



**UNIVERSITA' POLITECNICA DELLE MARCHE**

**FACOLTA' DI INGEGNERIA**

**Corso di Laurea triennale Ingegneria elettronica**

**Progettazione, realizzazione e test di sistemi automatici di  
alimentazione e misurazione dell'energia basati su INA229 per  
dispositivi elettronici a bassa potenza**

**Design, implementation and testing of automatic power supply and  
energy measurement systems based on INA229 for low power  
electronic devices**

Relatore: Chiar.mo

Prof. **Biagetti Giorgio**

Tesi di Laurea di:

**Claudio Rausa**

**A.A. 2023 / 2024**

# INDICE

---

<b>1. INTRODUZIONE</b> .....	4
<b>2. COMPONENTI</b> .....	6
<b>3. MIGLIORAMENTI APPORTATI</b> .....	11
<b>3.1. Sostituzione della Raspberry Pi con un Microcontrollore con Interfaccia USB</b> .....	11
<b>3.2. Aggiornamento dei Componenti</b> .....	12
<b>3.3. Ottimizzazione della Tensione di Uscita e Integrazione di un Secondo LDO</b> .....	12
<b>4. ANALISI DEI COMPONENTI (CRITERI DI SELEZIONE, CARATTERISTICHE DESIDERATE E NON IDONEE)</b> .....	14
<b>4.1 Analisi dei Possibili Sostituti per il DAC e Scelta del Componente Finale</b> .....	15
<b>4.2 Analisi dei Possibili Sostituti per il TLP e Scelta del Componente Finale</b> .....	20
<b>4.4 Differenze di Collegamento tra INA226 e INA229</b> .....	21
<b>4.5 Analisi dell'LDO</b> .....	22
<b>4.5.1 Simulazione LDO</b> .....	25
<b>5. POMPA DI CARICA</b> .....	29
<b>5.1. Funzionamento</b> .....	29
<b>5.2. Analisi delle possibili pompe di carica e scelta del componente finale</b> .....	31
<b>5.3. Simulazione</b> .....	36
<b>6. REVISIONE DEL PROGETTO E DELLE MODIFICHE APPORTATE</b> .....	41
<b>7. PROBLEMA CON L'UTILIZZO DEI DUE LDO INSIEME</b> .....	43
<b>7.1. Soluzione: Utilizzo di un Booster</b> .....	44

7.2.	<i>Dimensionamento del TLV61046A</i> .....	49
7.2.1.	<i>Dimensionamento dei resistori</i> .....	49
7.2.2.	<i>Selezione dell'induttore</i> .....	51
7.2.3.	<i>Selezione dei condensatori di ingresso e uscita</i> .....	55
8.	<b>SELEZIONE DEI CONDENSATORI PER OGNI COMPONENTE</b> .....	57
9.	<b>OTTIMIZZAZIONI FINALI</b> .....	58
10.	<b>SBROGLIO DEL PCB: CARATTERISTICHE E ACCORTEZZE</b> .....	63
10.1.	<i>Miglioramento e ottimizzazione del PCB</i> .....	64
11.	<b>ANALISI DELLE MODIFICHE AL SOFTWARE</b> .....	67
11.1.	<i>Analisi power.c</i> .....	69
11.2.	<i>Analisi ina229.c</i> .....	75
11.3.	<i>Analisi relay.c</i> .....	82
11.4.	<i>Analisi config.h</i> .....	85
11.5.	<i>Analisi main.c</i> .....	86
12.	<b>MISURAZIONI</b> .....	88
13.	<b>CONCLUSIONI E MIGLIORAMENTI</b> .....	100
14.	<b>RINGRAZIAMENTI</b> .....	103
15.	<b>BIBLIOGRAFIA E SITOGRAFIA</b> .....	105

## 1. INTRODUZIONE

La presente tesi illustra e continua il percorso intrapreso durante il tirocinio universitario, focalizzato sulla progettazione e realizzazione di sistemi automatici di alimentazione e misurazione dell'energia, basati sul componente INA229, destinati a dispositivi elettronici a bassa potenza. L'obiettivo primario consisteva nel perfezionare un progetto preesistente (figura 1.1), il quale comprendeva un Linear Drop-Out Regulator (LDO) comandato da un Convertitore Digitale-Analogico (DAC). L'uscita del LDO risultava connessa a uno dei tre photorelay selezionabili attraverso un Multiplexer (MUX), i quali assumevano la funzione di regolare la sensibilità dello strumento, permettendo di misurare correnti di differenti intensità. Il controllo dell'intero sistema in uscita veniva gestito da un componente digitale, il quale effettuava un campionamento della tensione e della corrente in uscita, per poi procedere alla loro digitalizzazione mediante un Convertitore Analogico-Digitale (ADC). La comunicazione con i dispositivi ADC e DAC era stata gestita tramite I2C, utilizzando una scheda Raspberry Pi 3.

Come prima cosa si è svolta un'analisi approfondita delle specifiche del progetto esistente e delle sue criticità, con l'obiettivo di proporre miglioramenti significativi che potessero ottimizzare le prestazioni

complessive del sistema. In questo contesto, sono stati affrontati diversi aspetti relativi alla progettazione, alla configurazione e all'implementazione dei componenti elettronici, nonché alla validazione dei risultati ottenuti attraverso test sperimentali e simulazioni.

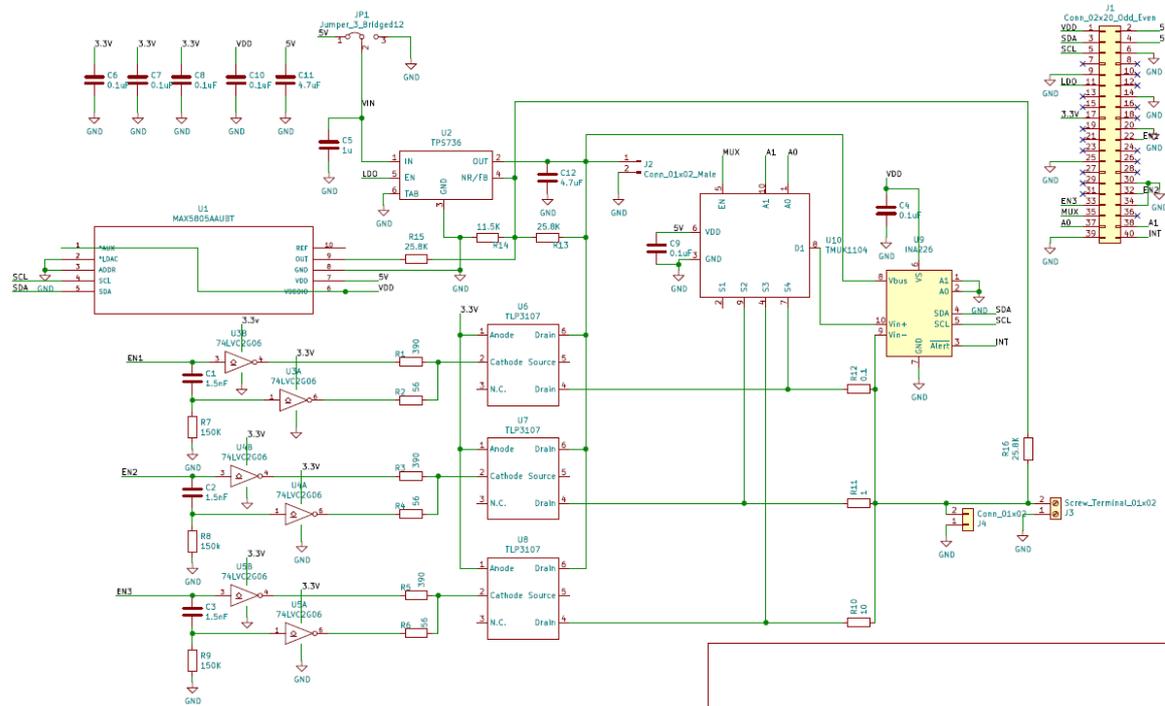


Figura 1.1: Schema circuitale del progetto esistente da modificare

## 2. COMPONENTI

L' LDO è generalmente un componente economico che regola e controlla la tensione fornita in ingresso da un alimentatore e dà in uscita una tensione stabile più bassa. I vantaggi di un regolatore LDO rispetto ad altri regolatori di tensione CC-CC includono: l'assenza di rumore di commutazione, dimensioni del dispositivo più piccole e una maggiore semplicità di progettazione (di solito è costituita da un riferimento, un amplificatore e un elemento passante). Lo svantaggio è che i regolatori lineari CC devono dissipare calore per funzionare.

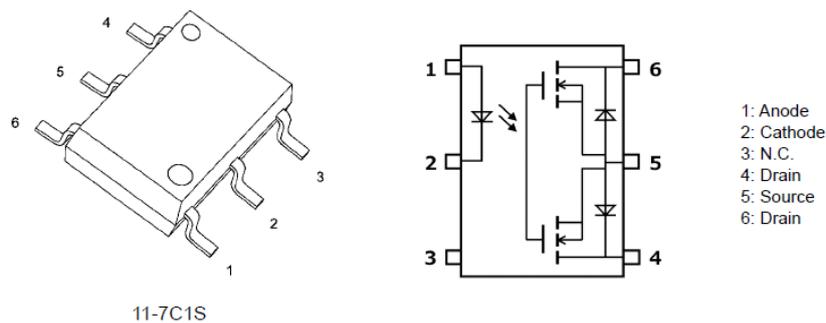
Nelle diverse applicazioni, i regolatori lineari a basso dropout (LDO) vengono impiegati principalmente per separare un carico sensibile da una fonte di alimentazione suscettibile a rumore. Tali regolatori dissipano potenza tramite il pass transistor o il MOSFET utilizzato per regolare e mantenere la tensione di uscita al valore richiesto. Pertanto, la dissipazione di potenza associata agli LDO può rappresentare un significativo svantaggio in termini di efficienza e può provocare problemi termici. Di conseguenza, minimizzare la dissipazione di potenza degli LDO è un aspetto cruciale per aumentare l'efficienza del sistema e prevenire possibili complicazioni termiche.

Durante il normale funzionamento, il componente passante agisce come una sorgente di corrente per il controller di tensione. Questo componente passante è controllato da un segnale di controllo compensato proveniente dall'amplificatore di errore, che rileva la tensione di uscita e la confronta con la tensione di riferimento. Tutti questi blocchi funzionali influenzano le prestazioni degli LDO. Le specifiche fornite dai produttori di LDO includono sempre dettagli sulle prestazioni di questi elementi funzionali. [1]

Il DAC è un componente elettronico che converte un segnale digitale in un segnale analogico, il suo utilizzo in questo circuito è quello di modificare la tensione di uscita del LDO tramite un partitore di resistenze, in modo tale da essere uguale al valore in digitale che riceve via I2C.

Il photorelay è costituito da due foto MOSFET accoppiati otticamente a un led a infrarossi (figura 2.1). La bassa resistenza e l'elevata corrente nello stato ON lo rendono adatto per applicazioni di controllo della linea elettrica; infatti, la sua funzione nel circuito è quella di fare portare la tensione d'uscita dell'LDO sulla resistenza ad esso collegata.

Per consentire la variazione del fondo scala, sono stati integrati tre circuiti identici, ciascuno dotato di resistori di uscita con valori differenziati e una tolleranza del 0.1%. Questa precisione è essenziale per garantire la precisione delle misurazioni. Quando il segnale di abilitazione viene attivato dalla scheda Raspberry Pi, gli inverter associati portano la loro uscita a livello logico basso, provocando l'illuminazione del LED a infrarossi. Ciò attiva i MOSFET interni, permettendo il passaggio di corrente. L'introduzione di due inverter ha lo scopo di accelerare il processo di attivazione.



*Figura 2.1: rappresentazione del photorelay*

L'INA è un misuratore di potenza digitale di alta precisione dotato di un convertitore analogico-digitale (ADC) appositamente progettato per applicazioni di rilevamento della corrente. Questo dispositivo rileva la corrente che attraversa le resistenze, chiamate resistenze di shunt,

calcolandola in base alla caduta di potenziale su di esse. Data la sua capacità di assorbire correnti estremamente basse, si presume che la corrente misurata sia rappresentativa di quella che scorre nel carico. [2]

Attraverso l'uso di un multiplexer (MUX), viene selezionata la resistenza attraverso la quale scorre la corrente da misurare, per poi convertire il valore analogico in digitale tramite il suo ADC integrato. Successivamente, il valore digitalizzato viene trasmesso tramite il protocollo di comunicazione I2C alla scheda Raspberry Pi per ulteriori elaborazioni.

Il multiplexer (MUX) è un dispositivo che consente la selezione tra diversi segnali di ingresso, analogici o digitali, instradando il segnale selezionato verso una singola linea di uscita. Questa selezione è determinata da un insieme separato di ingressi digitali noti come linee di selezione. Un multiplexer con  $2^N$  input possiede N linee di selezione, le quali vengono utilizzate per determinare quale segnale di ingresso sarà trasmesso all'uscita, attraverso valori binari applicati a tali linee. L'utilizzo di un multiplexer consente a diversi segnali di ingresso di condividere una risorsa o dispositivo, come ad esempio un convertitore analogico-digitale o un mezzo di

trasmissione delle comunicazioni, evitando la necessità di utilizzare un dispositivo separato per ciascun segnale di ingresso. [3]

Nel contesto specifico del progetto, il multiplexer viene impiegato per selezionare la resistenza desiderata in base al fondo scala richiesto. Ogni resistenza è collegata da un'estremità a un ingresso del multiplexer, mentre l'altra estremità è collegata insieme al pin  $V_{IN-}$  dell'INA. Questa configurazione consente di prelevare la tensione ai capi delle resistenze, consentendo così la misurazione della corrente che attraversa la resistenza selezionata.

### 3. MIGLIORAMENTI APPORTATI

#### *3.1. Sostituzione della Raspberry Pi con un Microcontrollore con Interfaccia USB*

Al fine di ottimizzare il controllo dei bus dei convertitori DAC e ADC e semplificare l'utilizzo complessivo del sistema, si è pensato in un primo momento di sostituire la scheda Raspberry Pi 3 con un microcontrollore dotato di interfaccia USB, come ad esempio la Raspberry Pi Pico. Questa modifica consentirebbe una gestione più efficiente dei bus di comunicazione e una trasmissione dei dati più rapida e stabile tramite USB verso un PC, migliorando significativamente l'esperienza utente.

Tuttavia, in ultima analisi, si è deciso di non procedere con la sostituzione della scheda Raspberry Pi 3, optando invece per l'implementazione di un componente che consentisse l'utilizzo complementare della Raspberry Pi Pico.

### ***3.2. Aggiornamento dei Componenti***

Per migliorare le prestazioni complessive del sistema, sono stati identificati diversi componenti da aggiornare. In particolare, si è proposta la sostituzione del DAC a 12 bit con un componente più preciso a 16 bit, al fine di garantire una maggiore precisione nelle misurazioni. È stata inoltre valutata la sostituzione del componente TLP3107 con una soluzione più recente, data la sua obsolescenza e la conseguente difficoltà di reperibilità. Infine, l'introduzione del nuovo chip INA229, per consentire una maggiore risoluzione, potenza e velocità di campionamento del sistema.

### ***3.3. Ottimizzazione della Tensione di Uscita e Integrazione di un Secondo LDO***

Un'altra area di intervento è stata l'ottimizzazione della tensione di uscita del sistema. Data la limitazione della tensione minima del regolatore lineare a basso dropout (LDO) a 1V, si è considerata l'integrazione di un secondo regolatore LDO, come l'LT3082, destinato a gestire le tensioni più basse al posto del componente TPS736. L'obiettivo iniziale è stato quello di trovare un

modo per utilizzare in modo sinergico entrambi i regolatori, massimizzando così l'efficienza e l'affidabilità del sistema, oltre a garantire una tensione di uscita ottimale per le specifiche applicazioni richieste. Purtroppo, dopo aver riscontrato alcuni problemi si è deciso di sostituire definitivamente il TPS736 con l'LT3082. Questo però ha richiesto ulteriori modifiche che verranno elencate e spiegate in maniera dettagliata in seguito.

#### **4. ANALISI DEI COMPONENTI (CRITERI DI SELEZIONE, CARATTERISTICHE DESIDERATE E NON IDONEE)**

Al fine di sostituire i componenti obsoleti con soluzioni aggiornate, è stata condotta un'approfondita ricerca tra i prodotti attualmente disponibili sul mercato. Questa ricerca è stata effettuata attraverso i siti web dei principali produttori e distributori di componenti elettronici, tra cui Texas Instruments, Linear Technology, Toshiba, Analog Devices e Mouser Electronics. Durante questa fase di ricerca, sono state prese in considerazione diverse specifiche tecniche che i nuovi componenti dovevano soddisfare per garantire il corretto funzionamento del sistema.

Uno degli aspetti chiave considerati durante la ricerca è stato il tipo di packaging del componente. È stato prioritario selezionare componenti con un packaging di tipo surface-mount device (SMD), in quanto questo formato è più adatto per l'integrazione su schede a circuito stampato doppio strato, garantendo una maggiore densità di componenti e una migliore efficienza spaziale.

## ***4.1 Analisi dei Possibili Sostituti per il DAC e Scelta del Componente***

### ***Finale***

Per quanto riguarda il DAC, il componente precedentemente utilizzato presentava un'accuratezza di 12 bit, una caratteristica che si è deciso di migliorare cercando un componente a 16 bit per ottenere una risoluzione superiore. Durante la fase iniziale di ricerca, sono stati individuati diversi possibili sostituti, tra cui il DAC161S055, il DAC81401, il DAC80502 e il MAX5705. Tuttavia, nessuno di questi componenti soddisfaceva appieno le caratteristiche desiderate per diverse ragioni:

- Il DAC161S055 è stato scartato perché presentava un riferimento di tensione esterno, il che avrebbe compromesso la precisione dello strumento a causa dell'aggiunta di rumore esterno. [4]
- Il DAC81401 è stato ritenuto non idoneo a causa di un range di uscita troppo elevato rispetto alle esigenze del progetto. [5]
- Il DAC80502 sarebbe stata una valida alternativa; infatti, avendo due canali d'uscita avrebbe consentito l'utilizzo di un solo componente per controllare entrambi i regolatori LDO. Tuttavia, questa soluzione

avrebbe comportato difficoltà nell'utilizzare i due regolatori alla stessa tensione simultaneamente. [6]

- Infine, il MAX5705, essendo a 12 bit, non soddisfaceva il requisito di risoluzione a 16 bit. [7]

Il componente finale selezionato, il DAC80501, soddisfa tutti i requisiti richiesti per il progetto. Dotato di una risoluzione a 16 bit, presenta un riferimento interno fisso da 2,5V e un range d'uscita da -0,3V a 6,3V, ideale per le esigenze dell'applicazione. È dotato di un'uscita singola e può essere controllato tramite interfaccia I2C o SPI, in base al collegamento del pin SPI2C. La sua modalità SPI utilizza un'interfaccia seriale a 3 fili e supporta frequenze di clock fino a 50MHz, mentre la modalità I2C può operare a velocità standard (100 Kbps), veloce (400 Kbps) o veloce plus(1,0 Mbps), a seconda delle esigenze del sistema.

Esso interpreta come segnale digitale alto una tensione minima di  $V_{IH} = 1,62V$ , e come segnale basso una tensione massima di  $V_{IL} = 0,45V$ , condizione che riusciamo a rispettare in quanto la raspberry emette segnali da 0V a 3,3V.

I pin per la comunicazione SPI/I2C sono i seguenti:

- SCLK/SCL: Questo pin è dedicato per ricevere il clock, utilizzato da entrambi i sistemi.

- SDIN/SDA:

- Modalità SPI: Questo è l'input dati dell'interfaccia seriale. I dati vengono sincronizzati nel registro a scorrimento di ingresso su ciascun fronte di discesa del pin SCLK.

- Modalità I2C: I dati vengono sincronizzati in input o in output dal registro di ingresso. Questo pin è una linea dati bidirezionale SDA che deve essere collegata alla tensione di alimentazione con un resistore pull-up esterno.

- SPI2C: Questo pin è utilizzato per la selezione dell'interfaccia.

- Interfaccia digitale in modalità SPI se SPI2C = 0.

- Interfaccia digitale in modalità I2C se SPI2C = 1.

Il pin SPI2C deve rimanere statico dopo l'accensione del dispositivo.

- \*SYNC/A0:

- Modalità SPI: Questo ingresso abilita i dati seriali attivi bassi. Quando il segnale diventa basso, il registro a scorrimento dell'ingresso dell'interfaccia seriale è abilitato.

- Modalità I2C: Controlla gli ultimi tre bit dell'indirizzo del componente in base al suo collegamento, la tabella 4.1 riporta tutte le combinazioni possibili.

ADDRESS TYPE	MSB							LSB
	AD6	AD5	AD4	AD3	AD2	AD1	AD0	R/ W
General address	1	0	0	1	See Table 8-3 (target address column)			0 or 1

TARGET ADDRESS	A0 PIN
1001 000	AGND
1001 001	VDD
1001 010	SDA
1001 011	SCL

*Tabella 4.1: descrive l'utilizzo di A0*

Se si dovessero utilizzare due DAC, per pilotarli con I2C, il pin A0 dovrà essere collegato in maniera differente per garantire che abbiano indirizzi univoci.

L'architettura dell'uscita è basata su una struttura ladder rail-to-rail con un amplificatore buffer di uscita. Il dispositivo include un riferimento interno da 2,5V che può essere amplificato o diviso tramite il registro di guadagno (indirizzo 4h).

$$\text{Essendo } V_{\text{OUT}} = \frac{\text{DAC}_{\text{DATA}}}{2^N} \cdot \frac{V_{\text{REFIO}}}{\text{DIV}} \cdot \text{GAIN}$$

Dove:

- $N = \text{risoluzione in bit} = 16$ ;
- $DAC_{DATA} = \text{equivalente decimale del codice binario caricato nel registro}$

DAC (indirizzo 8h)  $DAC_{DATA}$  varia da 0 a  $2^N - 1$ ;

- $V_{REFIO} = \text{tensione di riferimento interno del DAC} = 2,5V$ ;
- $DIV = 1$  (predefinito) o 2, come impostato dal bit REF-DIV nel registro

GAIN (indirizzo 4h);

- $GAIN = 1$  o 2 (default), come impostato dal bit BUFF-GAIN nel registro

GAIN (indirizzo 4h).

Per ottenere un'uscita che varia da 0V a 5V, il parametro di guadagno (GAIN)

deve essere impostato su 2 e il parametro di divisione (DIV) su 1,

conformemente alle specifiche del progetto. [8]

## ***4.2 Analisi dei Possibili Sostituti per il TLP e Scelta del Componente***

### ***Finale***

Nel processo di ricerca dei possibili sostituti per il TLP3017, con una resistenza di ON ( $R_{ON}$ ) di  $60m\Omega$ , in una prima ricerca si è deciso di valutare il TLP3555A, TLP3556A, TLP3823 e TLP3825, che non rispettavano le specifiche desiderate poiché presentavano valori di  $R_{ON}$  superiori al componente precedente (rispettivamente  $100m\Omega$ ,  $200m\Omega$ ,  $150m\Omega$ ,  $500m\Omega$  [9]). Questo aspetto risultava non congruente con le aspettative del progetto, poiché una resistenza ON più elevata comporta un maggiore consumo energetico del componente.

Successivamente, sono state prese in considerazione altre opzioni, tra cui il TLP3106A e il TLP3107A, entrambi con un valore di  $R_{ON}$  rispettivamente di  $30m\Omega$  [10] e  $40m\Omega$  [11], soddisfacendo quindi il requisito di bassa resistenza. Tuttavia, il TLP3106A è stato scartato a causa dell'assorbimento di corrente più elevato quando il componente è spento, che risultava essere più del doppio rispetto al TLP3107A.

### ***4.3 Migliorie sul MUX***

Nel circuito precedente venivano utilizzati solo 3 dei 4 ingressi disponibili del MUX. Per sfruttare appieno le sue capacità è stato deciso di collegare il quarto ingresso al pin  $V_{IN-}$  dell'INA, consentendo così di calcolare la tensione di offset ai capi  $V_{IN+}$  e  $V_{IN-}$ . Inoltre, si è optato per l'eliminazione della resistenza  $R_{16}$ , precedentemente inserita per tentare di modificare il guadagno di retroazione, ma senza risultati significativi.

### ***4.4 Differenze di Collegamento tra INA226 e INA229***

L'unica differenza di collegamento tra l'INA226 e l'INA229 risiede nel metodo di comunicazione utilizzato. Mentre l'INA226 comunica tramite I2C e richiede solo 2 pin, l'INA229 utilizza SPI e richiede 4 pin per la comunicazione.

Considerando che il nuovo INA comunica tramite SPI, inizialmente si era contemplata l'opzione di utilizzare lo stesso sistema di comunicazione anche per i DAC, poiché essi erano in grado di comunicare sia tramite SPI che I2C.

Tuttavia, dopo un'attenta valutazione, si è concluso che per ottenere una migliore acquisizione dei dati, sarebbe stato più opportuno comandare i DAC tramite I2C. Questa decisione è stata presa considerando che, utilizzando l'I2C per i DAC, sarebbe stato possibile ottenere un'acquisizione completa dei dati del transitorio dell'INA quando si decidesse di modificare la tensione di uscita. Questo perché la linea di comunicazione SPI sarebbe rimasta libera, il che non sarebbe stato possibile altrimenti.

#### ***4.5 Analisi dell'LDO***

Nell'analisi del regolatore lineare di tensione (LDO) TPS736 fatta precedentemente, si è sottolineato un problema significativo relativo alla sua limitazione nell'erogare una tensione al di sotto di 1,20V. Questa restrizione è derivata dalla struttura interna del componente, in cui l'uscita è connessa a un partitore di tensione interno che impone un valore minimo di tensione ai suoi capi (come si può notare nelle figure 4.2 e 4.3). Ciò costituisce una significativa limitazione operativa per un alimentatore elettronico a bassa potenza, poiché impedisce l'utilizzo delle tensioni inferiori a 1,20V.

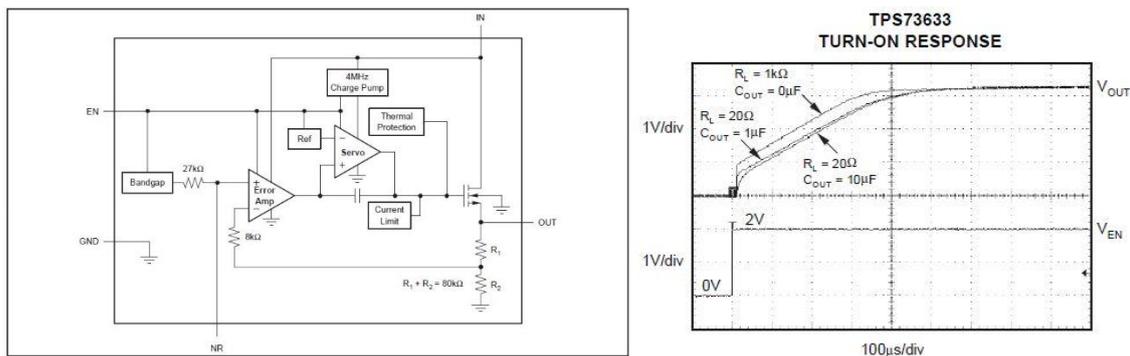
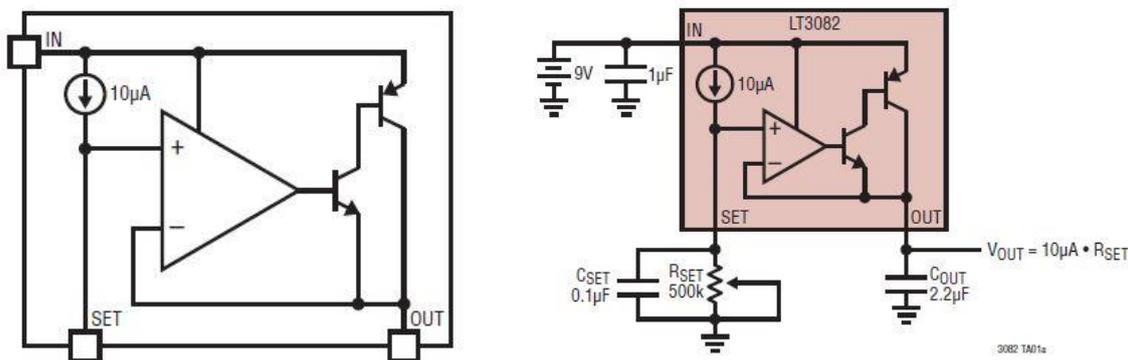


Figura 4.2 e 4.3 : schema circuitale interno TPS736 e risposta della tensione d'uscita nel tempo quando viene applicato un gradino di 2V.

Per affrontare questa problematica, si è considerata inizialmente l'opzione di inserire un secondo LDO, progettato per operare efficacemente a basse tensioni, e integrarlo nel sistema in modo da poterli utilizzare in maniera complementare. A tal fine, è stato scelto di impiegare il regolatore lineare LT3082, caratterizzato da una bassa caduta di potenziale e in grado di fornire un'ampia gamma di tensioni di uscita da 0V a 38,5V. Questo componente è costituito da un generatore di corrente di precisione da 10μA che, scorrendo su un resistore, controlla la tensione di uscita. È inoltre progettato per fornire una fonte di corrente precisa, risultando vantaggioso per applicazioni che richiedono elevate correnti e una minima dissipazione di calore.

L' LT3082 è dotato di tre pin: IN, OUT e SET (figura 4.4). Affinché il dispositivo funzioni correttamente, è necessario che la tensione sul pin IN sia maggiore di 1,2V - 1,4V rispetto a quella sul pin OUT. Inoltre, richiede una corrente di carico minima di 0,5mA per garantire la regolazione dell'uscita. Questo requisito può rappresentare un problema nel caso in cui si richieda un valore di tensione bassa sull'uscita, come nel contesto di utilizzo considerato per questo componente. Infatti, man mano che la tensione sulla resistenza in uscita diminuisce, diminuisce anche la corrente. Pertanto, si è reso necessario integrare al circuito un ulteriore componente, una pompa di carica, al fine di generare una tensione negativa sull'altro capo della resistenza, garantendo così il soddisfacimento del vincolo sulla corrente richiesta.

Il pin SET costituisce l'ingresso non invertente dell'amplificatore di errore e determina il punto di polarizzazione operativa del circuito, fornendo una corrente fissa di 10 $\mu$ A tramite un generatore di corrente. [12]



*Figura 4.4: schema circuitale interno LT3082 e un possibile schema circuitale di utilizzo del LT3082, utilizzato per testare il componente*

#### **4.5.1 Simulazione LDO**

Per ottenere una comprensione dettagliata del funzionamento del componente, oltre alla consultazione del datasheet, è stato deciso di condurre una simulazione utilizzando un software di progettazione grafica gratuito noto come . Questo software consente la creazione di progetti di circuiti e fornisce un oscilloscopio virtuale per l'analisi dei risultati. L'analisi ha confermato, come previsto dalle specifiche tecniche del datasheet, che il componente eroga in uscita una tensione pari a quella impostata sul pin SET, a condizione che siano rispettate tutte le condizioni indicate.

Nella figura 4.5 è possibile osservare il comportamento del componente con una tensione di ingresso  $V_{IN} = 9V$  in blu, una tensione  $V_{SET} = V_{OUT} = R_2 \cdot I_{SET} = 500k\Omega \cdot 10\mu A = 5V$  in verde, la corrente  $I_{SET} = I_{R2} = 10\mu A$  in fucsia e  $I_{OUT} = I_{R1} = \frac{V_{OUT}}{R_1} = \frac{5V}{10k\Omega} = 500\mu A$  in rosso.

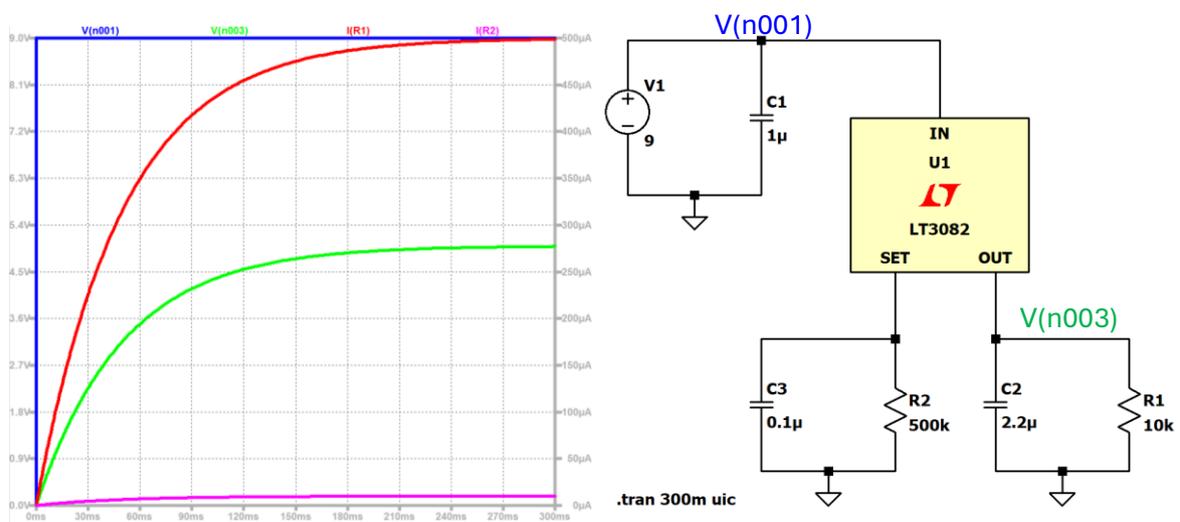
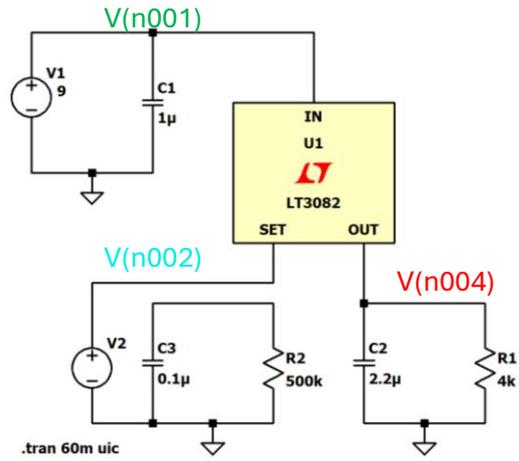
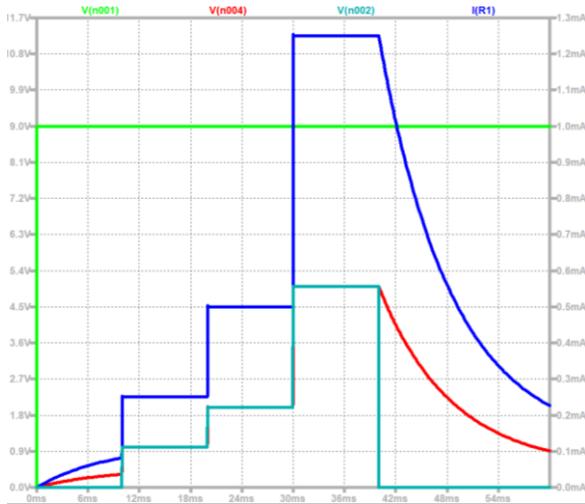
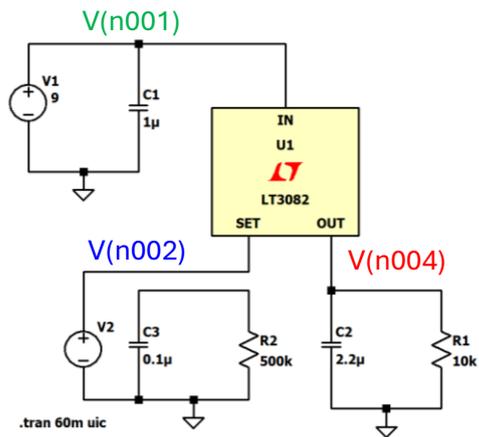
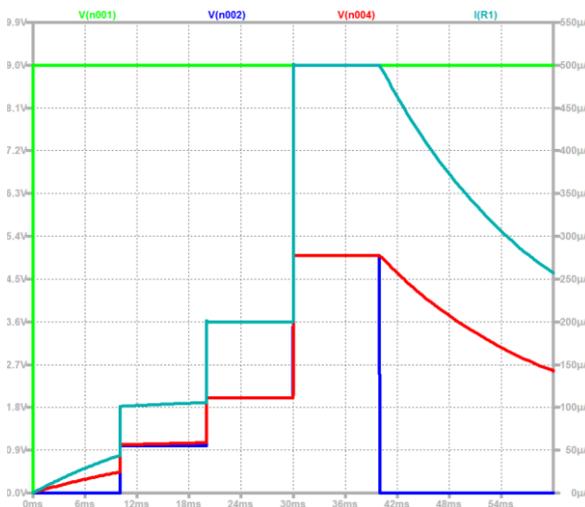


Figura 4.5: simulazione LT3082 per analizzare il transitorio tramite LTspice

Per studiare il comportamento dell'LT3082 al variare della tensione, è stato simulato il DAC tramite un generatore PWL, un generatore lineare a tratti che ci consente di generare dei gradini con diversi valori di tensione nel tempo (figura 4.6).



a)



b)

Figura 4.6: Simulazione LT3082 al variare della tensione

a)  $R1=4k\Omega$

b)  $R1=10k\Omega$

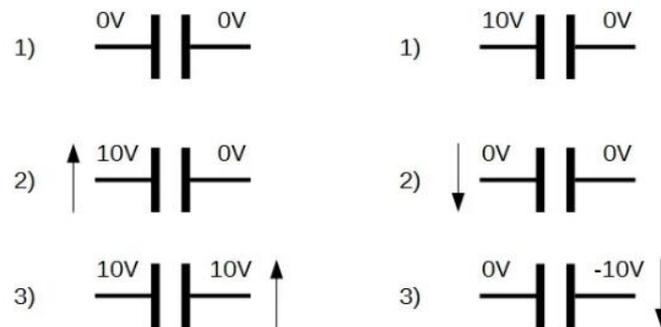
Considerando una tensione  $V_{SET} = 5V$ , il valore limite per il resistore  $R_1$  affinché scorra una corrente di  $0,5mA$ , necessaria per il funzionamento ideale del componente, è  $10k\Omega$ . È essenziale tenere presente che diminuendo la tensione, questa condizione non sarà più garantita. Pertanto, è fondamentale prestare attenzione al carico e calcolare la quantità di corrente che scorre attraverso di esso a diverse tensioni. Per risolvere questa problematica, è stata considerata l'opzione di connettere il terminale del resistore connesso a massa all'uscita di un ulteriore componente in grado di erogare una tensione negativa. Ciò ha lo scopo di aumentare la caduta di tensione sul resistore anche per tensioni di uscita basse del LDO, e di conseguenza aumentare la corrente che scorre su di esso, garantendo il corretto funzionamento del componente anche in condizioni non ideali.

## 5. POMPA DI CARICA

Per soddisfare le esigenze progettuali è stata selezionata una pompa di carica, comunemente costituita da un circuito elettronico che utilizza condensatori per immagazzinare energia al fine di generare tensioni negative.

### 5.1. *Funzionamento*

Questo componente sfrutta l'accoppiamento capacitivo, un fenomeno mediante il quale un condensatore tende a mantenere costante la differenza di tensione ai suoi terminali. L'idea alla base di questo fenomeno è che, se un lato del condensatore viene portato a una tensione diversa, per mantenere la stessa differenza di potenziale l'altro lato aumenterà o diminuirà la sua tensione. Ad esempio, se una piastra di un condensatore viene portata da 0V a 10V, l'altra piastra inizierà ad aumentare la tensione fino a raggiungere 10V. Il processo si verifica anche in senso contrario: se la tensione sulla prima piastra passa da 10V a 0V, l'altra piastra cercherà di abbassarsi di 10V per mantenere la tensione costante ai capi del condensatore. (figura 5.1)



*Figura 5.1: rappresentazione dell'accoppiamento capacitivo su un condensatore*

Per generare la tensione negativa desiderata, è necessario caricare una delle piastre del condensatore impedendo che l'altra raggiunga la stessa tensione. Successivamente, la prima piastra viene scaricata rapidamente, causando lo scaricamento anche della seconda piastra e generando così una tensione negativa. Per fare ciò vengono utilizzati in generale dei componenti come diodi o MOSFET, che fungono da interruttori. Questo processo rappresenta la sequenza operativa che sfrutta il componente per produrre la tensione negativa richiesta. [13]

## ***5.2. Analisi delle possibili pompe di carica e scelta del componente finale***

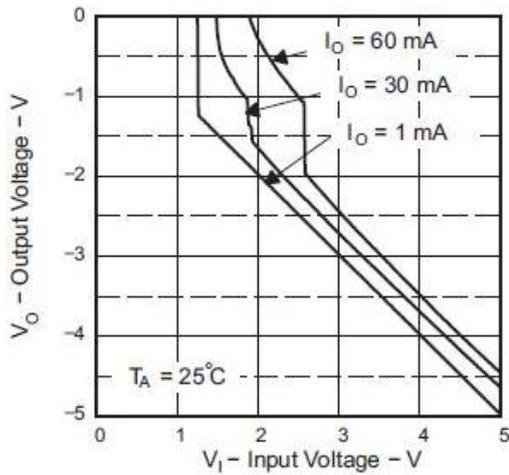
Si è intrapresa dunque una ricerca per individuare una pompa di carica con specifiche caratteristiche, tra le quali primeggia la capacità di produrre una tensione di uscita negativa, variabile entro un range compreso tra 0V e 5V, e la capacità di assorbire una corrente in uscita superiore a 0,5mA. Tra i potenziali componenti individuati sul mercato:

- Il TPS63710 rispecchiava i requisiti relativi alla tensione, tuttavia il suo intervallo di corrente in uscita andava da 0A a 1A. Per un funzionamento ottimale richiedeva un assorbimento di corrente superiore rispetto alle specifiche del progetto, inclusa un'elevata corrente di stand-by. Di conseguenza, è stato escluso poiché progettato per applicazioni ad elevata potenza non coerenti con quelle del nostro circuito. [14]
- Il LM7705 offre una tensione di uscita negativa fissa di  $-0,23V$  [15], risultando troppo bassa per le nostre esigenze. Per rispettare il vincolo sulla corrente, anche quando la tensione di uscita dell'LDO è pari a zero, si dovrebbe avere una resistenza pari a:

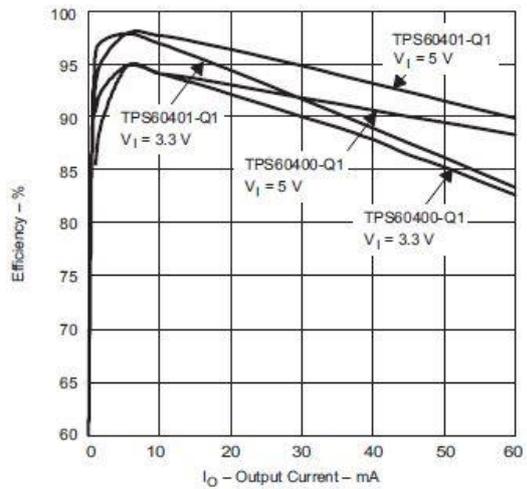
$$R \leq \frac{V_{OUTLDO} - V_{OUTPOMPA}}{I_{OUT}} = \frac{(0 + 0,23)}{(0,5 \cdot 10^{-3})} = 460\Omega$$

Realmente si dovrebbe considerare una resistenza inferiore a 460Ω; infatti, va notato che il valore della corrente in questione è un limite minimo. Perciò, per garantire una maggiore robustezza al circuito, si dovrebbe considerare una corrente più elevata attraverso il resistore e quindi una resistenza ancora più bassa. Questa scelta comporterebbe un consumo eccessivo quando la tensione su di essa aumenta, dovuto all'aumento dell'uscita dell'LDO e, quindi, della corrente. Questo dispositivo è progettato per fornire tensione negativa agli amplificatori operazionali a bassa tensione, una finalità differente dalla nostra.

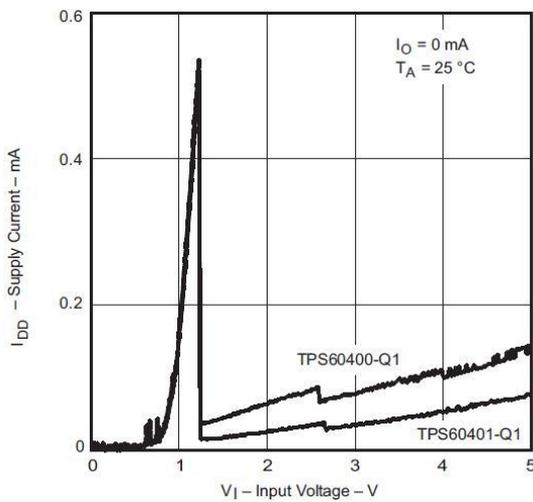
- Il LM27761 presenta caratteristiche interessanti, tuttavia, rispetto al TPS60401-Q1, presenta una corrente massima in uscita e assorbita in modalità stand-by quadruple, risultando eccessive per le nostre specifiche. [16]
- Il TPS60401-Q1, invece, genera una tensione di uscita negativa regolata da una tensione di ingresso compresa tra 1,8V e 5,25V. La dipendenza è rappresentata graficamente al variare delle correnti in uscita in figura 5.2.a, evidenziando un andamento lineare per determinati valori di tensione.



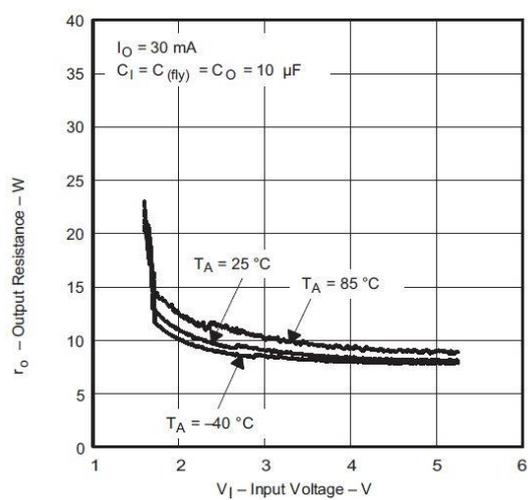
a)



b)



c)



d)

Figura 5.2: grafici della pompa di carica, rispettivamente:

a) grafico della tensione d'uscita in funzione della tensione d'ingresso al variare della corrente, a temperatura stabile  $T=25^{\circ}\text{C}$

*b) grafico dell'efficienza in funzione della corrente d'uscita, per diversi modelli della pompa di carica*

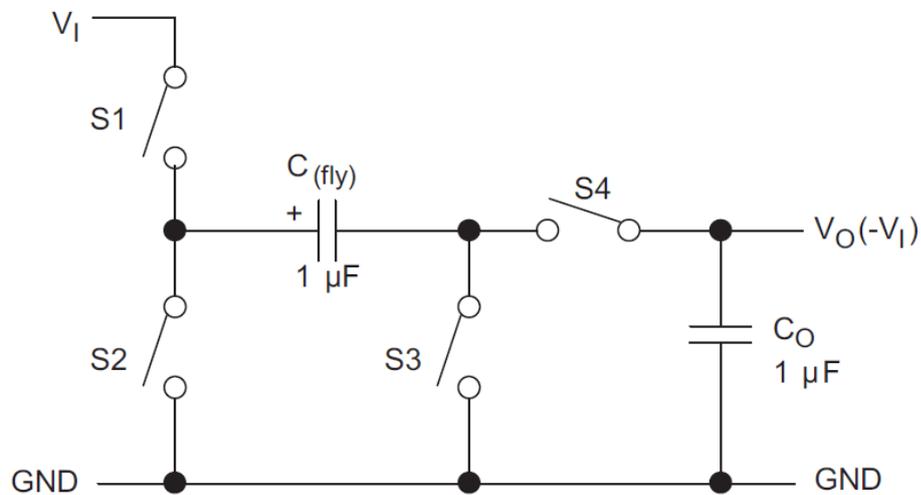
*c) grafico della corrente d'alimentazione in funzione della tensione d'ingresso*

*d) grafico della resistenza d'uscita in funzione della tensione in ingresso*

Il valore massimo della corrente in uscita è di 60mA, mentre il consumo di corrente in modalità stand-by è tipicamente di 100 $\mu$ A. Dal grafico in figura 5.2.b, si osserva che per il TPS60401, l'efficienza per un ingresso di 5V e correnti in uscita superiori a 0,5mA supera l'85%. Inoltre, la resistenza in uscita è di circa 9 $\Omega$ .

Il funzionamento di base del componente consiste nell'utilizzo di MOSFET come interruttori per caricare e scaricare i condensatori, illustrato nella figura 5.3. Durante la prima metà del ciclo, gli interruttori S2 e S4 sono aperti, mentre gli interruttori S1 e S3 sono chiusi, consentendo al condensatore ( $C_{FLY}$ ) di caricarsi alla tensione di ingresso. Durante la seconda metà del ciclo, gli interruttori S1 e S3 sono aperti e S2 e S4 sono chiusi, collegando il terminale positivo di  $C_{FLY}$  a GND e il

negativo a  $V_O$ . Collegando  $C_{FLY}$  in parallelo,  $C_O$  viene caricato negativamente. La tensione effettiva in uscita è leggermente superiore a  $-V_I$ , poiché gli interruttori S1-S4 presentano una resistenza parassita e  $C_O$  perde carica. [17]



*Figura 5.3: schema circuitale di base sul funzionamento della pompa di carica*

Attraverso la simulazione dell'LT3082 insieme a un generatore di tensione negativa che funge da pompa di carica, si è confermato che è una soluzione valida per rispettare il vincolo dell'LT3082 sulla corrente in uscita.

### 5.3. Simulazione

Attraverso una simulazione condotta sul componente LT3082, associato a un generatore di tensione negativa che opera come pompa di carica, è stato verificato che questo metodo costituisce una soluzione efficace per rispettare il vincolo sulla corrente in uscita imposto dal LT3082.

In figura 5.4 si può notare il risultato della simulazione, dove V1 in verde è l'alimentazione fornita dal booster, V2 in blu rappresenta il DAC che cambia valore d'uscita nel tempo, V3 è la pompa di carica, la tensione in rosso è quella in uscita e la corrente in celestino è quella sul resistore R1.

Notare come la corrente non scende mai al di sotto della soglia limite.

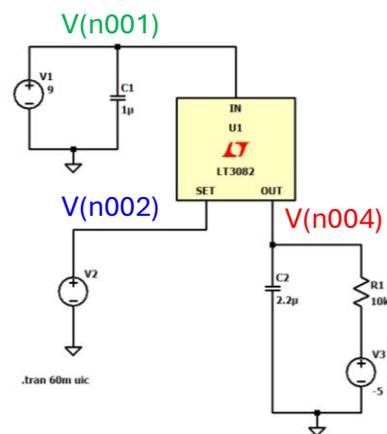
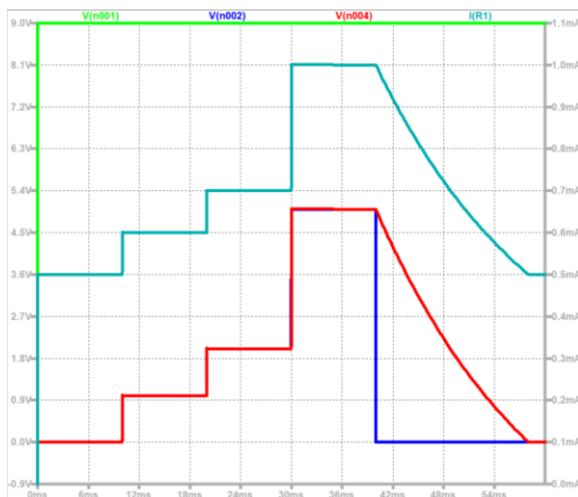
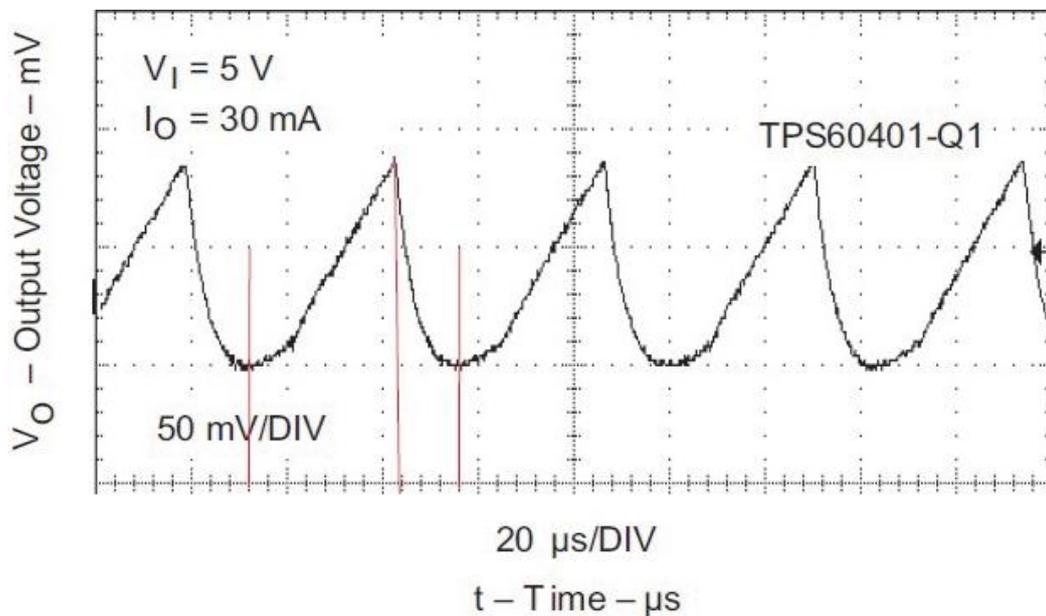


Figura 5.4: Simulazione LT3082 con pompa di carica

Nell'analisi del datasheet del componente [17], si è prestata particolare attenzione alla variazione temporale dell'output, considerando il possibile impatto di tale variazione, soprattutto se significativa. Come illustrato nel grafico fornito direttamente dal datasheet (figura 5.5), la variazione dell'uscita è di 80mV, con un periodo di 44 $\mu$ s, presentando una forma d'onda approssimativamente triangolare.



*Figura 5.5: grafico della variazione della tensione d'uscita in funzione del tempo, a tensione d'ingresso e corrente d'uscita costante risp. 5V e 30mA*

Al fine di mitigare tali variazioni, è stato progettato un filtro da collegare all'uscita della pompa di carica mediante l'utilizzo di un simulatore.

Inizialmente, è stata simulata l'onda da filtrare, deducendo i parametri dal grafico fornito, ovvero un periodo di  $44\mu\text{s}$ , un semiperiodo di salita di  $31,2\mu\text{s}$  e un semiperiodo di discesa di  $12,8\mu\text{s}$ , con un'ampiezza di  $80\text{mV}$ .

Successivamente, questi parametri sono stati inseriti nel generatore di forme d'onda, selezionando la funzione d'onda appropriata (figura 5.6).

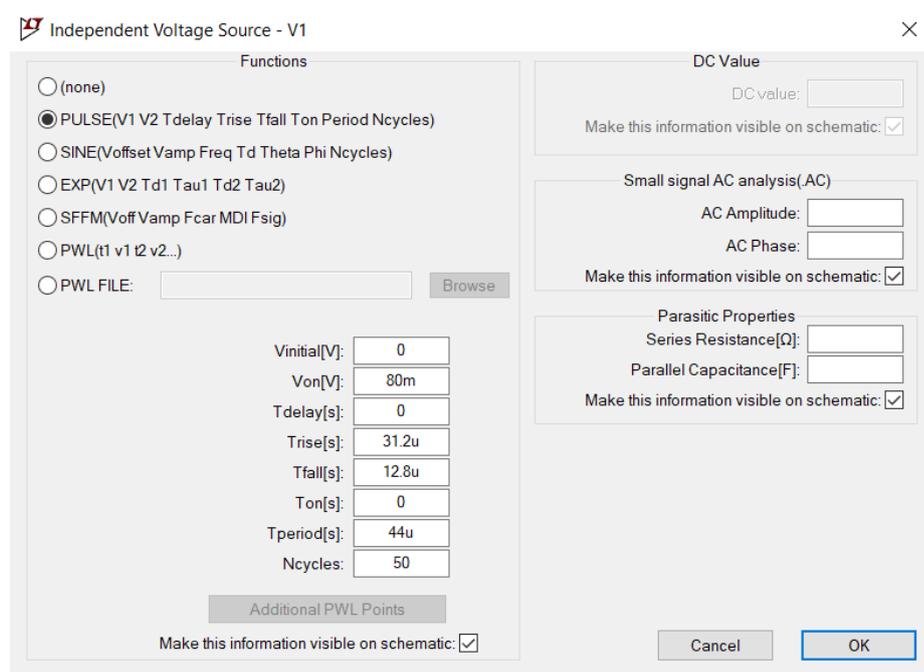
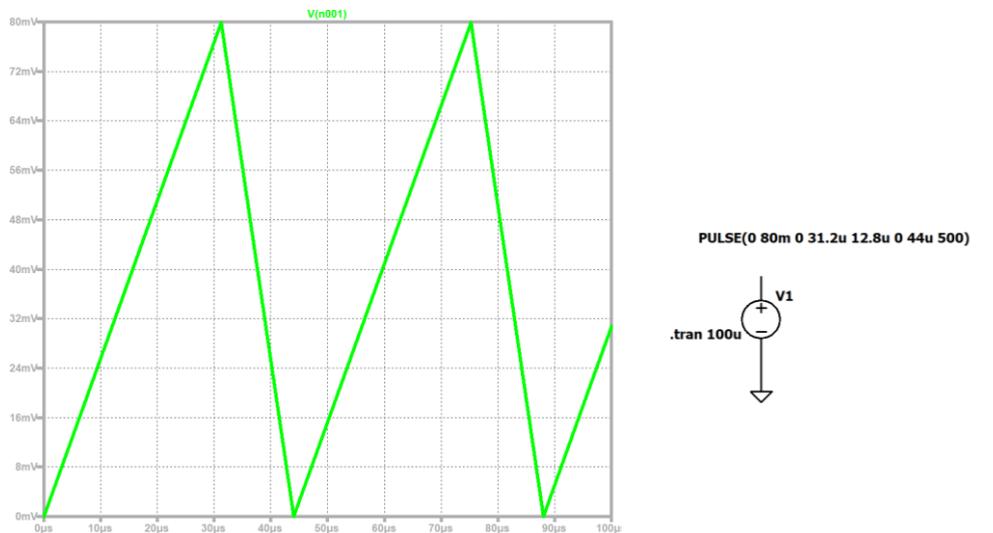


Figura 5.6: Valori per simulare l'onda da filtrare usando LTspice

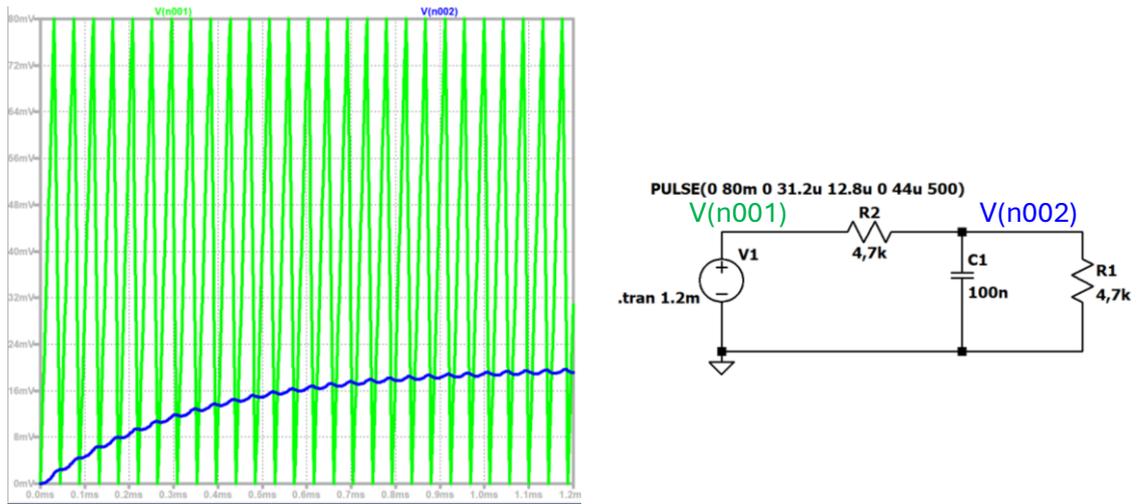
Il risultato è stato visualizzato sull'oscilloscopio di LTspice (figura 5.7), producendo un'onda molto simile a quella reale.



*Figura 5.7: onda che simula la variazione di tensione all'uscita della pompa di carica*

Successivamente, è stato dimensionato un filtro composto da due resistenze e un condensatore in modo da filtrare in modo efficace il segnale, considerando che il filtro sarà collegato anche all'uscita dell'LT3082. Di conseguenza, la somma dei valori delle resistenze non deve essere inferiore a  $10\text{k}\Omega$  per rispettare le specifiche sulla corrente.

Sono stati scelti i valori di  $R_1 = R_2 = R = 4,7\text{k}\Omega$  per le resistenze e  $C_1 = 100\text{nF}$  per il condensatore, ottenendo un efficace filtraggio, come dimostrato nel grafico in figura 5.8.



*Figura 5.8: Simulazione del filtraggio della variazione di tensione all'uscita della pompa di carica*

## 6. REVISIONE DEL PROGETTO E DELLE MODIFICHE APPORTATE

In seguito alla valutazione delle specifiche e delle esigenze del progetto, sono state implementate le modifiche sul precedente progetto, il quale era stato realizzato utilizzando il software di progettazione di schemi elettrici e circuiti stampati denominato KiCad.

Le modifiche apportate comprendono la sostituzione dei seguenti componenti:

- L'INA226 è stato sostituito con l'INA229;
- Il DAC MAX5805 è stato sostituito con il DAC80501;
- Il TLP3107 è stato sostituito con il TLP3107A;
- È stato inserito l'LT3082;
- È stato aggiunto un ulteriore DAC80501 per comandare l'LT3082;
- È stata aggiunta una pompa di carica con filtro per migliorare la qualità e la stabilità dell'alimentazione dell'LT3082;
- È stato incluso un connettore per il Raspberry Pico;
- È stata eliminata la resistenza  $R_{16}$  di retroazione collegata tra INA226 e TPS736;

- Infine, è stato realizzato il collegamento del pin  $V_{IN-}$  dell'ina al pin S1 del TMUX.

In figura 6.1 si può vedere una prima revisione e modifica del circuito precedente.

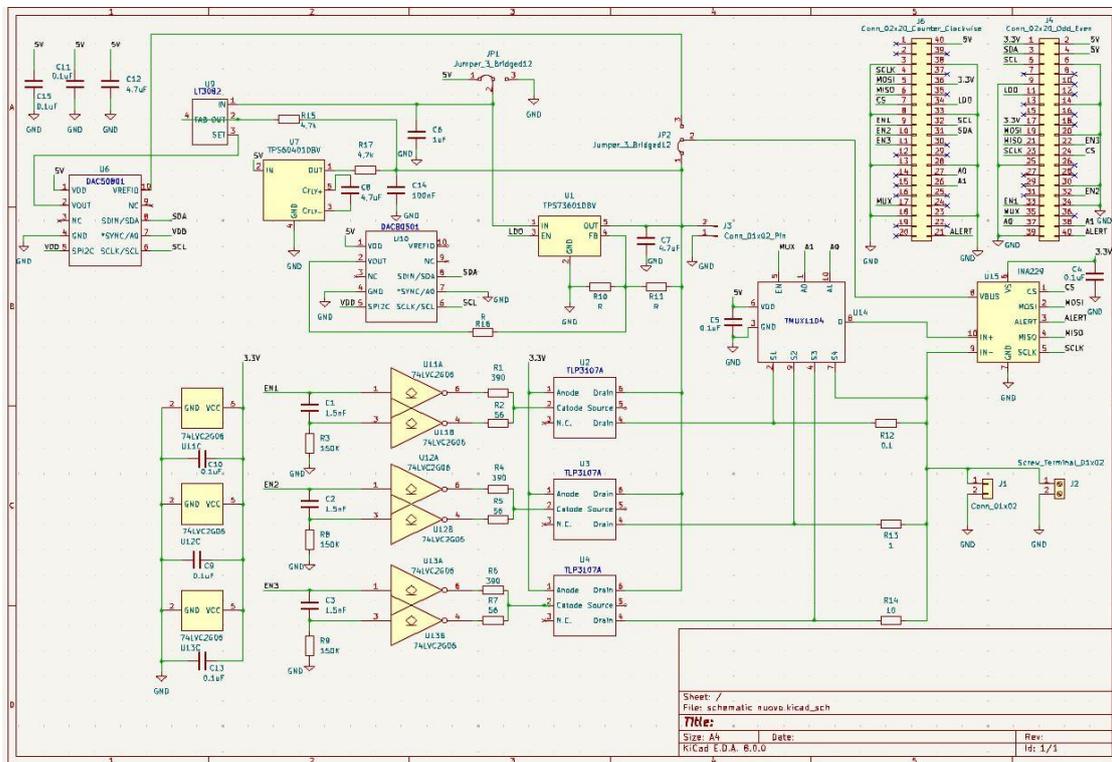


Figura 6.1: Prima revisione e modifica del circuito precedente

## 7. PROBLEMA CON L'UTILIZZO DEI DUE LDO INSIEME

Al fine di verificare l'interazione tra i dispositivi DAC80501 e LT3082, è stato condotto un test con i componenti. Durante il test, si è osservato un comportamento inatteso: impostando la tensione d'uscita del DAC a 4V e successivamente riportandola a 0V, l'uscita dell'LT3082 è salita a 4,7V.

Questo fenomeno si è ripetuto in modo costante con tensioni d'uscita elevate, manifestando un comportamento di saturazione del regolatore anziché il suo normale funzionamento.

Sebbene il datasheet del LT3082 [12] indichi chiaramente una protezione per tensioni d'uscita superiori a  $V_{SET}$  e  $V_{IN}$ , in realtà questa protezione sembra proteggere solo il dispositivo. Invece di spegnersi per proteggere il carico, il regolatore continua ad erogare tensione, raggiungendo il massimo possibile. È emerso che ogni volta che si esce dalla regione di funzionamento specificata (cioè, quando  $V_{IN} > V_{OUT} + 1,65V$  e  $I_{OUT} > 0,5mA$ ), il dispositivo entra in uno stato di "blocco" con

$V_{OUT} = V_{IN} - 0,3V$ . È probabile che questo problema si manifesti anche con  $V_{SET} \ll V_{OUT}$ , ma non è stato possibile verificarlo con certezza. Tuttavia, se la differenza supera 0,6V, il dispositivo entra in blocco.

Questo scenario va in contrasto l'idea iniziale di connettere in parallelo le uscite dell'LT3082 e del TPS736 per un funzionamento complementare perché non si può tollerare che il regolatore vada in blocco e continui a erogare una tensione di circa 4,8V in uscita, indipendentemente dalla tensione sul pin SET fornita dal DAC.

Quindi si è deciso di abbandonare il vecchio LDO e utilizzare solo l'LT3082.

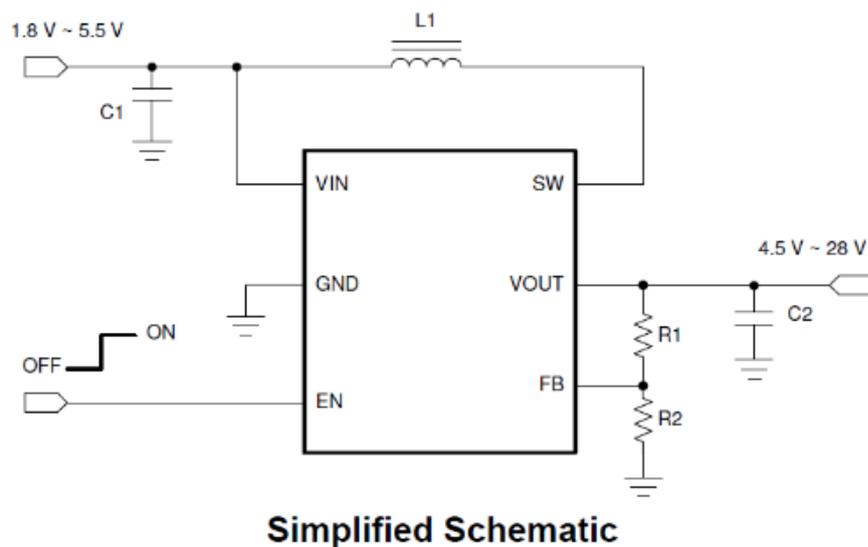
Questa soluzione non è stata subito adottata poiché per ottenere una tensione di 5V in uscita dall'LT3082, con un margine sufficiente per evitare il blocco, sarebbe stato necessario alimentarlo con una tensione di almeno 6,5V - 7V, a causa dei transistor tra IN e OUT. Dal momento che la Raspberry non dispone di un connettore adatto per fornire queste tensioni, implementare questa soluzione avrebbe richiesto l'aggiunta di ulteriori componenti e complessità al sistema.

### ***7.1. Soluzione: Utilizzo di un Booster***

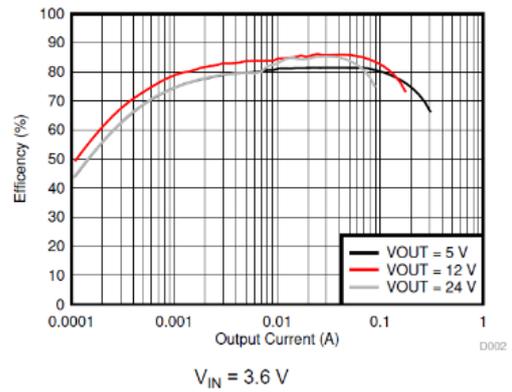
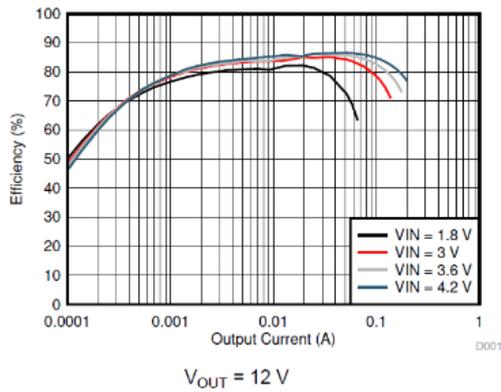
Per risolvere questo problema, è stata scelta l'opzione di utilizzare un booster (o convertitore step-up), precisamente il TLV61046A. Questa scelta è stata

motivata dalla sua facilità d'uso, pur presentando un'efficienza dell'80% (figura 7.1.b) che, sebbene non ottimale per le nostre esigenze, è stata ritenuta accettabile. Tuttavia, è importante sottolineare che questa efficienza è influenzata da un diodo interno che causa una caduta di potenziale ai suoi capi di 0,6V. Sebbene questa caduta di tensione possa sembrare trascurabile, soprattutto in condizioni di tensioni elevate, rappresenta una percentuale significativa delle basse tensioni coinvolte nel nostro contesto operativo.

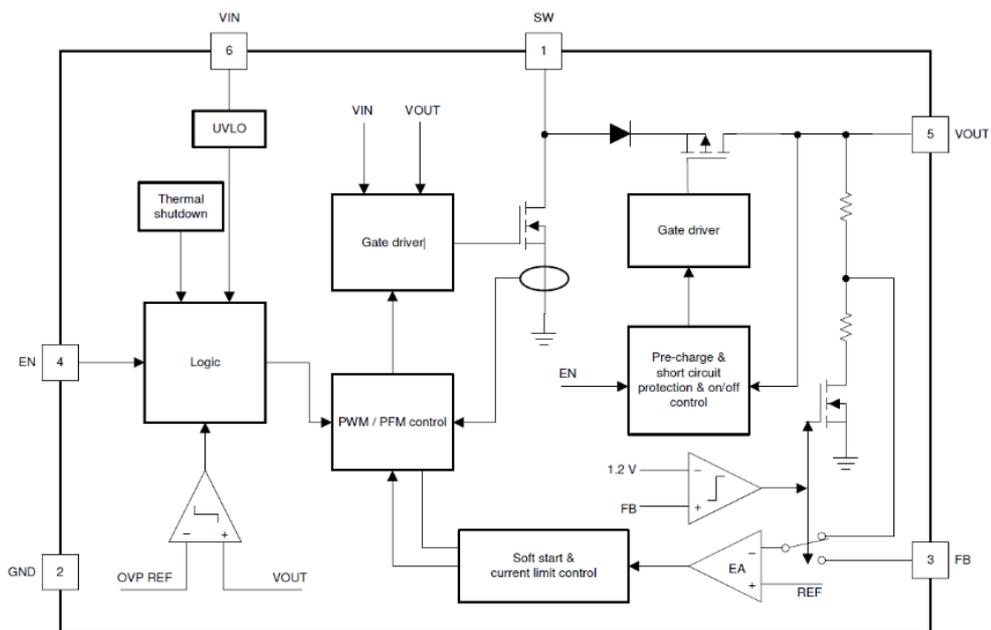
È da notare che, sebbene siano disponibili alternative con caratteristiche migliori sotto questo aspetto, tali opzioni presentavano un package che non rispettava le specifiche del nostro progetto, rendendo il TLV61046A la scelta più praticabile tra le opzioni disponibili.



a)



b)



c)

Figura 7.1: a) schema semplificato dell'utilizzo del booster

b) grafici dell'efficienza al variare della corrente d'uscita per diverse tensioni d'ingresso e d'uscita

c) schema circuitale a blocchi del TLV61046A

Il TLV61046A opera attraverso un meccanismo di modulazione della larghezza di impulso a frequenza quasi costante da 1,0MHz (PWM) con una corrente di carico che varia da moderata a pesante. Un circuito interno regola il periodo di inattività in funzione del rapporto tra la tensione di ingresso e quella di uscita. Durante il ciclo di commutazione, il transistor NMOS di commutazione, illustrato nello schema a blocchi in Figura 7.1.c, viene acceso, permettendo all'induttore di aumentare la sua corrente mentre il condensatore di uscita si scarica (il circuito è rappresentato in figura 7.1.a). Quando la corrente dell'induttore raggiunge la soglia impostata dall'amplificatore di errore, l'interruttore PWM si spegne e il diodo di potenza viene polarizzato direttamente. L'energia immagazzinata nell'induttore viene quindi trasferita al condensatore di uscita per alimentare il carico. Questo processo si ripete ciclicamente a intervalli regolari. L'amplificatore di errore confronta la tensione del pin FB con una tensione di riferimento interna e la sua uscita determina la corrente di picco dell'induttore.

Inoltre, il TLV61046A può operare in modalità di risparmio energetico utilizzando la modulazione della frequenza degli impulsi (PFM) per migliorare l'efficienza nei casi di carichi leggeri. Quando la corrente di carico diminuisce, la corrente di picco dell'induttore diminuisce, regolando di

conseguenza la tensione di uscita. Se la corrente di picco dell'induttore scende al di sotto di un limite di 200mA, la tensione di uscita supera quella impostata man mano che la corrente di carico diminuisce ulteriormente. Quando la tensione di feedback raggiunge una tensione di riferimento specifica per la modalità PFM, il TLV61046A passa in modalità di risparmio energetico. Durante questa modalità, il dispositivo continua a commutare per alcuni cicli a causa del ritardo interno del comparatore, dopodiché interrompe la commutazione, alimentando il carico direttamente dal condensatore di uscita. Quando la tensione di feedback scende al di sotto della tensione di riferimento PFM, il dispositivo riprende la commutazione per aumentare la tensione di uscita. [18]

Poiché intendiamo utilizzare principalmente correnti basse, il componente funzionerà principalmente in modalità di risparmio energetico. Pertanto, è fondamentale dimensionare con precisione i condensatori, gli induttori e i resistori collegati al TLV61046A utilizzando le formule e le specifiche fornite nel datasheet.

## **7.2. Dimensionamento del TLV61046A**

### **7.2.1. Dimensionamento dei resistori**

Per impostare la tensione di uscita del TLV61046A, vi sono due approcci distinti. Il primo prevede il collegamento del pin FB direttamente alla tensione di ingresso, stabilendo così una tensione di uscita fissa di 12V. Questa configurazione richiede solo tre componenti esterni, riducendo al minimo le dimensioni della soluzione, tuttavia, non rispecchia le nostre specifiche. Il secondo metodo implica l'utilizzo di un partitore resistivo esterno, costituito da due resistenze,  $R_1$  e  $R_2$ , per impostare la tensione di uscita desiderata.

Utilizzando il partitore resistivo esterno, la tensione di uscita può essere programmata sul valore desiderato mediante la seguente equazione:

$$R_1 = \left( \frac{V_{OUT}}{V_{REF}} - 1 \right) \cdot R_2$$

Dove:

-  $V_{OUT}$  rappresenta la tensione di uscita desiderata;

-  $V_{REF}$  è la tensione di riferimento interna al pin FB.

Quando la tensione di uscita è regolata, la tensione tipica sul pin FB è  $V_{REF}$  di 795mV.

Per garantire la massima precisione, è consigliabile mantenere il valore di  $R_2$  inferiore a 80k $\Omega$ , in modo che la corrente che scorre attraverso  $R_2$  sia almeno 100 volte maggiore della corrente di dispersione del pin FB. La scelta di  $R_2$  influisce sull'immunità al rumore, con valori inferiori che offrono una maggiore immunità. D'altra parte, valori più elevati di  $R_2$  riducono la corrente di riposo, migliorando l'efficienza a carichi bassi.

Nel nostro caso, desideriamo una tensione di uscita di 7V per alimentare l'LT3082 e ottenere 5V sull'uscita dell'LDO. Utilizzando  $V_{OUT} = 7V$  e  $V_{REF} = 0,8V$ , possiamo calcolare il valore di  $R_1$  in funzione di diversi valori di  $R_2$  disponibili sul mercato.

Si è scelto di utilizzare valori standard di resistori, come definito dalla norma IEC 60063, con una tolleranza dell'1%. Utilizzando la serie E96 per i valori standard, sono stati selezionati diversi valori di  $R_2$  disponibili sul mercato e calcolati i corrispondenti valori di  $R_1$  per ognuno di essi.

Tramite essa si sono scelti come valori  $R_2 = 40,2k\Omega$ ,  $R_2 = 71,5k\Omega$  e  $R_2 = 78,7k\Omega$  e si è calcolata la resistenza  $R_1$  rispettivamente per tutti e tre i valori.

$R_2 = 40,2\text{k}\Omega$     $R_1 = 311,55\text{k}\Omega$    valori della serie E96 più vicini sono  $309\text{k}\Omega$   
e  $316\text{k}\Omega$

$R_2 = 71,5\text{k}\Omega$     $R_1 = 554,125\text{k}\Omega$    valori della serie E96 più vicini sono  
 $549\text{k}\Omega$  e  $562\text{k}\Omega$

$R_2 = 78,7\text{k}\Omega$     $R_1 = 609,925\text{k}\Omega$    valori della serie E96 più vicini sono  
 $604\text{k}\Omega$  e  $619\text{k}\Omega$

Sulla base di vari parametri, come il rapporto tra  $R_1$  e  $R_2$  per vedere quanto si  
discostano da

$\frac{V_{OUT}}{V_{REF}} - 1 = 7,75\text{V}$  e la distanza tra il valore calcolato e il valore in

commercio, si è optato per

$R_2 = 40,2\text{k}\Omega$  e  $R_1 = 311,55\text{k}\Omega$  in quanto  $R_2$  è il valore più basso, garantendo  
una maggiore precisione.

### ***7.2.2. Selezione dell'induttore***

La selezione dell'induttore riveste un ruolo fondamentale nella progettazione  
del regolatore di potenza, influenzando il funzionamento in regime

stazionario, il comportamento transitorio e la stabilità del circuito. L'induttore è il componente più importante nella progettazione del regolatore di potenza e sono tre le specifiche principali che guidano sulla sua scelta: il suo valore, la corrente di saturazione e la resistenza CC (DCR).

Il TLV61046A è progettato per operare con valori dell'induttore compresi tra 2,2µH e 22µH. Per calcolare la corrente di picco dell'induttore, che è cruciale per l'applicazione, è necessario considerare la tensione di ingresso minima, la tensione di uscita massima e la corrente di carico massima, mantenendo un margine di progettazione sufficiente e scegliendo un induttore con tolleranza del 3% e bassa efficienza di conversione di potenza.

Nel caso di un regolatore boost, la corrente continua dell'induttore può essere calcolata utilizzando l'equazione:

$$I_{LDC} = \frac{(V_{OUT} \cdot I_{OUT})}{(V_{IN} \cdot \eta)}$$

Dove:

- $V_{OUT}$  è la tensione di uscita;
- $I_{OUT}$  è la corrente in uscita;
- $V_{IN}$  è la tensione di ingresso;

-  $\eta$  è l'efficienza di conversione di potenza.

Utilizzando i valori noti  $V_{OUT} = 7V$ ,  $I_{OUT} = 200mA$  (massima corrente richiesta dall'LT3082),

$V_{IN} = (4,5 - 5,3)V$  (valore della tensione erogata dal connettore della Raspberry) e  $\eta = 80\%$ , si ottengono:

- per  $V_{IN} = 4,5V$   $I_{LDC} = 388,88mA$

- per  $V_{IN} = 5V$   $I_{LDC} = 350mA$

- per  $V_{IN} = 5,3V$   $I_{LDC} = 330,18mA$

Si decide di utilizzare  $I_{LDC} = 400mA$ , il doppio della corrente desiderata in uscita.

La corrente di ripple dell'induttore può essere calcolata con l'equazione:

$$\Delta I_{L(P-P)} = \frac{V_{IN} \cdot V_{OUT} + 0,8V - V_{IN}}{L \cdot f_{SW} \cdot (V_{OUT} + 0,8V)}$$

Dove:

-  $\Delta I_{L(P-P)}$  è la corrente di ripple dell'induttore;

- L è il valore dell'induttore;
- fsw è la frequenza di switching;
- V<sub>OUT</sub> è la tensione di uscita;
- V<sub>IN</sub> è la tensione di ingresso.

La corrente di picco dell'induttore viene quindi calcolata utilizzando l'equazione:

$$I_{LP} = I_{LDC} + \frac{\Delta I_{L(P-P)}}{2}$$

Per massimizzare la corrente di uscita, è consigliabile mantenere la corrente di picco-picco dell'induttore inferiore al 40% della corrente media dell'induttore.

Calcolando  $\Delta I_{L(P-P)}$  e sostituendolo nell'equazione precedente con i valori noti di

$V_{OUT} = 7V$ ,  $V_{IN} = 5V$  e  $f_{sw} = 1050kHz$ , si ottiene  $L = 11,21\mu H$ . Alla luce di questa valutazione, si è scelto l'induttore LPS4018-223MRC, disponibile sul mercato.

### 7.2.3. Selezione dei condensatori di ingresso e uscita

La scelta dei condensatori di ingresso e uscita è cruciale per garantire il soddisfacimento dei requisiti di ripple di uscita e la stabilità del circuito.

La capacità minima richiesta per un determinato ripple può essere calcolata utilizzando la seguente formula:

$$C_{OUT} = \frac{I_{OUT} \cdot D_{MAX}}{f_{SW} \cdot V_{RIPPLE}}$$

Dove:

- $D_{MAX}$  rappresenta il duty cycle di commutazione massimo;
- $V_{RIPPLE}$  è la tensione di uscita da picco a picco.

Per  $V_{RIPPLE}$ , considerando un'uscita non inferiore a 6,6V, si è scelto il valore di 0,6V. Con  $D_{MAX} = 0,36$ ,  $I_{OUT} = 200\text{mA}$  e  $f_{sw} = 1050\text{kHz}$ , si calcola  $C_{OUT} = 120\text{nF}$ . Questo valore soddisfa i requisiti e attenua il ripple sulla tensione; tuttavia, scegliendo una capacità più elevata si otterrà un'attenuazione maggiore. Per questo motivo si è optato per  $C_{OUT} = 4,7\mu\text{F}$  anche considerando che le specifiche suggeriscono l'utilizzo di un condensatore di uscita con una capacità effettiva compresa tra  $0,47\mu\text{F}$  e  $10\mu\text{F}$ .

È importante notare che il condensatore di uscita ha un impatto significativo sulla stabilità del circuito del regolatore boost: se la capacità del condensatore di uscita fosse inferiore a tale intervallo, il regolatore boost potrebbe diventare instabile.

Per quanto riguarda il condensatore di ingresso, per la maggior parte delle applicazioni il produttore raccomanda l'utilizzo di un condensatore ceramico da  $1,0\mu\text{F}$ .

Una regola empirica suggerisce di selezionare i valori dei condensatori in modo che quello di ingresso sia maggiore di quello di uscita, specialmente quando ingresso e uscita sono connessi. Questo facilita il caricamento del condensatore di uscita all'accensione del dispositivo e previene eventuali problemi causati da uno scaricamento anomalo del condensatore di ingresso.

## **8. SELEZIONE DEI CONDENSATORI PER OGNI COMPONENTE**

Per garantire un'alimentazione stabile e mitigare gli spike di corrente nei circuiti digitali, è stata inclusa una selezione appropriata di condensatori sulla linea di alimentazione di ciascun componente. Gli spike di corrente possono causare un calo di tensione dovuto all'effetto induttivo dei fili, e l'utilizzo di condensatori è un metodo efficace per risolvere questa problematica, in quanto sono in grado di immagazzinare energia e rilasciarla gradualmente nel tempo.

Nella progettazione dei circuiti digitali, è comune utilizzare condensatori con una capacità di  $0,1\mu\text{F}$ . Questi condensatori contribuiscono a stabilizzare l'alimentazione e a mantenere la tensione sui componenti digitali entro i limiti accettabili, garantendo così il corretto funzionamento del circuito.

## 9. OTTIMIZZAZIONI FINALI

Prima di procedere alla fase di realizzazione del circuito stampato, sono state apportate alcune ottimizzazioni e aggiunte per migliorare il funzionamento complessivo del sistema.

- Inizialmente, è stato eliminato il connettore J2, poiché l'idea originale di utilizzarlo per misurare la tensione in uscita dell'LDO si è rivelata di scarso valore pratico nell'utilizzo effettivo del circuito. Al suo posto, è stato deciso di sfruttare il pin  $V_{REF}$  del DAC80501, il quale fornisce una tensione di riferimento precisa di 2,5V. Questa tensione è stata utilizzata per calibrare l'INA229, collegandolo tramite un jumper al pin  $V_{BUS}$ . È stato inserito un jumper a 6 ingressi per consentire la sperimentazione di differenti collegamenti di  $V_{bus}$ , per prelevare la tensione direttamente dall'uscita sul pin  $V_{IN-}$ , sfruttando il fatto che l'INA229 ha un basso assorbimento di corrente sul pin  $V_{BUS}$  (10 nA), un valore trascurabile ai fini del funzionamento del circuito.
- Successivamente, è stato aggiunto un connettore per collegare il circuito alla Raspberry Pico, al fine di garantire una comunicazione più veloce e

una maggiore facilità di utilizzo. Durante questa operazione, è stata posta particolare attenzione al collegamento degli enable dei componenti ai GPIO della Raspberry. È importante notare che i GPIO da 0 a 8 sono attivi alti di default e i componenti utilizzati vengono abilitati solo quando ricevono un segnale di tensione alto sul pin enable. Pertanto, è stato evitato di collegare quest'ultimi ai GPIO attivi di default alti per prevenire l'accensione inavvertita del circuito all'avvio della Raspberry. È stato deciso di collegarli a GPIO di default bassi per garantire che i componenti siano disabilitati all'accensione della scheda e si accendano solo quando il firmware prenderà il controllo dei GPIO e manderà un livello di tensione alto.

Inoltre, per bilanciare il carico di corrente dall'alimentazione della Raspberry Pi, sono stati utilizzati entrambi i pin che erogano una tensione di 5V per evitare di sovraccaricare un unico pin.

- Per quanto riguarda l'ottimizzazione dell'accensione della pompa di carica, si è deciso di inserire un componente che consentisse di spegnerla e accenderla a piacimento. Questo è stato fatto per evitare possibili

tensioni negative in uscita nel caso in cui l'LDO venisse spento mentre la pompa rimanesse accesa.

Inizialmente, si è pensato di utilizzare un transistor PMOS all'ingresso della pompa di carica. Tuttavia, poiché la pompa è alimentata a 5V e il PMOS richiede un segnale di controllo di 3,3V, con una tensione di soglia di -1,7V, si è riscontrata difficoltà nel trovare in commercio PMOS con questo livello di tensione. Anche se è stato trovato un componente con una tensione di soglia di -1,9V, che sembrava adatto, c'era il rischio che la differenza di tensione potesse causare problemi, dato che la tensione sui GPIO non è estremamente precisa. Per questo motivo, si è scelto di utilizzare un componente più avanzato che incorporasse sia il transistor che circuiti di abilitazione, protezione e segnalazione di eventuali problemi. Questo assicura una maggiore stabilità complessiva del circuito. Il componente scelto è il TPS2001D, progettato per gestire carichi capacitivi pesanti e cortocircuiti, come nei casi di utilizzo USB, ma che rispecchia perfettamente le nostre specifiche.

Per applicazioni USB il componente richiede un condensatore in uscita di 150 $\mu$ F, perché serve per dare 900mA su un cavo USB che può essere lungo metri, mentre nel nostro progetto, considerando le basse correnti

richieste è stato utilizzato un condensatore da  $4,7\mu\text{F}$  nell'uscita dell'interruttore e all'ingresso della pompa.

Inoltre, è stato inserito un condensatore da  $0,1\mu\text{F}$  in ingresso, come consigliato dai produttori. Sebbene venga consigliato di inserire un resistore tra l'ingresso e il pin FLT per segnalare condizioni di sovracorrente o sovratemperatura, data la bassa corrente di lavoro, questa precauzione è stata ritenuta superflua, considerando che la Raspberry Pi non potrà mai erogare una corrente superiore a  $2\text{A}$ , che è il limite del componente. [19]

- Infine, è stata condotta una ricerca per sostituire i resistori di precisione in uscita con resistori dal valore doppio, mantenendo comunque una precisione del  $0,1\%$ , per migliorare la caduta di tensione. Tuttavia, questa ricerca non ha avuto successo.

Infine, tutti i cambiamenti sono stati riportati sul programma, conducendo alla realizzazione dello schema definitivo, che è mostrato nella figura 9.1.

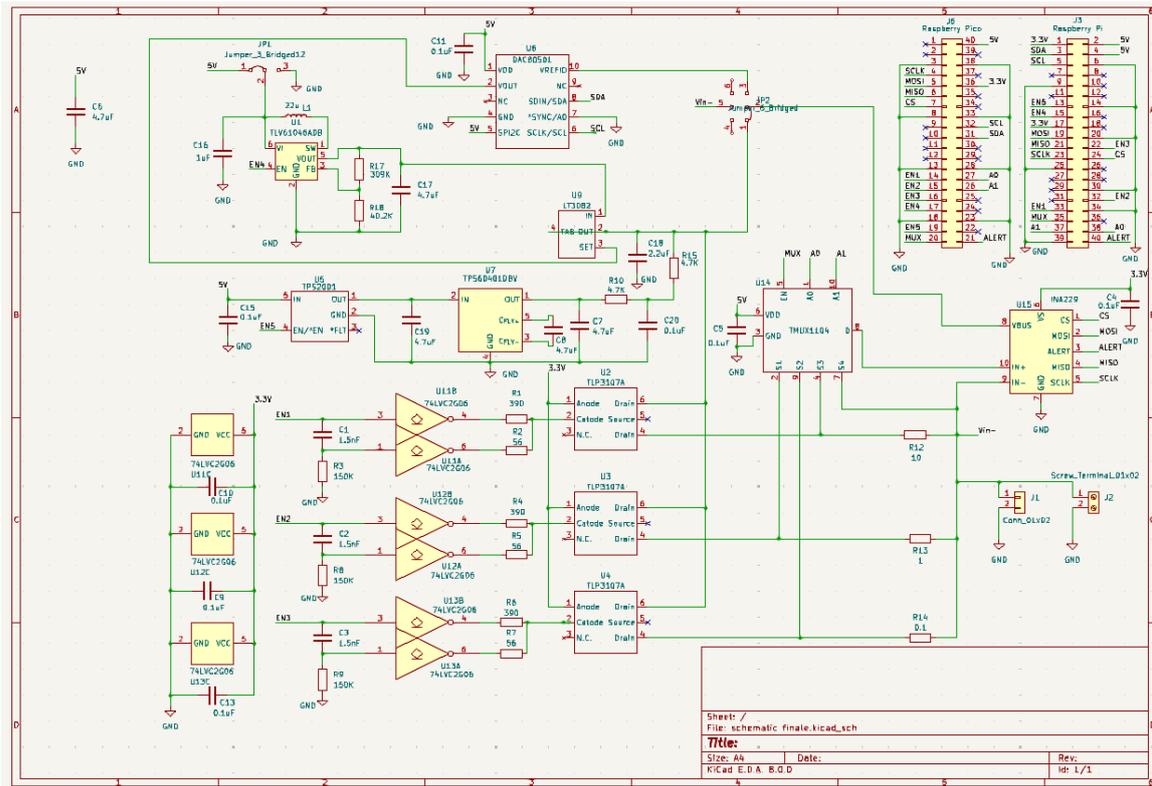


Figura 9.1: Schema circuitale finale del progetto

## 10. SBROGLIO DEL PCB: CARATTERISTICHE E ACCORTEZZE

Una volta conclusa la fase di progettazione, si è proceduto con la realizzazione del circuito stampato (PCB). Inizialmente, è stato assegnato a ciascun componente il relativo "footprint", ovvero l'impronta che identifica il componente elettronico sul PCB.

Successivamente, è stato avviato il processo di sbroglio del circuito, una fase delicata e complessa. Durante questa operazione, ci si è basati sul PCB precedente per mantenere invariata la parte del circuito che non richiedeva modifiche. È consigliabile posizionare i connettori ai lati del PCB per agevolarne l'accessibilità e nel nostro caso lasciare spazio sufficiente per l'LDO, considerando la sua tendenza a dissipare una considerevole quantità di potenza e, di conseguenza, a generare calore.

Per quanto riguarda i connettori destinati alla Raspberry Pi, si è deciso di includere soltanto quello per la Raspberry Pi, optando per l'utilizzo di un adattatore esterno per collegare la Raspberry Pi Pico.

Il circuito stampato realizzato è a doppio strato, caratterizzato quindi da due strati di rame che consentono la creazione delle piste di collegamento per i componenti. Una prassi efficace è destinare uno degli strati interamente come

piano di massa, contribuendo così a garantire una migliore distribuzione delle correnti nel circuito.

### ***10.1. Miglioramento e ottimizzazione del PCB***

Al fine di migliorare l'usabilità e ottimizzare il circuito stampato (PCB), si è deciso di aumentarne le dimensioni, prendendo in considerazione le dimensioni standard di un HAT per la Raspberry Pi. Questa scelta è stata dettata dalla necessità di avere più spazio per lo sbroglio dei fili e per consentire una migliore dissipazione del calore generato dal regolatore. Pertanto, le dimensioni finali della scheda sono state fissate a 56,5mm di larghezza e 65mm di lunghezza.

Le clearances posizionate ai quattro lati della scheda presentano un diametro interno di 2,7mm, un diametro esterno di 3mm e una distanza del centro dal bordo di 3,5mm. [20]

Durante la realizzazione dei collegamenti, si è prestata particolare attenzione alle dimensioni delle piste e al loro percorso, poiché un collegamento errato o piste troppo strette possono causare problemi durante la fase di saldatura e

compromettere il corretto funzionamento del circuito. Pertanto, le piste destinate all'alimentazione e alle uscite sono state dimensionate in modo più ampio rispetto alle altre. Inoltre, le piste sono state progettate in modo da seguire la direzione dei pin dei componenti e da essere il più centrali possibile rispetto ad essi.

Per quanto riguarda le piste utilizzate per la comunicazione, si è cercato di renderle tutte della stessa lunghezza, facendole passare tutte dalla parte alta della scheda, al fine di evitare ritardi.

Un'ulteriore attenzione è stata dedicata al posizionamento della pista di selezione del chip (CS) tra le linee di dati (MOSI e MISO) e il clock (SCLK). Questo è stato fatto perché il clock commuta proprio durante l'acquisizione dei dati, e a frequenze elevate ciò potrebbe generare rumore e impulsi aggiuntivi a causa dell'accoppiamento capacitivo tra piste adiacenti.

Infine, per quanto riguarda il piano di massa, si è cercato di ridurre al minimo i collegamenti su questa superficie, poiché questi tagliano il piano di massa stesso e possono causare problemi, specialmente se i collegamenti sono molto lunghi.

Si può vedere la realizzazione finale del circuito stampato in figura 10.1

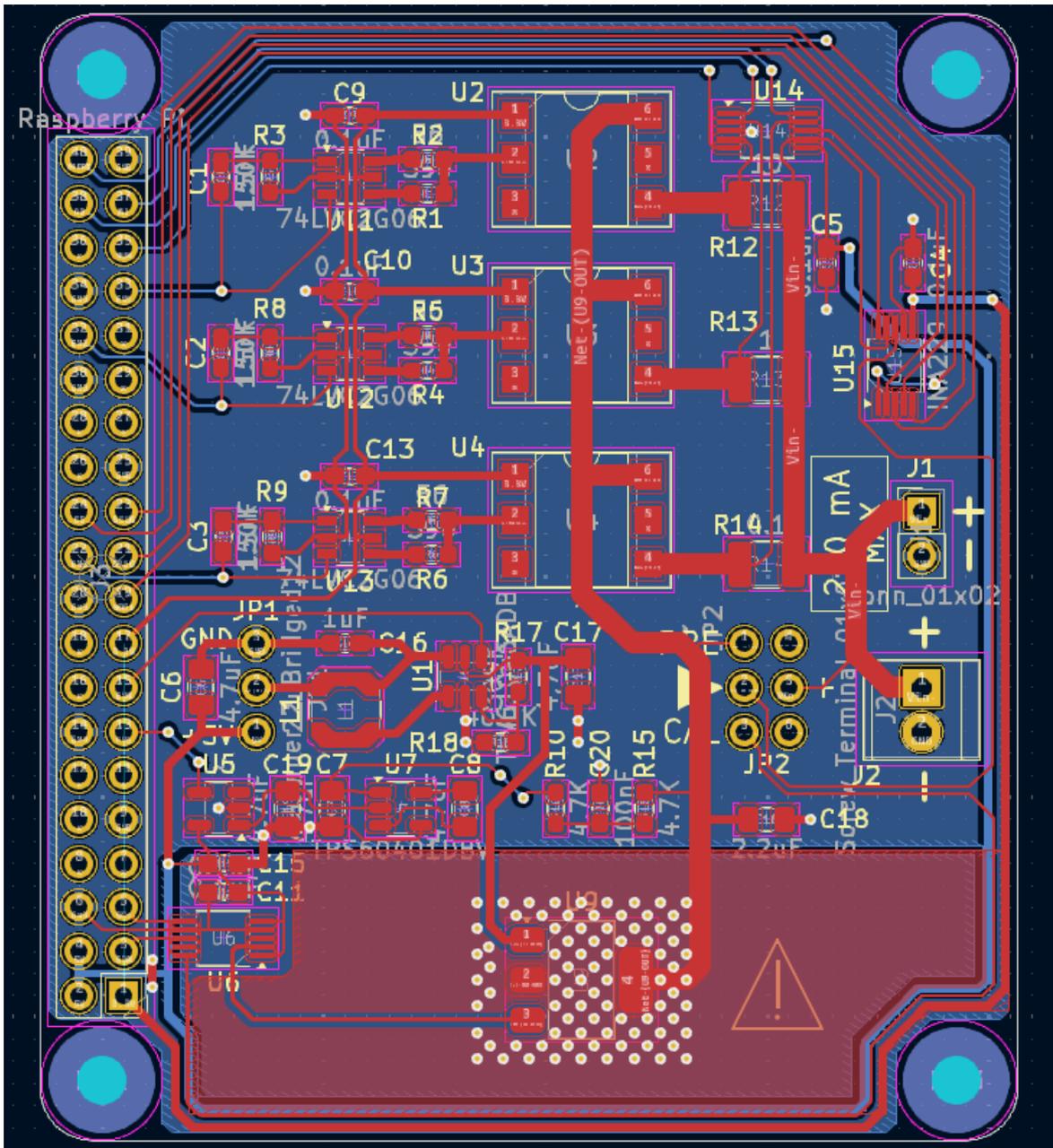


Figura 10.1: schema del circuito stampato finale

## 11. ANALISI DELLE MODIFICHE AL SOFTWARE

Poiché il progetto era già esistente, il software era già stato sviluppato.

Tuttavia, è necessario apportare modifiche alla gestione del DAC e dell'INA, poiché questi componenti sono stati sostituiti con versioni più recenti, e adeguare alcune parti relative ai relay.

Si è iniziato con un'analisi approfondita del software per comprendere appieno il suo funzionamento. Successivamente, si è passati a un'accurata analisi dei registri del DAC80501 e dell'INA229, per capire come devono essere utilizzati nel programma per garantire il corretto funzionamento del circuito.

Il programma è composto da diversi file, ognuno con una funzione specifica: alcuni gestiscono i componenti, altri la comunicazione, e altri ancora l'interfaccia grafica. Infine, c'è il corpo principale del programma, che richiama tutte le funzioni necessarie al momento opportuno, ovvero il main. L'organizzazione è modulare, con l'utilizzo di file header che separano le dichiarazioni delle funzioni dalle loro definizioni. Questo approccio migliora l'organizzazione del codice e ne facilita la manutenzione. Inoltre, il

programma salva tutti i dati in un file txt, visibile una volta chiuso il programma.

Quando viene caricato sulla Raspberry Pi e avviato, si apre un'interfaccia grafica sullo schermo del PC, tramite la quale è possibile variare la tensione d'uscita richiesta e la resistenza d'uscita per modificare la precisione della misurazione.

I file che è stato necessario modificare sono principalmente quelli relativi al DAC (denominato power.c), all'INA229 (precedentemente chiamato INA226.c), ai relay (relay.c) e il file che conteneva le costanti corrispondenti ai pin di enable dei componenti (config.h). Inoltre, è stato necessario inserire un nuovo file per la comunicazione seriale tramite SPI, denominato spi.c.

OFFSET	REGISTER NAME	REGISTER DESCRIPTION
0h	NOOP	No operation
1h	DEVID	Device identification
2h	SYNC	Synchronization
3h	CONFIG	Configuration
4h	GAIN	Gain
5h	TRIGGER	Trigger
7h	STATUS	Status
8h	DAC	Digital-to-analog converter

*Tabella 11.1: Registri del DAC80501*

### ***11.1. Analisi power.c***

Il file “power.c” è utilizzato per l’accensione dell’interruttore della pompa di carica, del booster e per la gestione del DAC.

Il DAC è formato da 9 registri, tutti a 16 bit, riportati nella Tabella 11.1, ognuno utilizzato per uno specifico compito:

0. NOOP: registro scrivibile il quale compie nessuna operazione se scritto un qualsiasi valore all’interno.
1. DEVID: registro di sola lettura utilizzato per conoscere la risoluzione del DAC (bit 14 – 12) e il valore di tensione in uscita che viene erogato quando viene resettato (bit 7).

Utile se si vuole controllare la tipologia di DAC e le sue caratteristiche.

Nel nostro caso leggendo questo registro si riceverebbe 0115h come valore, essendo un DAC80501Z.

2. SYNC: questo registro, leggibile e scrivibile, viene usato per impostare il DAC in modalità sincrona o asincrona.

Infatti, quando il bit0 vale 1, l’uscita è impostata per aggiornarsi in risposta a un trigger sul bit “LDAC” del registro “TRIGGER” (modalità sincrona).

Quando vale 0, l'uscita è impostata per l'aggiornamento immediato (modalità asincrona), impostazione predefinita.

3. CONFIG: questo registro, scrivibile e leggibile, viene utilizzato per spegnere e accendere il componente e/o la sua tensione di riferimento interna.

Tramite il bit8 si può accendere (bit8=1) o spegnere (bit8=0) la tensione di riferimento; quando il bit0 viene impostato a 0 il DAC è acceso, invece, quando il bit0 viene impostato a 1, il DAC è spento e l'uscita è collegata a GND tramite un resistore interno da 1 k $\Omega$ .

4. GAIN: Come spiegato alla fine del paragrafo 4.1 il riferimento del DAC può essere diviso o moltiplicato per 2 a seconda delle esigenze, ciò viene gestito tramite questo registro di lettura e scrittura.

Infatti, impostando il bit8 a 1 si può sfruttare la divisione della tensione di riferimento, impostando il bit0 a 1 si aumenta il riferimento di un fattore 2. Di default bit8=0 e bit0=1.

5. TRIGGER: questo registro è solo scrivibile e contiene il bit4 "LDAC" che viene impostato a 1 per caricare in modo sincrono l'uscita del DAC. Questo bit si riporta automaticamente a 0.

I bit 3-0 se impostati a 1010 ripristinano il dispositivo allo stato di default. Questi bit si ripristinano automaticamente a 0.

7. STATUS: registro di sola lettura dove sul bit0 si legge 1 quando la differenza tra i pin di riferimento e di alimentazione è inferiore a una soglia analogica minima, altrimenti si legge 0. Quando è 1, il buffer di riferimento viene spento e l'uscita è a 0V. I registri non vengono influenzati e l'uscita ritorna normale quando la differenza è superiore alla soglia analogica.
8. DAC: registro di lettura e scrittura dove viene letto o scritto il valore in uscita del DAC.

Nel file `power.c` ci sono cinque funzioni per gestire il DAC:

- *“bool power\_init (void)”*: controlla se il sottosistema GPIO è già inizializzato, inizia la comunicazione con il DAC, legge il registro “DEVID” (01h) per controllare che sia effettivamente il DAC80501Z (0115h), controlla la corretta accensione del booster e dell'interruttore della pompa e in caso contrario li spegne. In più imposta i GPIO come output. Se il registro “STATUS” è settato nella maniera corretta copia il valore del registro “DAC” in una variabile, per conoscere il valore attuale della tensione in uscita e utilizzarlo nelle altre funzioni. (Figura 11.1)

```

38 bool power_init (void)
39 {
40     bool gpio_ok = gpio_init();
41     if (!gpio_ok) return false;
42
43     if (!open_i2c_dev(DAC50801_I2C_BUS, DAC50801_I2C_DEV)) {
44         fprintf(stderr, MYNAME "unable to open I2C bus!\n");
45         return false;
46     }
47
48     // check that there is actually a DAC50801 there:
49     uint16_t id = read_i2c_reg(0x01);
50     if (id != 0x0115) {
51         fprintf(stderr, MYNAME "device id not found!\n");
52         return false;
53     }
54
55     // check if power was already on:
56     enabled_booster = gpio_get_value(BOOSTER_EN);
57     enabled_switch = gpio_get_value(SWITCH_EN);
58
59
60
61     uint16_t status = read_i2c_reg(0x04);
62     if ((status & 0x0101) == 0x0001) {
63         dac_code = read_i2c_reg(0x08);
64     } else {
65         enabled_booster = false;
66         enabled_switch = false;
67     }
68
69     if (!(enabled_booster && enabled_switch) ) {
70         write_i2c_reg(0x03, 0x0101);
71         enabled_booster = false;
72         enabled_switch = false;
73     }
74
75
76     gpio_set_value(BOOSTER_EN, enabled_booster);
77     gpio_set_fsel(BOOSTER_EN, 1);
78     gpio_set_value(SWITCH_EN, enabled_switch);
79     gpio_set_fsel(SWITCH_EN, 1);
80
81     write_i2c_reg(0x04, 0x0001);    // setta il guadagno
82     return true;
83 }

```

Figura 11.1: funzione “power\_init” del file “power.c”

- “*void power\_enable (bool en)*”: riceve in input il valore booleano true o false a seconda se si vuole spegnere o accendere l’interruttore, la pompa e il DAC.

Se i componenti sono già nello stato desiderato non esegue le istruzioni successive, altrimenti si appresta a portarli nello stato voluto.

(Figura 11.2)

```

85 void power_enable (bool en)
86 {
87     if ( (enabled_booster == en) && (enabled_switch == en) ) return;
88     enabled_booster = en; enabled_switch = en;
89     if ( en ) {
90         write_i2c_reg(0x08, dac_code);
91         gpio_set_value(BOOSTER_EN, 1); //booster on
92         //delay
93         gpio_set_value(SWITCH_EN, 1); //switch on
94         write_i2c_reg(0x03, 0x0000); // power on.
95     }
96     } else {
97         write_i2c_reg(0x03, 0x0101); //disable device internal reference and set the DAC
98         //in power-down mode and the DAC output is connected to GND through a 1-kΩ internal resistor.
99         gpio_set_value(SWITCH_EN, 0); // switch off
100        gpio_set_value(BOOSTER_EN, 0); //booster off
101    }
102 }

```

Figura 11.2: funzione “*power\_enable*” del file “*power.c*”

- “*bool power\_status (void)*”: restituisce lo stato in cui si trovano i componenti, se sono accesi o spenti. (Figura 11.3)

```

104 bool power_status (void)
105 {
106     return enabled_booster && enabled_switch;
107 }

```

Figura 11.3: funzione “*power\_status*” del file “*power.c*”

- “*int power\_get (void)*”: restituisce il valore della tensione che sta erogando, prestando attenzione agli arrotondamenti. (Figura 11.4)

```
108 int power_get (void)
109 {
110     if (dac_code == 0x0000) return 0;
111     unsigned actual = (dac_code * 5000 + 32768) / 65536;
112
113     return actual;
114     // Vout = dac_code * fondo_scala_analogico / fondo_scala_digitale
115 }
```

Figura 11.4: funzione “*power\_get*” del file “*power.c*”

- “*bool power\_set (unsigned voltage)*”: imposta il valore del registro DAC alla tensione desiderata gestendo in maniera accurata gli arrotondamenti e gli overflow. (Figura 11.5)

Per evitare di scrivere più volte consecutive sul registro “DAC”, in quanto questa operazione mandava in blocco il componente, si è inserito l’if finale per gestire questa problematica.

```

117  bool power_set (unsigned voltage)
118  {
119      if (voltage > 5000) return false;
120
121      unsigned desired = (voltage * 65536 + 2500) / 5000;
122      if (desired > 0xFFFF) desired = 0xFFFF;
123
124      dac_code = desired;
125
126      if (power_status()) write_i2c_reg(0x08, dac_code);
127      return true;
128  }

```

*Figura 11.5: funzione “power\_set” del file “power.c”*

## **11.2. Analisi ina229.c**

Il file “ina229.c” gestisce i registri del componente INA229 tramite i quali si regola l’accensione, lo spegnimento, il campionamento della tensione ai capi del resistore e vari parametri utilissimi per avere la massima accuratezza nella misurazione della corrente d’uscita.

I registri principali utilizzati per questo progetto sono:

0. CONFIG: Questo registro di lettura/scrittura a 16 bit contiene i bit per resettare il componente (bit 15 RST), per resettare i registri “ENERGY” e “CHARGE” (bit 14 RSTACC), per impostare un ritardo nell’inizio della conversione ADC (bit 13 – 6 CONVDLY), per

abilitare la compensazione della temperatura (bit 5 TEMPCOMP) e per la selezione del range di fondo scala su IN+ e IN- (bit 4 ADCRANGE).

1. ADC\_CONFIG: Questo registro di lettura/scrittura a 16 bit contiene i bit per configurare la conversione ADC.

Tramite questi si può selezionare la modalità di campionamento e quale tensione campionare (bit 15 – 12 MODE) , il tempo di conversione della tensione VBUS e VSHUNT e della temperatura (risp. Bit 11 – 9 VBUSCT, bit 8 – 6 VSHCT, bit 5 – 3 VTCT), in più si può selezionare il numero di campioni per fare la media prima di restituire il risultato (bit 2 – 0 AVG); i produttori consigliano un maggior numero di campioni e un maggiore tempo di conversione per una riduzione del rumore.

2. SHUNT\_CAL: Questo registro di lettura/scrittura a 16 bit fornisce al dispositivo un valore costante di conversione che rappresenta la resistenza di shunt utilizzata per calcolare il valore di corrente in Ampere. Questo imposta anche la risoluzione per il registro “CURRENT”. Il calcolo del valore è descritto da:

$$\text{SHUNT\_CAL} = 13107.2 \cdot 106 \cdot \text{CURRENT\_LSB} \cdot \text{RSHUNT}$$

Dove CURRENT\_LSB viene calcolato da:

$$\text{CURRENT\_LSB} = \frac{\text{Maximum Expected Current}}{2^{19}}$$

ed è la massima risoluzione della corrente.

4. VSHUNT: Questo registro di sola lettura a 24 bit contiene il valore di tensione misurato sulla resistenza di shunt, che può essere positivo o negativo a seconda del verso della corrente. I bit dedicati per la misura sono i bit 23 – 4.
5. VBUS: Questo registro di sola lettura a 24bit contiene il valore di tensione misurato tra il pin VBUS e massa. I bit dedicati alla misura sono i bit 23 – 4.
7. CURRENT: Questo registro di sola lettura a 24bit contiene il valore di corrente. Dopo aver programmato il registro “SHUNT\_CAL” la corrente misurata in Ampere può essere letta su questo registro. Il valore finale è scalato da CURRENT\_LSB e calcolato tramite:  
$$\text{Current [A]} = \text{CURRENT\_LSB} \cdot \text{CURRENT}$$
I bit dedicati alla misura sono i bit 23 – 4.

B. DIAG\_ALERT: Questo registro di lettura/scrittura (con alcune eccezioni di bit di sola lettura) contiene tutti i bit che segnalano l'occorrenza di specifiche situazioni, ad esempio quando su alcuni registri avviene un overflow, quando si superano dei valori limite prefissati della temperatura, della tensione, della potenza; ci sono anche

alcuni bit per abilitare la conversione e per sapere quando essa è stata completata.

3E. `MANUFACTURER_ID`: Questo registro di sola lettura contiene la scritta “TI” in ASCII, che corrisponde alle iniziali del produttore (Texas Instrument).

3F. `DEVICE_ID`: Questo registro di sola lettura contiene il numero “2290”, che corrisponde all’ID del componente.

Le funzioni sviluppate per gestire questo componente sono:

- “bool ina229\_init (ina229\_callback cb)”: Avvia la comunicazione aprendo il bus SPI, verifica la sua presenza leggendo i registri “`MANUFACTURER_ID`”(5449h) e “`DEVICE_ID`”(2290h) e facendo una verifica del contenuto, inizializza il dispositivo INA229 configurando il registro “`CONFIG`” in modo tale da resettarlo, stoppa la conversione ADC resettando il registro “`ADC_CONFIG`” e abilita l’interrupt per la “conversione pronta” (bit 14 registro `DIAG_ALERT`). (Figura 11.6)

```

45 bool ina229_init (ina229_callback cb)
46 {
47
48     if (!spi_init()) {
49         fprintf(stderr, MYNAME "unable to open SPI bus!\n");
50         return false;
51     }
52
53     // check that there is actually an INA229 there:
54     struct timespec t1, t2;
55     clock_gettime(CLOCK_MONOTONIC, &t1); //INA229 3F Die ID Contains unique die identification
56     uint16_t id1 = read_SPI_reg_16(0x3E); //INA229 Manufacturer ID (MANUFACTURER_ID) Register (Address = 3Eh)
57     uint16_t id2 = read_SPI_reg_16(0x3F); //INA229 Device ID (DEVICE_ID) Register (Address = 3Fh)
58     clock_gettime(CLOCK_MONOTONIC, &t2);
59     if (id1 != 0x5449 || id2 != 0x2291) {
60         fprintf(stderr, MYNAME "device id not found!\n");
61         return false;
62     }
63     double t = (t2.tv_nsec - t1.tv_nsec) * 0.000000001 + (t2.tv_sec - t1.tv_sec);
64     printf(MYNAME "I2C access time: %e\n", t);
65
66     gpio_file = open("/sys/class/gpio/gpio" STRING(INA229_IRQ_GPIO) "/value", O_RDONLY);
67     if (gpio_file < 0) {
68         perror(MYNAME "unable to open IRQ GPIO device");
69         return false;
70     }
71
72     write_SPI_reg_16(0x00, 0xC010); // reset INA229
73     write_SPI_reg_16(0x01, 000000); // stop conversion ADC
74     write_SPI_reg_16(0x0B, 0x4000); // enable "conversion ready" interrupt 0x0B , 0x4000 BIT 14
75
76     callback = cb;
77
78     return true;
79 }

```

Figura 11.6: funzione “ina229\_init” del file “ina229.c”

- “void ina229\_enable (bool en)”: Questa funzione abilita o disabilita il dispositivo e avvia o ferma la conversione in base al suo ingresso.

(Figura 11.7)

```

81 void ina229_enable (bool en)
82 {
83     if ((enabled = en)){
84         write_SPI_reg_16(0x00, 0x0010);
85         write_SPI_reg_16(0x01, 0172220); // start conversions
86     }
87     else{
88         write_SPI_reg_16(0x01, 000000); // stop conversion ADC
89     }
90
91 }

```

Figura 11.7: funzione “ina229\_enable” del file “ina229.c”

- “bool ina229\_status (void)”: Questa funzione restituisce lo stato del dispositivo, se è spento o acceso. (Figura 11.8)

```
93  bool ina229_status (void)
94  {
95      return enabled;
96  }
```

*Figura 11.8: funzione “ina229\_status” del file “ina229.c”*

- “bool ina229\_run (void)”: Questa funzione esegue un ciclo continuo di lettura dei dati dal dispositivo INA229 tramite SPI, gestendo i GPIO e chiamando la funzione “callback” per elaborare i dati letti (il tempo, la tensione sulla resistenza e la tensione sul pin VBUS, shiftati a destra di 4 per considerare i bit che contengono la misura). La funzione è destinata a funzionare fino a quando una variabile globale exit\_from\_loop non viene impostata su true, indicando che il ciclo deve terminare. (Figura 11.9)

```

98  bool ina229_run (void)
99  □ {
100     struct timespec t2;
101     int i;
102     while (!exit_from_loop) {
103         uint64_t t;
104         char status = '-';
105         struct timeval timeout = {1, 0};
106         fd_set fds;
107         FD_ZERO(&fds);
108         FD_SET(gpio_file, &fds);
109         if (lseek(gpio_file, 0, SEEK_SET)) {
110             perror("GPIO seek failed");
111             return false;
112         }
113         if (read(gpio_file, &status, 1) != 1) {
114             perror("GPIO read failed");
115             return false;
116         }
117         if (status == '1') {
118             int s = select(gpio_file + 1, NULL, NULL, &fds, &timeout);
119             int e = clock_gettime(CLOCK_MONOTONIC, &t2);
120             if (s < 0 || e < 0) {
121                 perror("GPIO poll failed");
122                 return false;
123             } else if (!s) {
124                 if (!enabled) continue;
125                 fprintf(stderr, "INA226 interrupt timeout.\n");
126                 return false;
127             }
128             t = timespec_diff(&t2, &t0);
129         } else {
130             int e = clock_gettime(CLOCK_MONOTONIC, &t2);
131             t = timespec_diff(&t2, &t0) + TIMINGERROR_FLAG;
132         }
133     }

150     □ //uint16_t ss = read_SPI_reg_16(0x0B);
151         int32_t vs = read_SPI_reg_24_int(4) >> 4;
152         int32_t vb = read_SPI_reg_24(5) >> 4;
153     □ //int16_t dietemp = read_SPI_reg_16(6);
154         int32_t current = read_SPI_reg_24_int(7) >> 4;
155         //uint32_t power = read_SPI_reg_24(8);
156         //uint64_t energy = read_SPI_reg_40(9);
157         //int64_t charge = read_SPI_reg_40_int(0x0A);
158
159         callback(t, vb, vs);
160     }

```

Figura 11.9: funzione “ina229\_run” del file “ina229.c”

- Infine, è stato necessario sviluppare delle funzioni per scrivere e leggere sui registri dell'INA229 tramite SPI. Poiché i registri variano in

dimensione, sono state implementate diverse funzioni per gestire ciascun registro specifico, considerando che le variabili in C hanno dimensioni multiple di 2, mentre alcuni registri non seguono questa regola. Pertanto, è stato necessario apportare correzioni, anche perché alcuni valori erano rappresentati in complemento a due (con segno) e altri no.

### ***11.3. Analisi relay.c***

Il file “relay.c” gestisce i tre photorelay, accendendone uno alla volta a seconda della precisione e dei valori in uscita richiesti e di conseguenza gestisce anche il MUX per un corretto collegamento con l’INA229.

È composta da cinque funzioni:

- “*bool relay\_init (void)*”: serve per controllare lo stato precedente dei GPIO relativi ai relay, settare quello corretto e impostarli come output insieme a quelli relativi al MUX. (Figura 11.10)

```

17 bool relay_init (void)
18 {
19     bool gpio_ok = gpio_init();
20     if (!gpio_ok) {
21         fprintf(stderr, MYNAME "unable to open GPIO port!\n");
22         return false;
23     }
24
25     // determine if system was previously on:
26     uint32_t on = gpio_mask_get(RELAY_MASK);
27     // and select the lowest-indexed relay:
28     index = 0;
29     if (on & (1 << RELAY_EN3)) index = 3;
30     if (on & (1 << RELAY_EN2)) index = 2;
31     if (on & (1 << RELAY_EN1)) index = 1;
32
33     // clean up GPIO settings:
34     relay_set(index);
35     // set all GPIOs as outputs:
36     gpio_set_fsel(RELAY_EN1, 1);
37     gpio_set_fsel(RELAY_EN2, 1);
38     gpio_set_fsel(RELAY_EN3, 1);
39     gpio_set_fsel(MUX_A0, 1);
40     gpio_set_fsel(MUX_A1, 1);
41     gpio_set_fsel(MUX_EN, 1);
42
43     return true;
44 }

```

Figura 11.10: funzione “*relay\_init*” del file “*relay.c*”

- “*void relay\_enable (bool en)*”: attiva o disattiva i relay in base all’input che riceve la funzione. (Figura 11.11)

```

46 void relay_enable (bool en)
47 {
48     if ((enabled = en))
49         relay_set(index);
50     else
51         relay_set(!index);
52 }
53

```

Figura 11.11: funzione “*relay\_enable*” del file “*relay.c*”

- “*bool relay\_status (void)*”: restituisce come output lo stato attuale di on/off del relay. (Figura 11.12)

```
54  bool relay_status (void)
55  {
56      return enabled;
57  }
58
```

*Figura 11.12: funzione “relay\_status” del file “relay.c”*

- “*int relay\_get (void)*”: restituisce il numero del relay in uso. (Figura 11.13)

```
59  int relay_get (void)
60  {
61      return index;
62  }
```

*Figura 11.13: funzione “relay\_get” del file “relay.c”*

- “*bool relay\_set (unsigned n)*”: attiva il relay corrispondente al numero che riceve in ingresso, controllando che sia un numero corretto. Tramite il MUX collega la resistenza scelta all’INA. (Figura 11.14)

```

64  bool relay_set (unsigned n)
65  {
66      if (n >= 4) return false;
67
68      uint8_t mux_a = mux_inputs[n ? n : enabled];
69      gpio_set_value(MUX_A0, !(mux_a & 0x01));
70      gpio_set_value(MUX_A1, !(mux_a & 0x02));
71      gpio_set_value(MUX_EN, enabled);
72
73      uint32_t mask = 0;
74      if (n) {
75          mask = 1 << relay_gpio[n];
76          gpio_mask_set(mask);
77          spindelay_us(100); // da cambiare
78
79      }
80      gpio_mask_clr(mask ^ RELAY_MASK);
81      index = n;
82
83      return true;
84  }

```

Figura 11.14: funzione “relay\_get” del file “relay.c”

#### 11.4. Analisi config.h

“config.h” è il file che contiene tutte le costanti utili, come ad esempio i GPIO di tutti i componenti, i numeri dei bus, l’indirizzo del DAC, i valori delle resistenze, come mostrato in figura 11.15.

I numeri dei bus sono settati per funzionare su Raspberry Pi 4.

```

1  #ifndef PIN_CONFIG_H
2  #define PIN_CONFIG_H
3
4  #define DAC50801_I2C_BUS      3 // I2C bus number
5  #define DAC50801_I2C_DEV    0x48 // I2C slave address
6  #define INA229_SPI_BUS      0 // SPI bus number
7  #define INA229_SPI_DEV      0 // SPI CS# number
8  #define INA229_IRQ_GPIO     533 // (pin 40) must enable pull-up on GPIO
9
10 #define BOOSTER_EN          22 //(pin 15) GPIO (active high)
11 #define SWITCH_EN          27 //(pin 13) GPIO (active high)
12
13 #define RELAY_EN1           25 //(pin 22) GPIO (active high)
14 #define RELAY_EN2           12 //(pin 32) GPIO (active high)
15 #define RELAY_EN3           13 //(pin 33) GPIO (active high)
16
17 #define MUX_A0              26 //(pin 37) GPIO
18 #define MUX_A1              20 //(pin 38) GPIO
19 #define MUX_EN              19 //(pin 35) GPIO (active high)
20
21 #define MUX_SEL0            3 // Values in MUX_A[1:0] to select 0 Ω
22 #define MUX_SEL1            1 // Values in MUX_A[1:0] to select RES1
23 #define MUX_SEL2            0 // Values in MUX_A[1:0] to select RES2
24 #define MUX_SEL3            2 // Values in MUX_A[1:0] to select RES3
25
26 #define RES1                0.1 // [Ω] sense resistor value for RELAY 1
27 #define RES2                1.0 // [Ω] sense resistor value for RELAY 2
28 #define RES3                10.0 // [Ω] sense resistor value for RELAY 3
29
30
31 #define STRING(x) TO_STR(x)
32 #define TO_STR(x) #x
33
34 #endif

```

*Figura 11.15: file “config.h”*

## 11.5. Analisi main.c

La funzione main esegue le seguenti operazioni:

- Legge i comandi da un file;

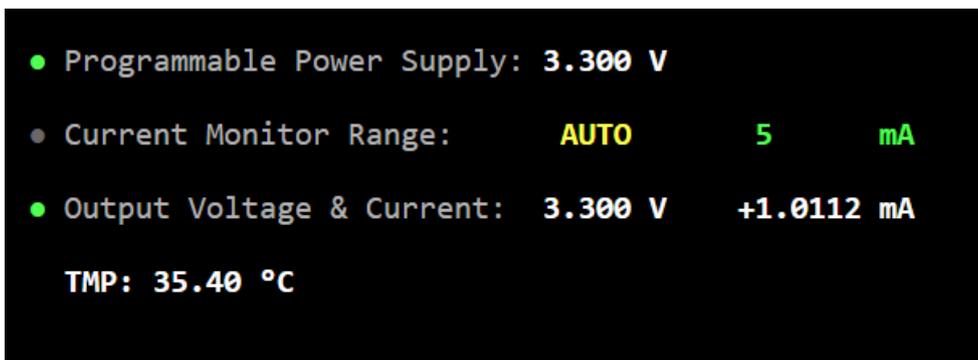
- Inizializza il timestamp e apre un file di output per i dati della misurazione;
- Inizializza l'hardware (relè, alimentazione, INA, DAC);
- Avvia il thread per la misurazione dell' INA229;
- Abilita le misurazioni e l'interfaccia utente per il controllo in tempo reale del sistema;
- Alla chiusura del programma disabilita il relè e attende la fine del thread.

## 12. MISURAZIONI

Dopo aver ricevuto la scheda mandata in produzione e tutti i componenti necessari si è proceduto con la fase di saldatura, successivamente si è collaudata la scheda per verificarne il corretto funzionamento attraverso una serie di test e misurazioni.

La prima prova della scheda è stata eseguita tramite una misurazione, realizzata collegando un carico con una resistenza in ingresso da  $3.3\text{k}\Omega$ .

In figura 12.1 possiamo vedere la schermata utente del programma che riporta una tensione d'uscita di  $3.300\text{V}$ , una corrente di  $+1.0112\text{mA}$  sul carico, valore conforme alle aspettative.

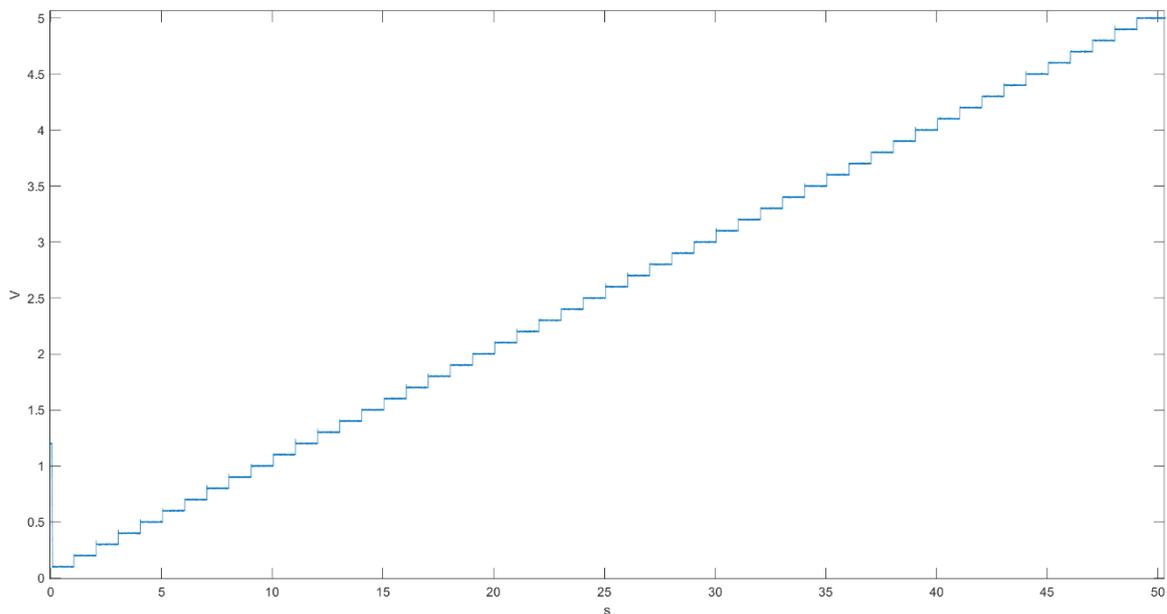


*Figura 12.1: Schermata utente della prima misurazione eseguita*

Sono state eseguite delle misurazioni variando la tensione nel tempo del DAC e osservando il comportamento della tensione d'uscita del circuito:

- 1) La tensione di uscita del DAC, partendo da 100mV, varia ogni secondo di 100mV fino ad arrivare ad un massimo di 5000mV, senza il collegamento di un carico in uscita.

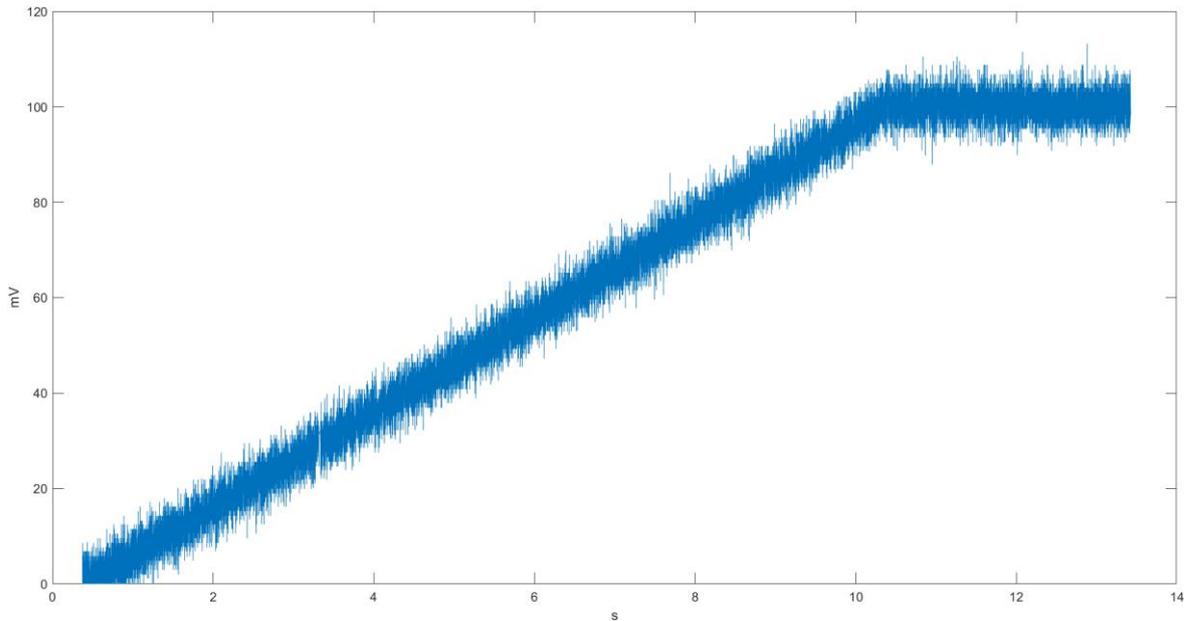
Tramite i dati temporali della tensione salvati dal programma su un file .txt si è successivamente generato un grafico utilizzando il software MATLAB per visualizzare l'andamento della tensione nel tempo, come mostrato in figura 12.2.



*Figura 12.2: Andamento nel tempo della tensione  $V_{bus}$  quando l'uscita del DAC varia ogni secondo di 100mV partendo da 100mV*

2) La tensione di uscita del DAC, partendo da 1mV, varia ogni 100ms di 1mV fino ad arrivare ad un massimo di 100mV, senza il collegamento di un carico in uscita.

L'uscita è mostrata nella figura 12.3.

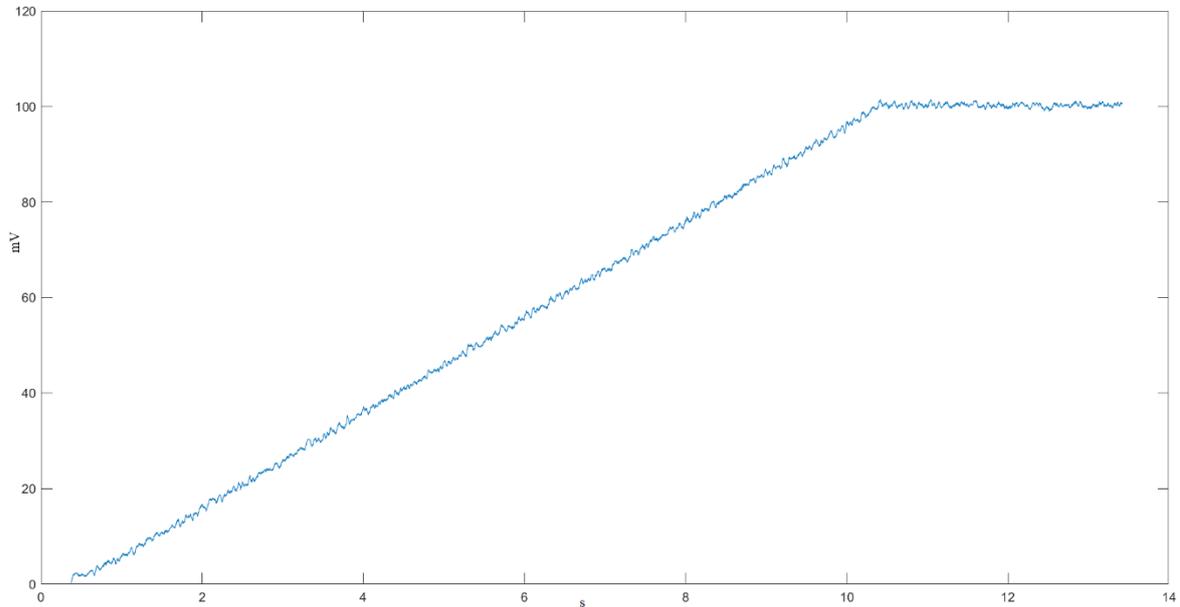


*Figura 12.3: Andamento nel tempo della tensione  $V_{bus}$  quando l'uscita del DAC varia ogni 100ms di 1mV partendo da 1mV*

Per avere una visione migliore della misurazione si è optato per mediare il campionamento.

Essendo che il segnale varia ogni 100ms e il programma campiona ogni 25 $\mu$ s si può mediare ogni 1000 campioni e mantenere intatta la misura, in quanto si rispetta il teorema del campionamento di Nyquist.

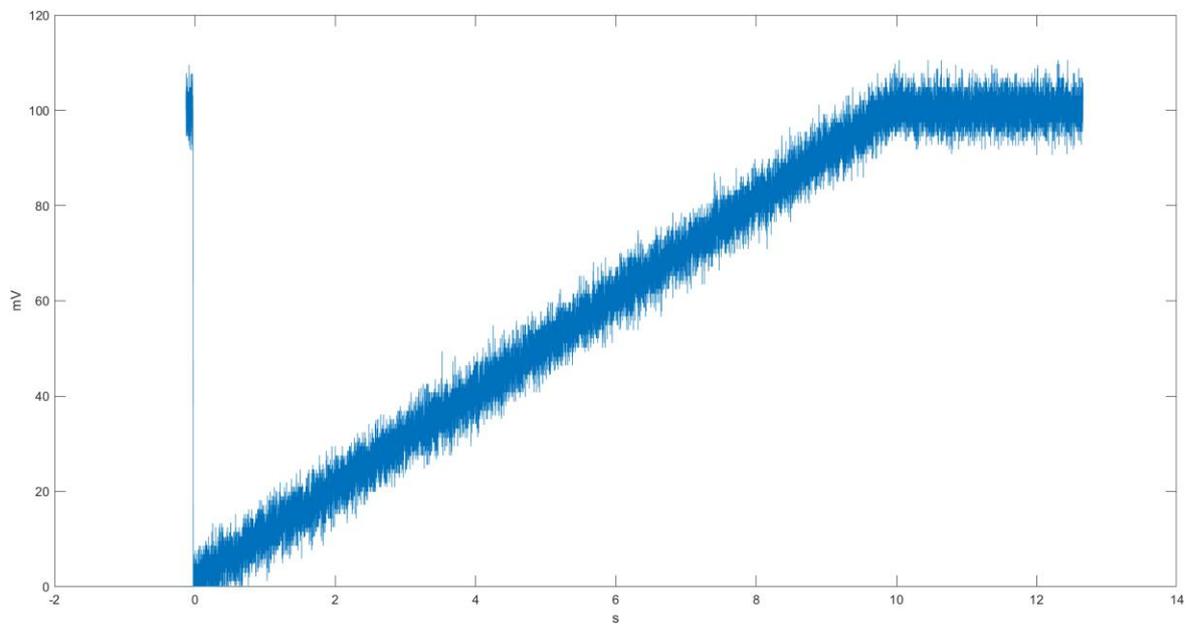
Il risultato è mostrato in figura 12.4, dove si può notare un notevole miglioramento nella visualizzazione della misura.



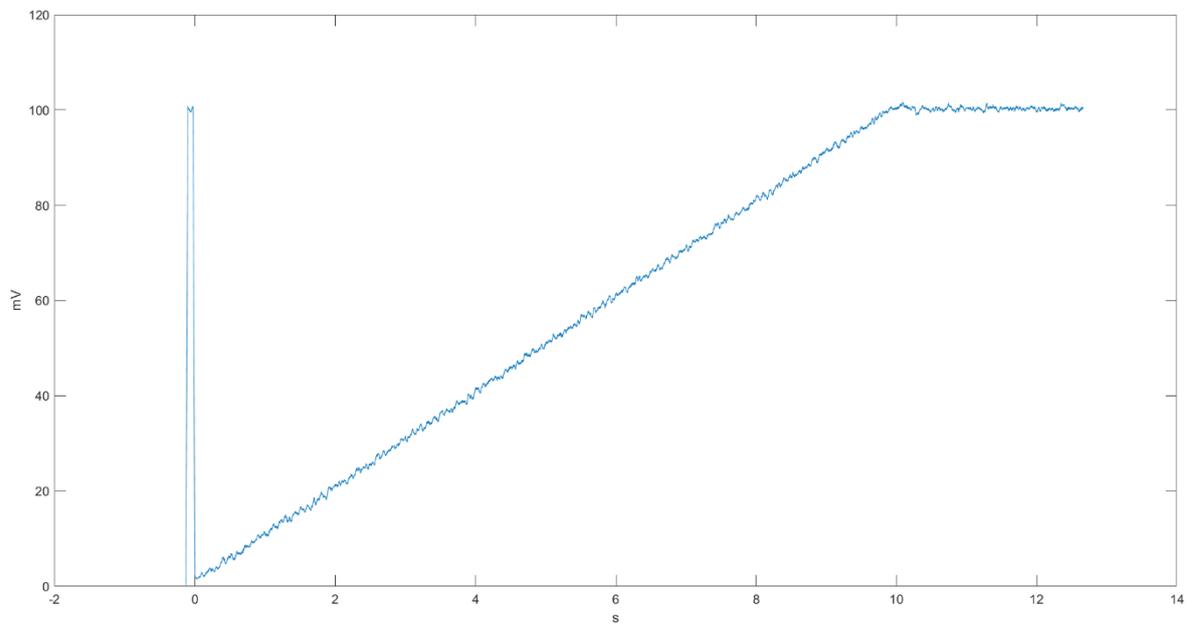
*Figura 12.4: Andamento nel tempo della tensione  $V_{bus}$  quando l'uscita del DAC varia ogni 100ms di 1mV partendo da 1mV dopo aver mediato ogni 1000 campioni*

- 3) La tensione di uscita del DAC, partendo da 1mV, varia ogni 100ms di 1mV fino ad arrivare ad un massimo di 100mV, collegando un carico in uscita di 3.3k $\Omega$ .

In figura 12.5 si può vedere l'andamento nel tempo della tensione  $V_{bus}$ , molto simile all'andamento precedente; infatti, anche in questo caso si è ritenuto necessario mediare per ottenere una misurazione più leggibile, come si può vedere in figura 12.6.



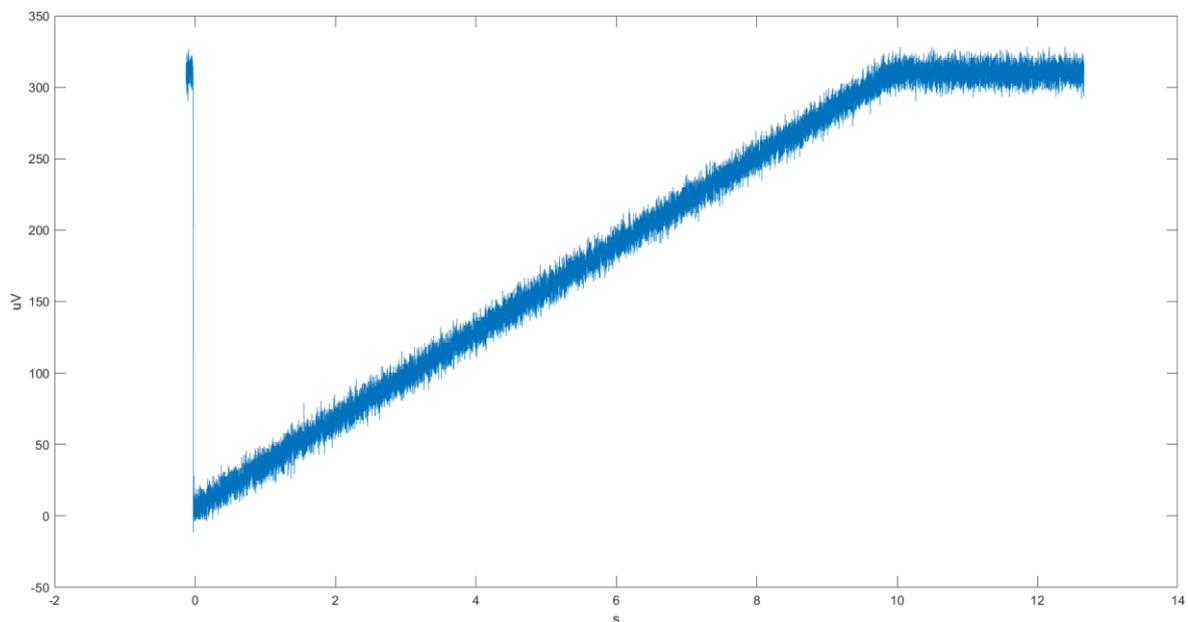
*Figura 12.5: Andamento nel tempo della tensione  $V_{bus}$  quando l'uscita del DAC varia ogni 100ms di 1mV partendo da 1mV, con un carico collegato in uscita di  $3.3k\Omega$*



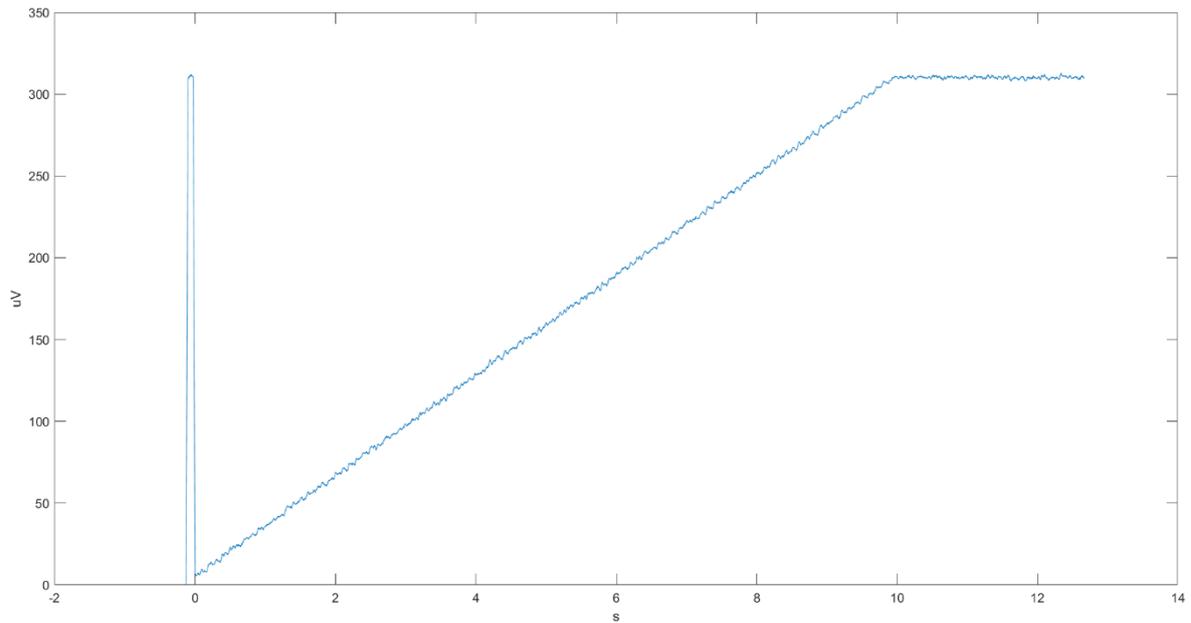
*Figura 12.6: Andamento nel tempo della tensione  $V_{bus}$  quando l'uscita del DAC varia ogni 100ms di 1mV partendo da 1mV dopo aver mediato ogni 1000 campioni, con un carico collegato in uscita di 3.3k $\Omega$*

Si può notare in questo caso, avendo collegato un carico, la variazione di tensione ai capi del resistore di shunt, raffigurato in figura 12.7.

Anche in questo caso si è dovuto ricorrere alla mediazione ogni 1000 campioni, come si può notare in figura 12.8.



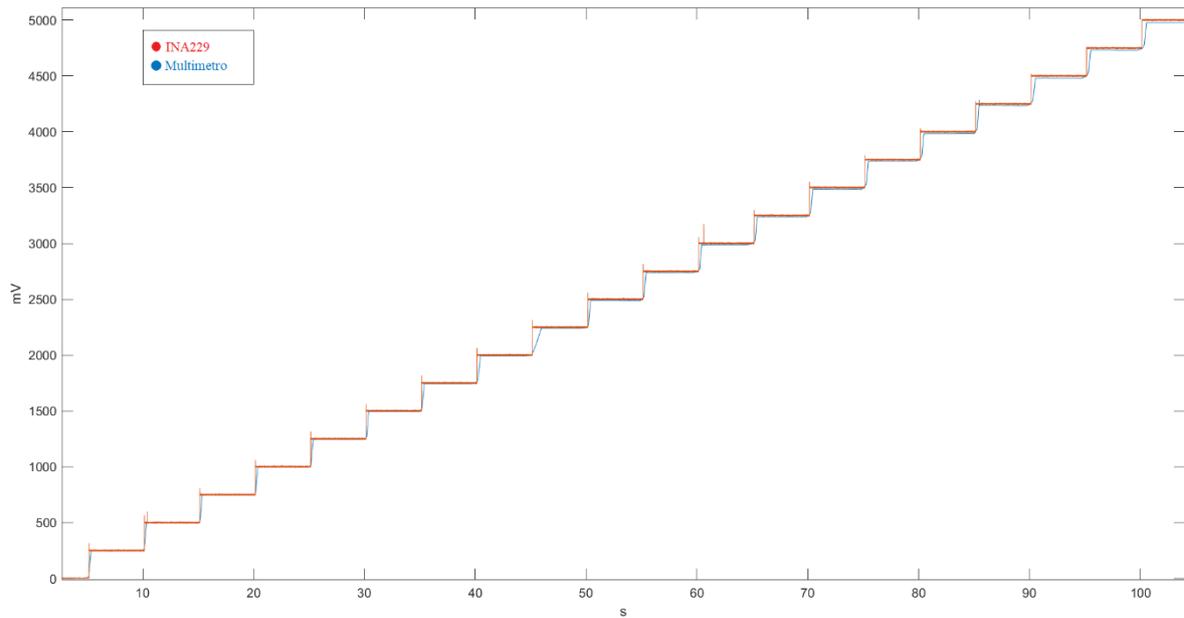
*Figura 12.7: Andamento nel tempo della tensione  $V_{shunt}$  quando l'uscita del DAC varia ogni 100ms di 1mV partendo da 1mV, con un carico collegato in uscita di 3.3k $\Omega$*



*Figura 12.8: Andamento nel tempo della tensione  $V_{shunt}$  quando l'uscita del DAC varia ogni 100ms di 1mV partendo da 1mV dopo aver mediato ogni 1000 campioni, con un carico collegato in uscita di 3.3k $\Omega$*

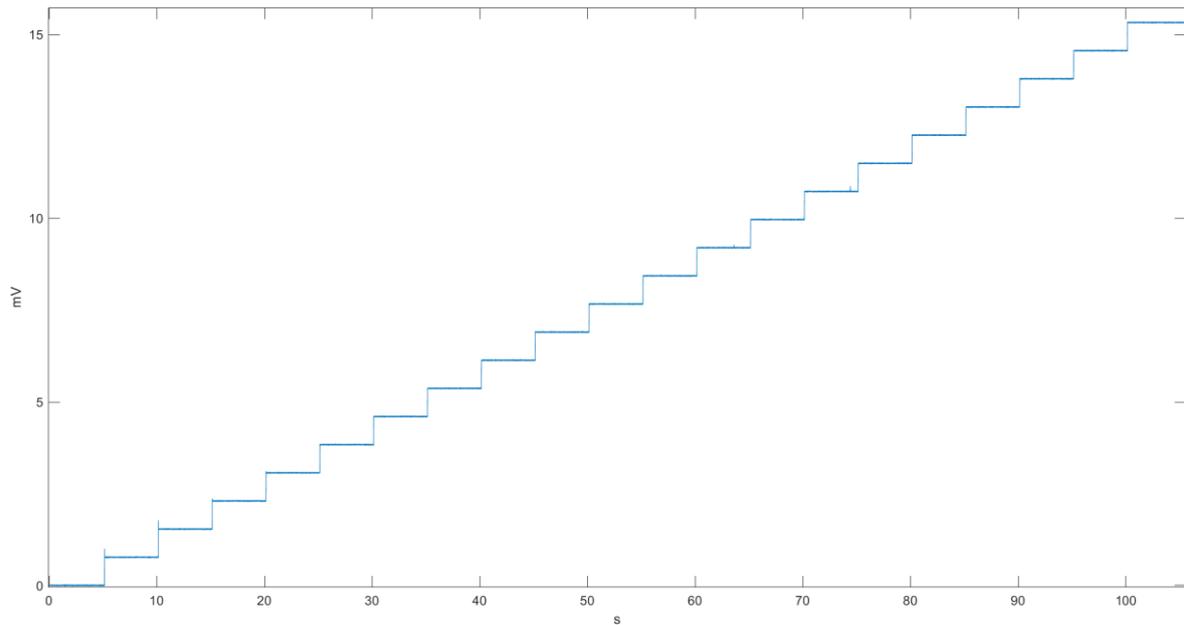
- 4) La tensione di uscita del DAC, partendo da 1mV, varia ogni secondo di 100mV fino ad arrivare ad un massimo di 5000mV, con un carico in uscita di 3.3k $\Omega$ . Si è voluto fare questo ulteriore test come controprova utilizzando un multimetro esterno e misurando anche da quest'ultimo la variazione di tensione nel tempo.

In figura 12.9 si possono osservare le tensioni misurate dall'ina e dal multimetro a confronto.



*Figura 12.9: Andamento nel tempo della tensione  $V_{bus}$  misurata dall'ina229 e da un multimetro esterno quando l'uscita del DAC varia ogni secondo di  $100mV$  partendo da  $1mV$ , con un carico collegato in uscita di  $3.3k\Omega$*

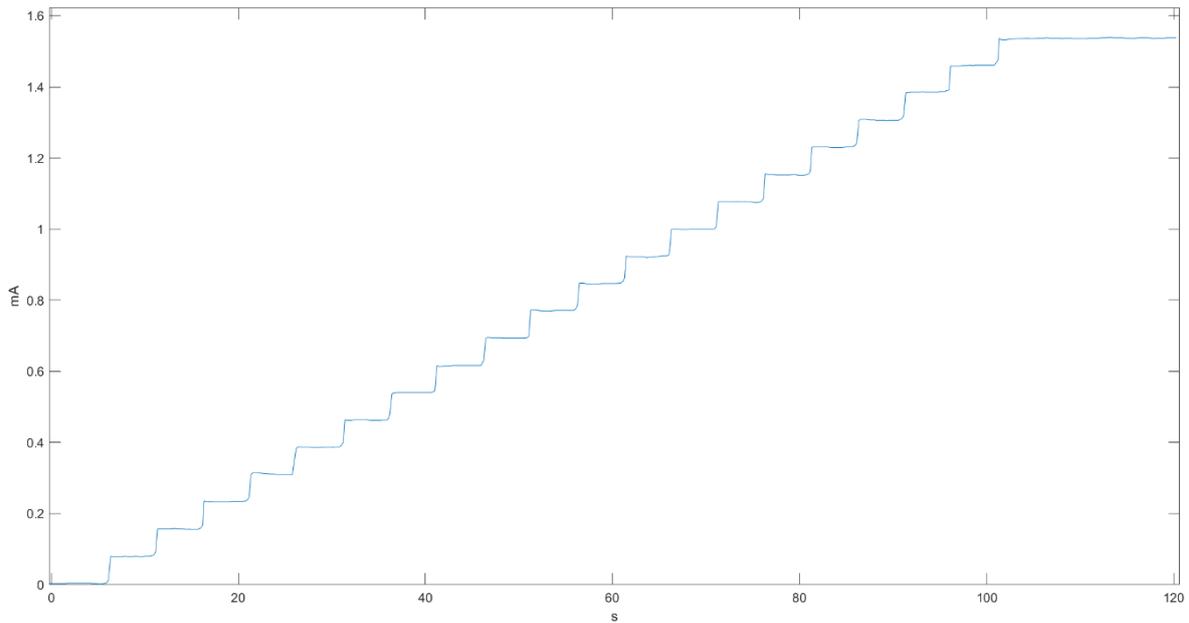
Si è voluto misurare anche la tensione sul resistore di shunt al variare del tempo, riportata in figura 12.10.



*Figura 12.10: Andamento nel tempo della tensione  $V_{shunt}$  misurata dall'ina229 quando l'uscita del DAC varia ogni secondo di 100mV partendo da 1mV, con un carico collegato in uscita di 3.3k $\Omega$*

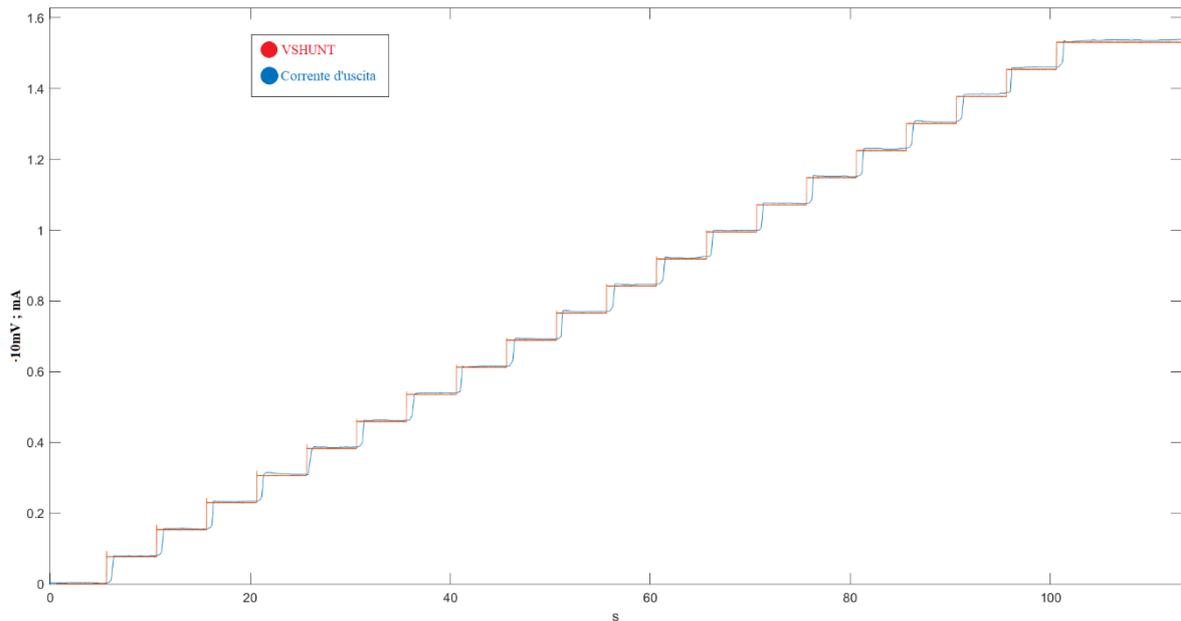
- 5) La corrente che scorre sul carico quando la tensione di uscita del DAC, partendo da 1mV, varia ogni secondo di 100mV fino ad arrivare ad un massimo di 5000mV, con un carico in uscita di 3.3k $\Omega$ .

I risultati sono stati graficati e sono riportati in figura 12.11.



*Figura 12.11: Andamento nel tempo della corrente d'uscita misurata dall'ina229 quando l'uscita del DAC varia ogni secondo di 100mV partendo da 1mV, con un carico collegato in uscita di 3.3k $\Omega$*

In figura 12.12 è riportata la variazione della corrente e della tensione di shunt nel tempo.



*Figura 12.12: Andamento nel tempo della corrente d'uscita e della tensione di shunt misurata dall'ina229 quando l'uscita del DAC varia ogni secondo di 100mV partendo da 1mV, con un carico collegato in uscita di 3.3k $\Omega$*

Utilizzando il multimetro si sono volute verificare le tensioni d'ingresso e d'uscita di alcuni componenti, come ad esempio quelle della pompa di carico, progettata per avere una tensione d'ingresso di +5V e d'uscita di -5V, ai suoi capi si è misurato +5,186V in ingresso e -5,166V in uscita.

Ai capi del booster, che è stato progettato per erogare in uscita una tensione di 7V, si è misurata una tensione di 6,99V.

Sul pin Vref del DAC, che dovrebbe erogare una tensione costante di 2,5V, si è misurata una tensione di 2,497V.

Per quanto riguarda i photorelay, quando venivano accesi si è misurata una caduta di potenziale di 1,7722V tra anodo e catodo, tensione che soddisfa le caratteristiche del componente in quanto sul datasheet è riportata una caduta di tensione tra 1,50V e 1,80V.

Quando erano spenti la caduta di potenziale sugli stessi capi era di 1V.

### 13. CONCLUSIONI E MIGLIORAMENTI

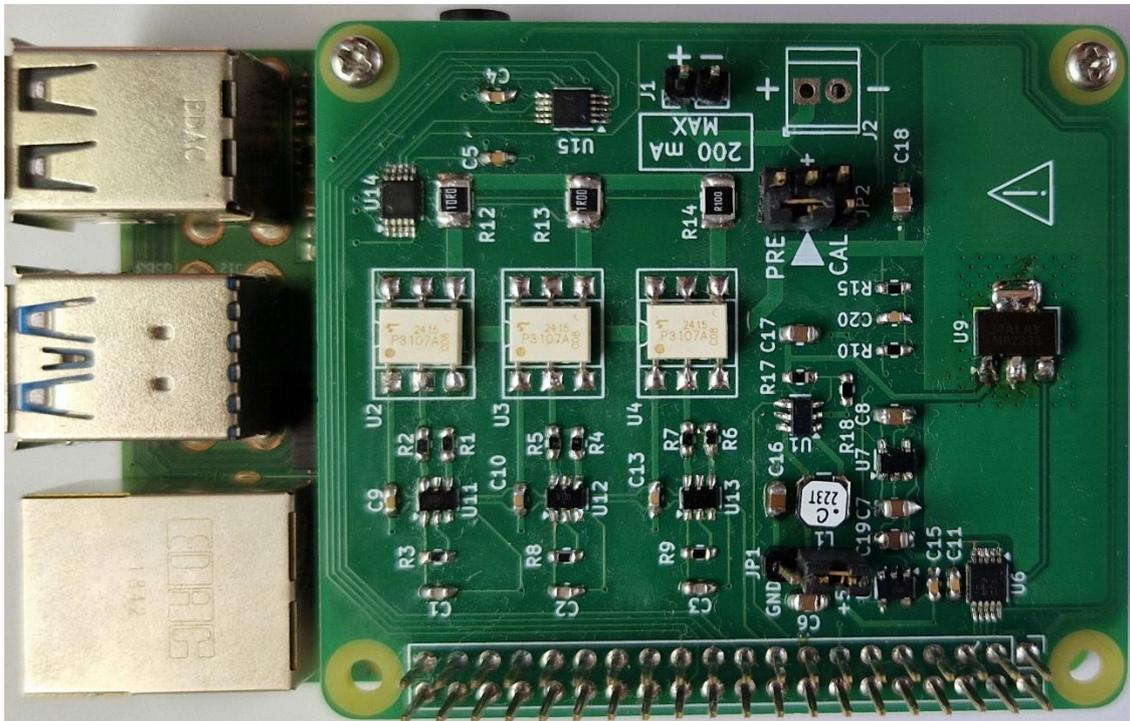
Questo progetto si è dimostrato utile per effettuare misurazioni della corrente che scorre sul carico, entro un massimo di 200mA. Tuttavia, per migliorare l'accuratezza delle misurazioni e sfruttare appieno le potenzialità dell'INA229, si potrebbero utilizzare i registri dedicati per impostare soglie minime e massime di tensione di shunt e di bus, nonché soglie massime di temperatura e di potenza. Inoltre, si potrebbe utilizzare il registro "CURRENT" per acquisire direttamente il valore della corrente in uscita, evitando così di doverla calcolare tramite la tensione di shunt. Un ulteriore miglioramento potrebbe consistere nello sfruttare la misura di temperatura interna per sviluppare ulteriori protezioni per il circuito.

Una problematica riscontrata nella fase di test è emersa quando la tensione d'uscita del DAC veniva abbassata troppo rapidamente, ad esempio passando da 5V a 1,2V. Questo comportamento causava un blocco dell'LT3082. Per ovviare a tale problematica, si è deciso di ridurre gradualmente la tensione ogni volta che era necessario.

Un piccolo errore in fase di progettazione da correggere, qualora si volesse realizzare nuovamente questo progetto, riguarda la connessione dei pin

dell'LT3082. Nello schema illustrato in Figura 9.1, il pin "IN" è stato erroneamente associato al pin 1 e il pin "SET" al pin 3, mentre in realtà è esattamente l'opposto. Fortunatamente, questo non ha rappresentato un problema significativo in fase di realizzazione poiché il componente è simmetrico; è stato sufficiente ruotarlo di 180° per saldarlo correttamente.

La Figura 13.1 mostra la realizzazione finale del progetto e la figura 13.2 mostra la BOM utile per effettuare l'ordine dei componenti esatti per la realizzazione del circuito.



*Figura 13.1 Foto raffigurante la scheda realizzata in laboratorio montata su una Raspberry Pi 4*

Indice	Quantità	Codice componente	Codice produttore	Descrizione
1	10	264-TLP3107A(TPFCT-ND	TLP3107A(TP,F	SSR RELAY SPST-NO 4A 0-60V
2	3	296-INA229AIDGSRCT-ND	INA229AIDGSR	IC CURR SENSE 1 CIRCUIT 10VSSOP
3	3	296-DAC80501ZDGSRCT-ND	DAC80501ZDGSR	SINGLE CHANNEL DAC 16BIT
4	3	296-53437-1-ND	TMUX1104DGSR	IC MUX 4:1 4OHM 10VSSOP
5	3	296-TPS2001EDBVRCT-ND	TPS2001EDBVR	IC
6	3	296-37679-1-ND	TPS60401DBVR	IC REG CHG PUMP INV 60MA SOT23
7	10	1727-7006-1-ND	74LVC2G06GV,125	IC INVERTER 2CH 2-INP 6TSOP
8	1	505-LT3082EST#PBF-ND	LT3082EST#PBF	IC REG LIN POS ADJ 200MA SOT223
9	1	296-47674-1-ND	TLV61046ADBVR	IC REG BOOST ADJ SOT23-6
10	3	445-175666-1-ND	CGA3E2C0G2A152J080AE	CAP CER 1500PF 100V C0G/NP0 0603
11	8	311-1344-1-ND	CC0603KRX7R9BB104	CAP CER 0.1UF 50V X7R 0603
12	5	445-6967-1-ND	CGA4J1X7R1E475K125AC	CAP CER 4.7UF 25V X7R 0805
13	1	445-5687-1-ND	CGA4J3X7R1E105K125AB	CAP CER 1UF 25V X7R 0805
14	1			C2.2uF
15	1	WM50016-06-ND	22284063	CONN HEADER VERT 6POS 2.54MM
16	1	A98333-ND	282834-2	TERM BLK 2P SIDE ENT 2.54MM PCB
17	1			JP1 Jumper_3_Bridged12
18	1			JP2 Jumper_6_Bridged
19	1	732-61304021821-ND	61304021821	WR-PHD 2.54 MM DUAL SOCKET HEADE (connettore rasp)
20	1	4652-LPS4018-223MRC-ND	LPS4018-223MRC	FIXED INDUCTOR 22UH 900MA 360MOHM SMD
21	3	541-390GCT-ND	CRCW0603390RJNEA	RES SMD 390 OHM 5% 1/10W 0603
22	3	541-56GCT-ND	CRCW060356R0JNEA	RES SMD 56 OHM 5% 1/10W 0603
23	3	541-150KHCT-ND	CRCW0603150KFKEA	RES 150K OHM 1% 1/10W 0603
24	2	541-4.7KGCT-ND	CRCW06034K70JNEA	RES SMD 4.7K OHM 5% 1/10W 0603
25	1	2019-SR732ERTTD10R0FCT-ND	SR732ERTTD10R0F	RES 10 OHM 1% 1/2W 1210
26	1	2019-SR732ERTTD1R00FCT-ND	SR732ERTTD1R00F	RES 1 OHM 1% 1/2W 1210
27	1	541-4390-1-ND	RCWE1210R100FKEA	RES 0.1 OHM 1% 1W 1210
28	1	541-309KHCT-ND	CRCW0603309KFKEA	RES SMD 309K OHM 1% 1/10W 0603
29	1	541-40.2KHCT-ND	CRCW060340K2FKEA	RES SMD 40.2K OHM 1% 1/10W 0603

*Figura 13.2: BOM dei componenti*

## 14. RINGRAZIAMENTI

Desidero esprimere la mia più sincera gratitudine a tutte le persone che hanno contribuito in maniera significativa al raggiungimento di questo importante traguardo.

In primo luogo, un ringraziamento speciale al Prof. Biagetti Giorgio, per la sua dedizione, la sua disponibilità e il costante supporto durante tutto il percorso di tirocinio e di tesi. Tutto ciò è stato fondamentale per il completamento di questo progetto e per la mia crescita personale e professionale.

Vorrei anche ringraziare profondamente la mia ragazza, P. . Il tuo supporto, il tuo incoraggiamento e l'essermi stata sempre al mio fianco mi hanno dato la forza di superare le avversità di questo lungo cammino. Grazie per aver creduto in me anche quando io stesso faticavo a farlo.

Infine, un ringraziamento dal profondo del cuore va alla mia famiglia. Ai miei genitori, per avermi dato la sicurezza di poter raggiungere qualsiasi obiettivo,

per avermi reso indipendente e per i sacrifici fatti per farmi arrivare fin qui. A mio fratello e sorella, per la loro vicinanza e il supporto.

Vorrei anche prendermi un momento per riconoscere il mio stesso impegno. A me stesso, grazie per la determinazione, la perseveranza e la dedizione dimostrate in questi anni. Grazie per non aver mai rinunciato, nonostante le difficoltà incontrate lungo il cammino. Questo traguardo è la dimostrazione di quanto il duro lavoro e la passione possano portare al raggiungimento degli obiettivi.

## 15. BIBLIOGRAFIA E SITOGRAFIA

[1] Zhihong Yu, “Understanding Linear Regulators and Their Key Performance Parameters”, Renesas Electronics 2022

[2] Texas Instrument INA229 datasheet , 12/2020  
<https://www.ti.com/product/INA229>

[3] <https://it.wikipedia.org/wiki/Multiplexer>

[4] Texas Instrument DAC161S055 datasheet, 01/2012  
<https://www.ti.com/product/DAC161S055>

[5] Texas Instrument DAC81401 datasheet, 05/2021  
<https://www.ti.com/product/DAC81401>

[6] Texas Instrument DAC80502 datasheet, 04/2020  
<https://www.ti.com/product/DAC80502>

[7] Maxim integrated MAX5705 datasheet, 11/2014  
<https://www.analog.com/en/products/max5705>

[8] Texas Instrument DAC80501 datasheet, 08/2023  
<https://www.ti.com/product/DAC80501>

[9] Toshiba “Application example of photorelay, replacing from Mechanical relay (Thermostat)” , 2019

[10] Toshiba TLP3106A datasheet, 05/2023

<https://toshiba.semicon-storage.com/ap-en/semiconductor/product/isolators-solid-state-relays/photorelay-mosfet-output/detail.TLP3106A.html>

[11] Toshiba TLP3107A datasheet, 05/2023

<https://toshiba.semicon-storage.com/ap-en/semiconductor/product/isolators-solid-state-relays/photorelay-mosfet-output/detail.TLP3107A.html>

[12] Linear Technology LT3082 datasheet, 2009

<https://www.analog.com/en/products/lt3082>

[13] Robin Mitchell “Build Your Own Negative Voltage Generator” ,  
06/2007 <https://www.allaboutcircuits.com/projects/build-your-own-negative-voltage-generator/>

[14] Texas Instrument TPS63710 datasheet, 06/2018

<https://www.ti.com/product/TPS63710>

[15] Texas Instrument LM7705 datasheet, 05/2018

<https://www.ti.com/product/TPS63710>

[16] Texas Instrument LM27761 datasheet, 01/2017

<https://www.ti.com/product/TPS63710>

[17] Texas Instrument TPS60401-Q1 datasheet, 10/2016

<https://www.ti.com/product/TPS60401-Q1>

[18] Texas Instrument TLV61046A datasheet, 02/2021

<https://www.ti.com/product/TLV61046A>

[19] Texas Instrument TPS2001D datasheet, 10/2017

<https://www.ti.com/product/TPS2001D>

[20] Raspberry Pi HAT+ Specification , 12/2020