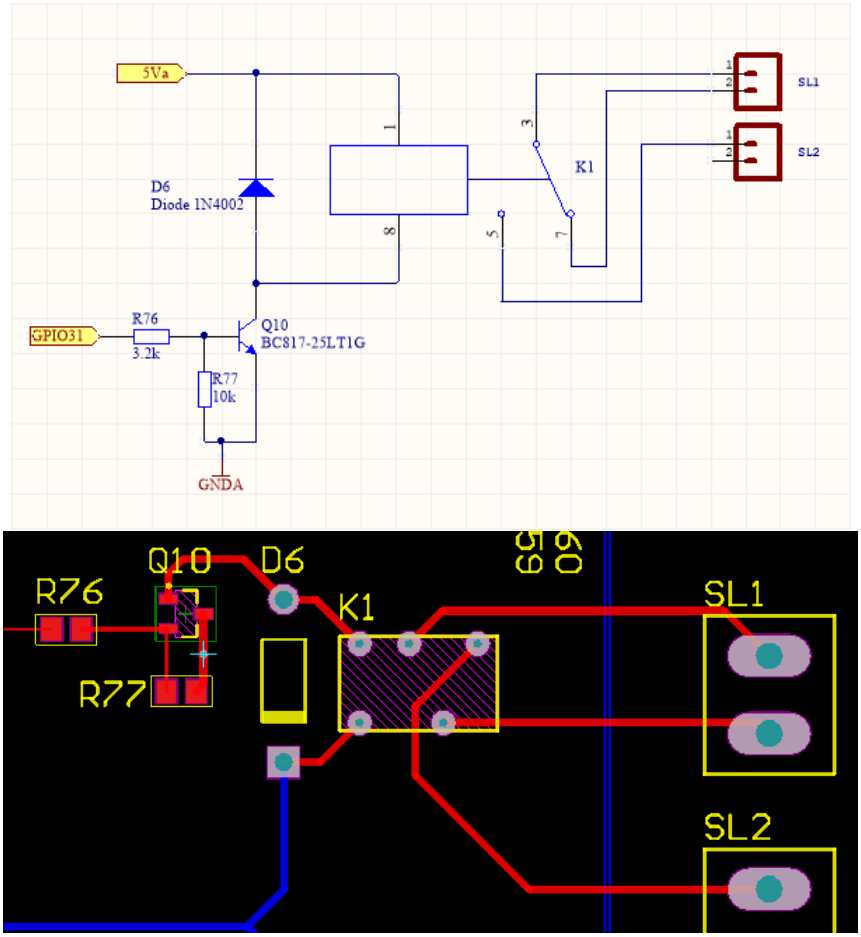


Figura 5.7: Relay, schema elettrico e su PCB

Il Raspberry è connesso alla base del trasistor che pilota il Relay, questo perchè la corrente che emette il Raspberry non è sufficiente a gestire il Relay, per cui con questo schema si è sicuri di pilotare correttamente il Relay senza recare danno al Raspberry.

STUDENTSUPSI

5.1.3.3 Risultato finale

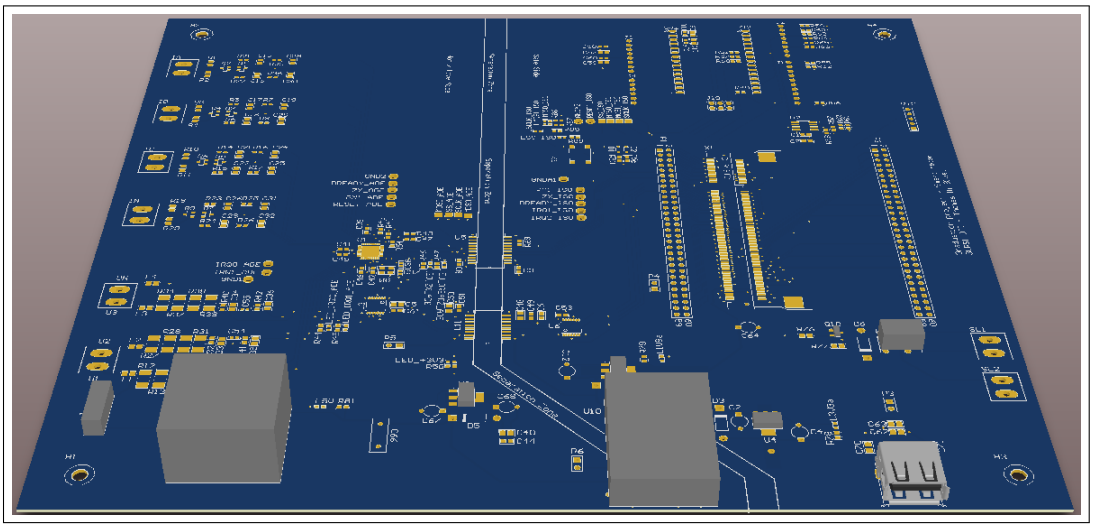


Figura 5.8: Visualizzazione 3D del PCB

Qui di seguito si vuole mostrare il PCB nel suo insieme.

Si può vedere la scheda in visualizzazione 3D (figura 5.8), successivamente sono riportate le stampe dei layer Top e Bottom (figure 5.9 e 5.10).

Infine in figura 5.11 si può osservare una foto del PCB assemblato con le indicazioni sulle parti prima discusse.

STUDENTSUPSI

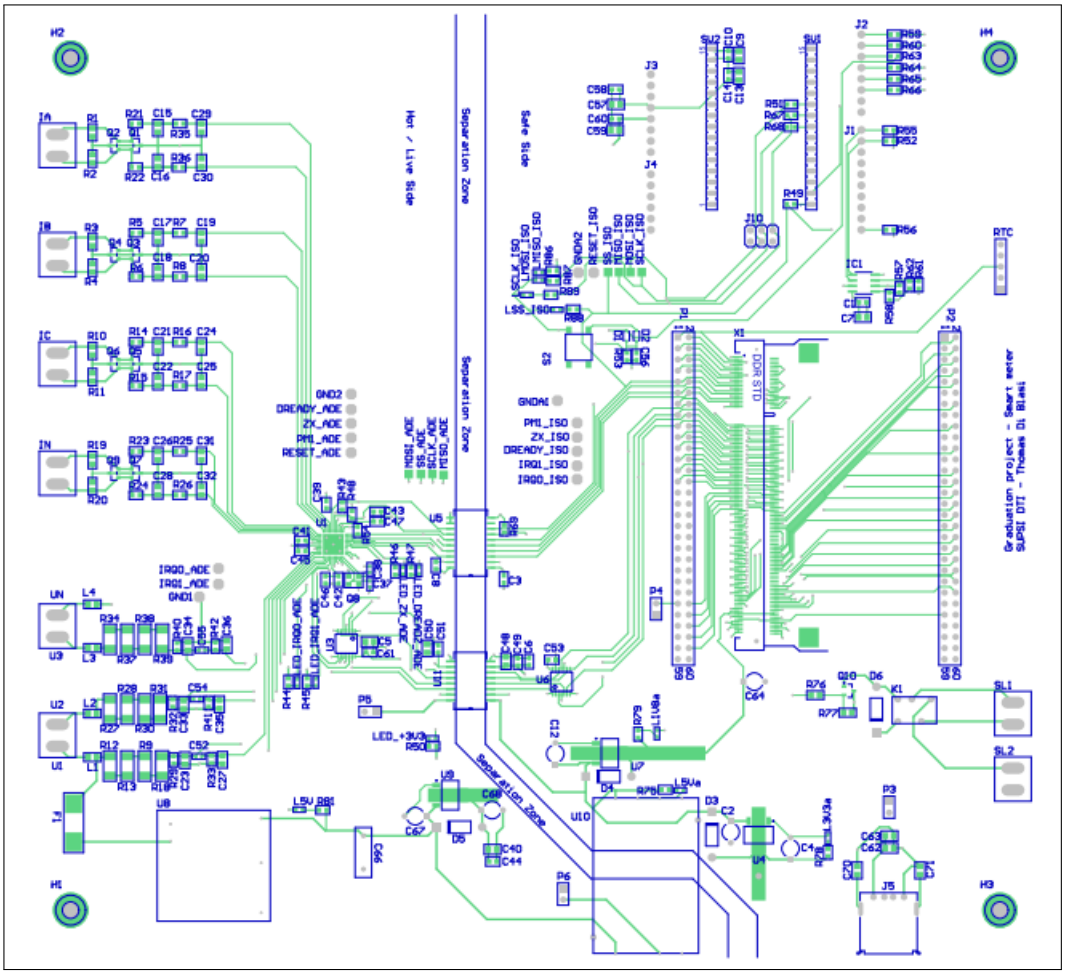


Figura 5.9: Top layer e top overlay

STUDENTSUPSI

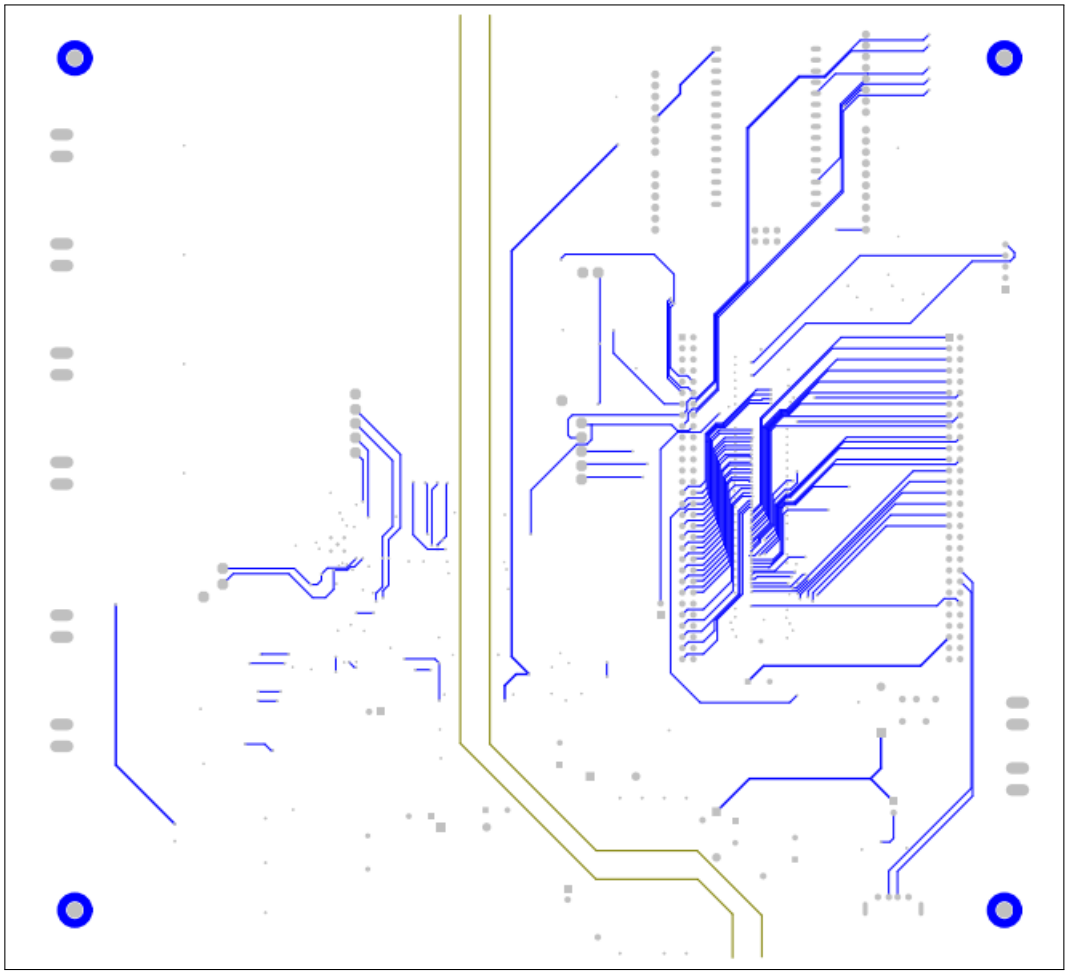


Figura 5.10: Bottom layer e bottom overlay

STUDENTSUPSI

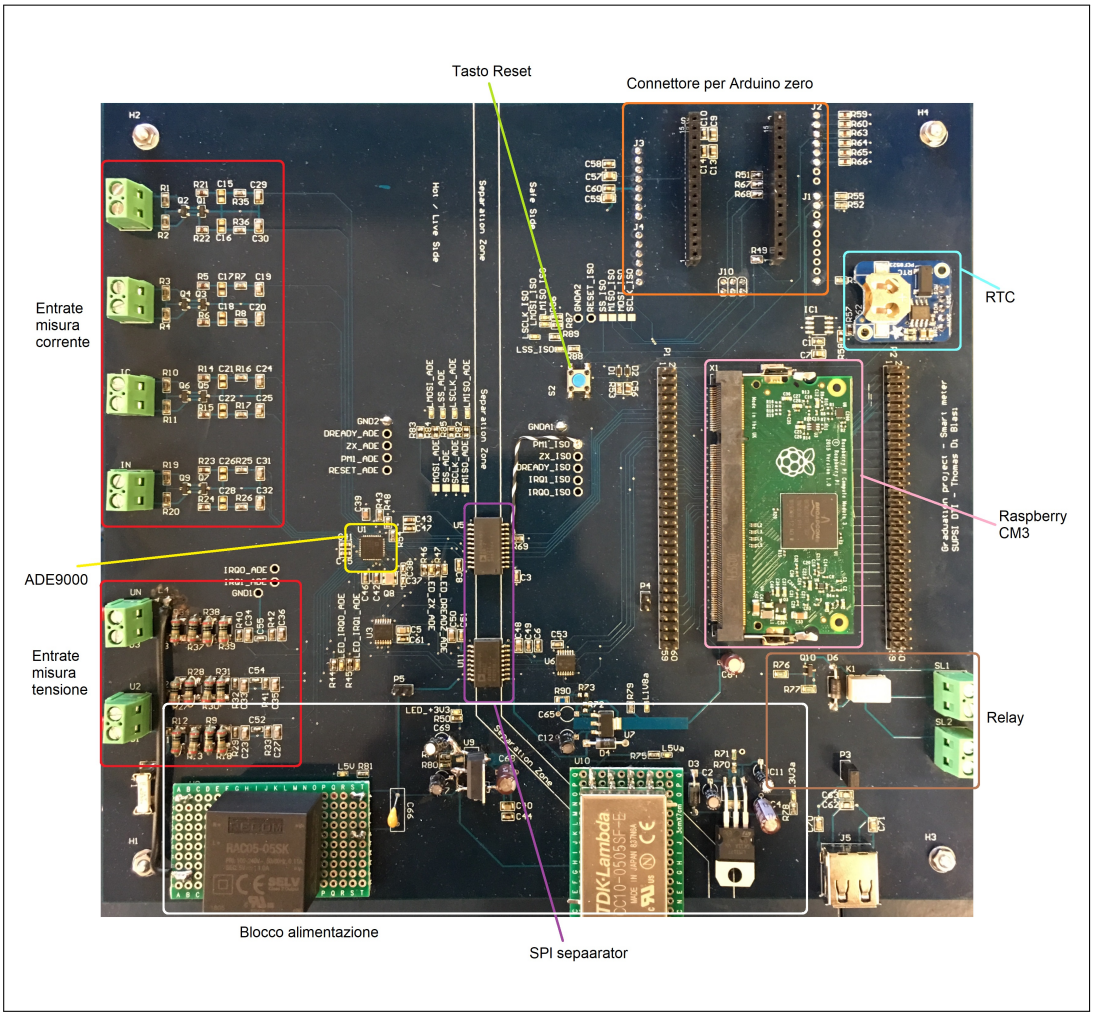


Figura 5.11: PCB finale

STUDENTSUPSI

5.2 Software

Il software che deve gestire le periferiche del PCB viene scritto nel linguaggio di programmazione Python. Sarà implementato nel Raspberry che si occuperà di comunicare col chip di misura e gestire tutte le periferiche del PCB. Prima di ciò bisogna configurare il Raspberry CM3.

5.2.1 Sistema operativo e attivazione SPI Raspberry

Come sistema operativo si è scelto di utilizzare quello ufficiale, prodotto dalla Raspberry, cioè Raspbian Stretch lite, siccome per questo sistema operativo si possono trovare diverse guide sul come attivare le periferiche. Di fabbrica il sistema operativo supporta Python 3 ed essendo fatto appositamente per il microprocessore BCM2837, non si prevedono problemi nell'utilizzo del Raspberry.

Per installare il sistema sulla memori eMMC incorporata si collega il Raspberry alla sua evaluation board, si ha a disposizione una porta HDMI per collegare un monitor ed una porta USB per collegare una tastiera. Per installare il sistema operativo, è necessario potersi

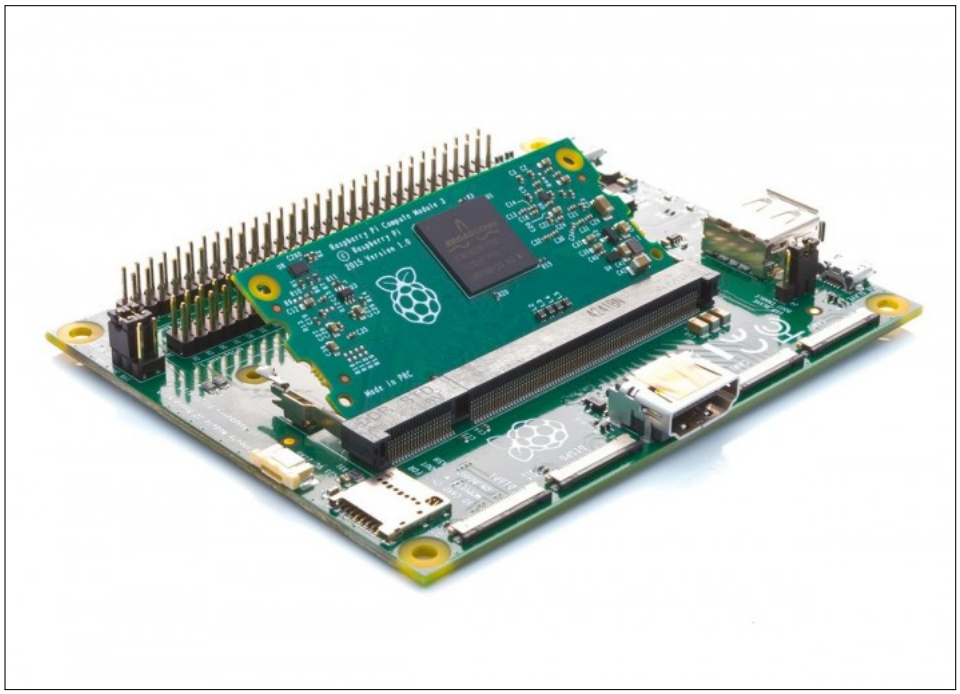


Figura 5.12: Raspberry CM3 con l'Evaluation board

interfacciare con il Raspberry dal computer. Per fare ciò su Windows, è sufficiente scaricare dal sito della Raspberry un installer. Successivamente si può installare il sistema operativo sul Raspberry, tramite il software Etcher.

STUDENTSUPSI

Una volta eseguita questa procedura, sul monitor collegato al Raspberry comparirà, prima della richiesta di nome utente e password, la dicitura che indica il sistema operativo Raspbian Stretch.

I passi seguenti sono quelli installare la libreria spidev per l'SPI e di aggiornare tutti i software installati. Una cosa che si è riscontrata durante il lavoro è il continuo cambiamento dell'indirizzo IP del Raspberry. Questo è uno svantaggio qual'ora non si avesse a disposizione un monitor, ma si volesse lavorare aprendo una connessione SSH[6] con il Raspberry. Le soluzioni sono due:

1. Richiedere al gestore della rete, in questo caso TiEdu, di assegnare un indirizzo IP statico, collegato al MacAdress dell'adattatore USB/Eterneth utilizzato.

2. Impostare, nel file boot/dhcpdp, un indirizzo IP statico, facendo attenzione ad utilizzare i primi tre numeri, uguali a quelli del Gateway (indirizzo IP del router) per avere accesso ad internet. Facendo così anche la scheda eterneth del computer deve avere i primi tre numeri dell'indirizzo IP uguale, altrimenti i due non si vedranno.

Siccome non si aveva un posto fisso di lavoro, ma si lavorava in tre sedi diverse, si è scelto il secondo metodo per potersi collegare al Raspberry tramite SSH.

Dopo questi passaggi è possibile lavorare con il Rapsberry, accedendo con il nome utente *pi* e la password *raspberry*, che sono impostati di fabbrica.

[6] In informatica e telecomunicazioni SSH (Secure SHell, shell sicura) è un protocollo che permette di stabilire una sessione remota cifrata tramite interfaccia a riga di comando con un altro host di una rete informatica

STUDENTSUPSI

5.2.2 Python

Figura 5.13: Python logo

Python è un linguaggio di programmazione ad alto livello, orientato agli oggetti, è un linguaggio multi-paradigma[7], che ha tra i principali obiettivi dinamicità, semplicità e flessibilità. Supporta il paradigma object oriented[8], la programmazione strutturata e molte caratteristiche di programmazione funzionale e riflessione.

È un linguaggio che possiede librerie open source, quindi scritte da tutti. È facile da capire e facile da correggere, nel caso ci fossero bug. Il fatto che sia di alto livello permette di ridurre il numero di comandi da scrivere per eseguire un operazione, con il risultato di essere veloce nello scrivere un programma.

Siccome è open source e conosciuto, ci sono moltissime persone che ne fanno uso e nel caso uno avesse qualche problema in rete si trovano svariati esempi che possono aiutare. Per questo in rete si trova moltissima documentazione e programmi già scritti per le più svariate applicazioni. Per il progetto in questione non c'è ancora nulla scritto in Python.

[7]Un paradigma è un modo di interpretare il codice che viene scritto. In tal senso si può scrivere lo stesso comando ed essere tradotto in linguaggio macchina in comandi diversi a seconda del paradigma usato.

[8]È un paradigma di programmazione che permette di definire oggetti software in grado di interagire gli uni con gli altri attraverso lo scambio di messaggi

5.2.3 Programma

Il programma che andremo a vedere non è ancora stato possibile verificarne il funzionamento. Come specificato nella sezione "Alimentazione" si è verificato un problema, per cui non è stato possibile comunicare col chip di misura. Ciò che si è riscontrato è l'assenza di risposta da parte dell'ADE9000.

Dopo un attenta valutazione, si è convinti che il codice di seguito riportato possa funzionare, siccome si è analizzato il protocollo SPI ed il metodo di comunicazione e verificato con l'oscilloscopio che i dati vengano trasmessi al chip ADE9000. Nella prima parte si fa

```
#Importazione librerie
import spidev                          # Libreria SPI
import time                            # Libreria delay
import RPi.GPIO as GPIO                # Libreria RPi.GPIO module

GPIO.setmode(GPIO.BCM)                 # choose BCM or BOARD, pinnaggio

spi = spidev.SpiDev()                  # Crea oggetto SPI
spi.open(0, 0)                         # Apre porta 0 SPI, device 0
spi.max_speed_hz = 5000000             # Massima frequenza di comunicazione SPI
spi.mode = 0                           # Clock polarity

PM1 = 17                               # PM1 pin 17
GPIO.setup(PM1, GPIO.OUT)              # PM1 uscita
GPIO.output(PM1, False)                # PM1 = 0

spi.cshigh = False                     # Chip Select va a 0 durante la comunicazione SPI

resp=spi.xfer2([0x48,0x00,0x00,0x01])  # RUN
```

Figura 5.14: Codice di Setup

un'importazione dei comandi per l'SPI e per il pinnaggio del Raspberry. La libreria *time* in questo esempio non viene riportato, ma è stato usato per effettuare un debugging del codice, distinguendo bene un invio di dati da un altro.

Grazie al comando *setmode(GPIO.BCM)* si dichiara di voler utilizzare la numerazione dei pin del processo.

La libreria *spidev* fornisce diversi comandi utili per la comunicazione SPI. Per iniziare la comunicazione bisogna prima di tutto creare un oggetto SPI ed assegnargli un numero ed una porta. Successivamente si stabilisce che la massima frequenza di trasmissione è di 5 MHz, una frequenza inferiore rispetto a quella supportata dal chip di misura, che può arrivare fino a 20 MHz. Il comando *mode = 0* serve per comunicare su quale fronte di clock devono essere letti i dati trasmessi, in questo modo i dati vengono letti sul fianco di salita del clock.

STUDENTSUPSI

Dopo tale operazione si collega la variabile PM1 al pin 17 del Raspberry, viene dichiarato che è un uscita e si imposta questa uscita al valore 0, entrando così in modalità PSM0, ovvero la modalità operativa.

Il comando *cshigh = False* serve per impostare il ChipSelect, uno dei quattro collegamenti SPI. In questo modo, come descritto nel manuale dell'ADE9000, il ChipSelect viene messo a 0 durante la comunicazione e rimesso a 1 al termine.

Dopo questi settaggi iniziale l'ADE9000 dovrebbe essere in operazione e pronto per fornire le misure. Quindi si fa uso del comando *xfer2* che prende i dati scritti in esagesimale e li invia attraverso il canale MOSI dell'SPI. Nello specifico vediamo cosa manda:

Il registro RUN ha l'indirizzo 0x480 e sono i primi 12 bit del CMD_HDR. successivamente viene specificato con il CMD_HDR[3] che viene fatta una scrittura con il bit 0. Infine per far partire il DSP bisogna scrivere il valore 1 in questo registro di 16 bit.

Quanto abbiamo visto è composto nel seguente modo:

	Indirizzo registro	R/W + 3 bit *don't care*	Dato da scrivere
Significato	ADDR_RUN	0 = R (Lettura)	Comando RUN
In esagesimale	0x480	0	0001
In binario	0100 1000 0000	0000	0000 0000 0000 0001

Per testare ulteriormente la corretta ricezione di un comando, è possibile fare riferimento a dei registri dell'ADE9000 appositamente creati per tenere l'ultimo comando ricevuto. Consultando questi registri è quindi possibile rileggere l'ultimo comando inviato. Ad esempio è possibile consultare i seguenti registri:

Nome registro	Indirizzo	Informazione
LAST_CMD	0x4AE	Ultimo CMD_HDR inviato
LAST_DATA_16	0x4AC	Ultimo dato di 16 bit inviato
LAST_DATA_32	0x423	Ultimo dato di 32 bit inviato

Per leggere un dato ricevuto attraverso il canale MISO dell'SPI, si fa uso del comando *readbytes(len)*, dove *len* serve per specificare la lunghezza del dato da leggere. Si voglio riportare alcuni dei registri su cui è possibile scrivere o leggere le misure:

Nome registro	Indirizzo	Informazione
AIGAIN	0x000	Guadagno corrente fase A
AVRMSOS	0x00D	Tensione RMS fase A
AWATTOS	0x00F	Potenza totale attiva fase A
AVAROS	0x010	Potenza totale reattiva fase A
AIFRMS	0x20E	ϕ fase A

STUDENTSUPSI

Capitolo 6

Consuntivo finale

Il budget a disposizione per questo progetto è stato fissa a 500.- CHF. Qui di seguito vengono riportate le spese che sono state fatte per costruire un prototipo. Per non andare troppo nel dettaglio vengono messi come unica spesa tutti i componenti di contorno, come condensatori, resistenze, fusibili e connettori. Così facendo si vedono i costi delle principali periferiche e del PCB:

Oggetto	Costo in CHF
Kit Raspberry cm3 con evaluation board	174.00
Energy metering chip ADE9000	13.40
Alimentazione	40.60
Blocco Relay	5.00
Real Time Clock	4.90
SPI separators	22.00
Arduino MKRZERO	25.00
Componenti di contorno	120.00
PCB da PCBway	250.00
Totale	655.00

In alimentazione sono inclusi tutti i componenti che si occupano di trasformare 230 V in 5 V, 3.3 V e 1.8 V, sia nella zona di alta tensione che nella zona protetta. Nella categoria *Blocco Relay* si vuole far vedere il costo di un circuito che da le capacità descritte precedentemente.

STUDENTSUPSI

STUDENTSUPSI

Capitolo 7

Conclusioni

Attualmente il prototipo dello Smart meter non è funzionante. Il fatto di non aver avuto a disposizione un evaluation board del chip di misura, per poter scrivere un programma parallelamente alla progettazione del PCB, ha reso molto importante l'ottenere un PCB funzionante il prima possibile.

Purtroppo la progettazione del PCB ha richiesto più tempo del previsto. La poca dimestichezza col software Altium ha fatto perdere diverse settimane per risolvere alcuni problemi. Inoltre l'elevato costo del progetto, che ha superato il budget previsto, è riconducibile alle dimensioni del PCB, fattore di cui non se ne aveva valutato attentamente l'importanza.

I lati positivi sono comunque molteplici, grazie a questo progetto, ci si è potuti rendere conto di molti dettagli ed accortezze. Ad esempio l'utilizzo di jumper per verificare le alimentazioni, o le dimensioni del PCB o il disegnare un footprint di un componente di cui non se ne trova la libreria. Si ha preso maggior dimestichezza con il software Altium, con il Raspberry e si è conosciuto il linguaggio di programmazione Python.

Ci si è resi conto di riuscire ad affrontare un progetto, organizzandolo strutturalmente, collaborando con le persone, il tutto per raggiungere un obiettivo. È stato entusiasmante, malgrado tutte le avversità, veder nascere il PCB o testare le GPIO del Raspberry o l'SPI per controllare dei LED.

Fino alla presentazione del progetto si tenterà comunque di ottenere un prototipo funzionante, ci sono ancora delle possibilità, delle altre strade che è possibile percorrere per arrivare alla soluzione.

Si proverà a testare la scheda con l'Arduino, per capire se è un problema hardware o software. In caso fosse la scheda ad avere il problema allora si dovrà riassemblarne un'altra.

STUDENTSUPSI

STUDENTSUPSI

Capitolo 8

Sitografia

ADE9000 Analog Devices
http://www.analog.com/en/products/ade9000.html#product-overview

ADE9000 datasheet
http://www.analog.com/media/en/technical-documentation/data-sheets/ADE9000.pdf

ADE9000 manuale tecnico
http://www.analog.com/media/en/technical-documentation/user-guides/ADE9000-UG-1098.pdf

Guida Altium
https://techdocs.altium.com/printpdf/231428

Altium tutorial
https://www.youtube.com/watch?v=MOedGOC40OU

Guida installazione Raspbian
https://www.raspberrypi.org/documentation/installation/installing-images/README.md

Documento Spidev
http://tightdev.net/SpiDev_Doc.pdf

SPI Raspberry
https://raspberrypi-aa.github.io/session3/spi.html