

POLITECNICO DI MILANO

V Facoltà di Ingegneria

Corso di laurea in Ingegneria Elettronica
Dipartimento di Elettronica e Informazione



PROGETTO E REALIZZAZIONE
DI UN CAPACIMETRO CON MICROCONTROLLORE
PER FOTODIODI SPAD

Relatore: Prof. Alberto TOSI
Correlatore: Ing. Alberto DALLA MORA

Tesi di Laurea di:
Davide TAMBORINI Matr. 701956
Alessandro RUGGERI Matr. 702898

Anno Accademico 2008/2009

Indice

Elenco delle figure	IV
Introduzione	1
1 Il fotodiode SPAD	3
1.1 Fotorivelatori	3
1.1.1 Principi di funzionamento dei fotodiode	3
1.1.2 Il Single Photon Avalanche Diode	4
1.1.3 Modello SPAD	7
1.2 Circuito di spegnimento del fotorivelatore	8
1.3 Misura della capacit� di giunzione	13
1.3.1 Problematiche legate alla misurazione	14
2 “Capacimetro per diode polarizzati”	16
2.1 Requisiti dello strumento di misura	16
2.2 Soluzione circuitale	16
2.3 Analisi circuitale	18
2.3.1 Guadagno ideale	18
2.3.2 Guadagno reale	19
2.4 Osservazioni	20
3 SpadCapMeter	22
3.1 Storia ed evoluzione del progetto	22
3.1.1 Il PSoC TM	23
3.1.2 Rilevatore di picco con PSoC TM	24

3.1.3	Rilevatore di picco con Amplificatori Operazionali	25
3.2	Front-end analogico	28
3.2.1	Filtro d'ingresso	28
3.2.2	Comparatore	29
3.2.3	Ritardatore	32
3.3	Convertitore A/D	34
3.3.1	Requisiti di conversione	34
3.3.2	LTC1417A	35
3.3.3	Circuito di conversione	36
3.4	Microcontrollore	38
3.4.1	Il PIC18F4550	39
3.4.2	Configurazione hardware	45
3.5	Generatore sinusoidale	46
3.6	Alimentazione	51
3.7	Realizzazione progetto hardware	53
4	Software	60
4.1	Microcontrollore	60
4.1.1	Protocolli di comunicazione	61
4.1.2	Programma PIC	64
4.2	LabVIEW™	73
4.2.1	Introduzione a LabVIEW™	73
4.2.2	Programma LabVIEW™	74
5	Istruzioni per l'uso	79
5.1	Collegamento e driver	79
5.2	Come effettuare la misura	80
5.2.1	Taratura strumento	80
5.2.2	Cparassite	80
5.2.3	Acquisizione singolo valore	82
5.2.4	Rilevazione caratteristica $C_{ak}-V_a$	83

6	Misure sperimentali	85
7	Conclusioni e prospettive future	88
7.1	Conclusioni	88
7.2	Sviluppi futuri e prospettive	89
7.2.1	Alimentatore	89
	Bibliografia	92

Elenco delle figure

1	Evoluzione del capacimetro	2
1.1	Rappresentazione schematica di un APD e campo elettrico al suo interno	4
1.2	Processo di ionizzazione ad impatto e conseguente innesco della valanga	4
1.3	Confronto tra i principi di funzionamento degli APD e degli SPAD e caratteristica I-V	5
1.4	Modello base per SPAD	7
1.5	Circuito di spegnimento passivo	8
1.6	Caratteristica I-V dello SPAD e retta di carico del PQC	9
1.7	Circuito di spegnimento passivo con modello dello SPAD	9
1.8	Andamenti della corrente che attraversa il diodo (a) e della tensione ai suoi capi (b) dall'innesco dalla valanga al suo spegnimento. . .	10
1.9	Circuito per spegnimento passivo in Gated-Mode	11
1.10	Forma d'onda dell'uscita di timing (blu-continuo) e del segnale dummy che rappresenta la soglia di discriminazione dinamica (rosso-tratteggiato)	11
1.11	Segnale di valanga accoppiato capacitivamente (blu) e soglia minima di discriminazione statica del fronte (rosso).	12
1.12	Circuito per spegnimento passivo in Gated-Mode con prelievo differenziale	13
2.1	Circuito di principio "Capacimetro per diodi polarizzati"	17
2.2	Circuito "Capacimetro per diodi polarizzati"	18

2.3	Diagrammi di Bode del modulo di G_I , G_A e G_R all'aumentare di C_{AK}	21
3.1	Schema di principio del rilevatore di picco con PSoC™	24
3.2	Semplice rilevatore di picco diodo condensatore resistore	25
3.3	Rilevatore di picco con rete di ritardo	26
3.4	Rilevatore di picco sfasatore di 90°	27
3.5	Schema a blocchi del circuito di ingresso SPADCAPMETER	28
3.6	Filtro passa banda passivo	29
3.7	Comparatore	30
3.8	Monostabile con 555	32
3.9	Generatore di impulsi per trigger	33
3.10	Circuito per sfasatore digitale	34
3.11	Circuito Interno dell'LTC1417A	36
3.12	Generatore di impulsi per SOC	37
3.13	Circuito del convertitore Analogico/Digitale	38
3.14	Blocco USB del PIC18F4550	40
3.16	Moduli clock e PWM	43
3.17	Schema modulo SPI	44
3.18	Risposta in frequenza di un filtro passa banda e punti salienti	47
3.19	Filtro a reazione multipla generico	47
3.20	Filtro Passa Banda a reazione multipla generico	48
3.21	Schema attenuatore	50
3.22	Schema completo generatore 50 kHz	50
3.23	Schema pompa di carica	52
3.24	Filtro alimentazione operativa	53
3.25	Schermate dell'Oscilloscopio	55
3.26	Schema Orcad con PIC, USB, alimentatore e filtro 50kHz	56
3.27	Schema Orcad catena di acquisizione	57
3.28	Top del Layout	58
3.29	Assembly Top del Layout	58

3.30	Scheda realizzata vista da sopra	59
3.31	Scheda realizzata vista da davanti	59
4.1	Collegamento tra due periferiche SPI	63
4.2	Diagramma LabVIEW™ acquisizione singola misura	75
4.3	Diagramma LabVIEW™ misura capacità parassite	75
4.4	Diagramma LabVIEW™ taratura	77
5.1	Schermata taratura	81
5.2	Schermata capacità parassite	82
5.3	Schermata misura singola	83
5.4	Schermata misura caratteristica	84
6.1	Grafico della caratteristica dello SPAD al Silicio	86
6.2	Schermate durante la misura	87
7.1	Circuito per convertitore Step-Up	91

Introduzione

Lo SPAD è un fotorivelatore allo stato solido, basato su una giunzione p-n inversamente polarizzata, nel quale un singolo fotone riesce ad innescare una corrente di valanga. Per progettare circuiti che sfruttano tale device, è importante conoscere la capacità parassita che presenta tra anodo e catodo.

Esistono svariati strumenti in grado di misurare una capacità; sul mercato infatti sono presenti dei capacimetri in grado di effettuare misure su un ampio range di valori. Necessitando di una grande accuratezza questi strumenti sono molto costosi. Siccome la capacità dello SPAD è nell'ordine dei picofarad, le potenzialità di queste strumentazioni non è sfruttata appieno. Perciò è stato pensato di realizzare un capacimetro, con un'adeguata accuratezza, dedicato alla misura di capacità tipiche di tale fotodiodo.

Un primo approccio per raggiungere lo scopo è stato quello di costruire un circuito in grado di fornire una tensione di uscita proporzionale alla capacità in esame. Tale strumento, che chiameremo nel seguito della trattazione "Capacimetro per diodi polarizzati", tuttavia necessita di una notevole strumentazione di supporto, peraltro molto costosa, e fornisce dati da elaborare successivamente tramite un programma di calcolo. Si evince l'importanza di semplificare la struttura di acquisizione della misura in questione.

Partendo da questi presupposti, l'obiettivo di tale elaborato sarà quello di arrivare ad una configurazione in grado di collegarsi al "Capacimetro per diodi polarizzati" e, attraverso un opportuno software, visualizzare la capacità dello SPAD al PC.

Inoltre, si sa che tale valore è strettamente legato alla tensione di polarizzazione

del fotodiode ed è soggetto a parassitismi intrinseci al dispositivo. Perciò altre due specifiche richieste per il progetto, saranno la misura di tali parassitismi e la caratteristica completa capacità-tensione di polarizzazione.

In più, per garantire una misura sempre precisa, sarà necessario sviluppare un'apposita sezione del software per la taratura dello strumento. Tale opzione garantirà sempre di ottenere risultati accurati.

Illustriamo schematicamente qui di seguito la semplificazione della misura che si vuole ottenere con questo elaborato.

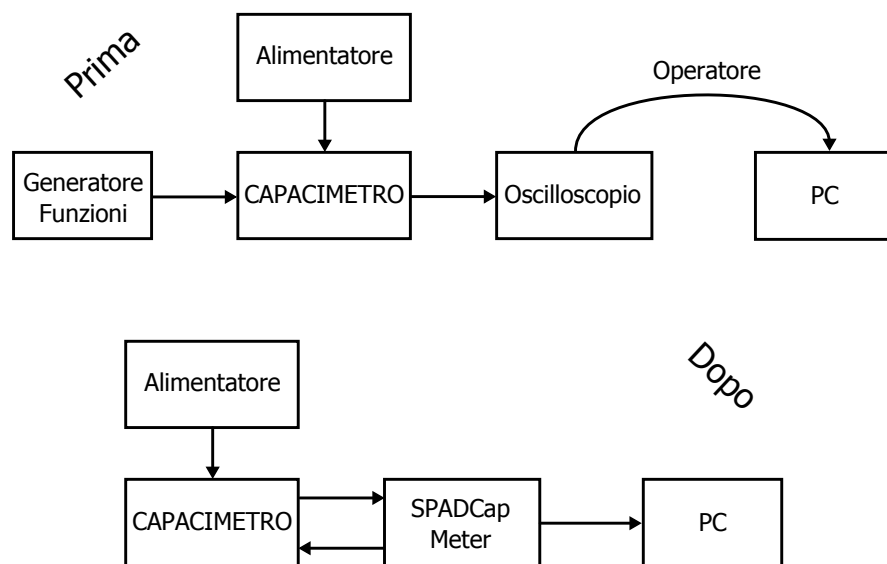


Figura 1: *Evoluzione del capacitometro*

Capitolo 1

Il fotodiode SPAD

Prima di iniziare a descrivere il progetto e lo sviluppo dello strumento di misura, è indispensabile focalizzare l'attenzione sui dispositivi a semiconduttore atti a convertire un segnale luminoso in uno elettrico.

1.1 Fotorivelatori

1.1.1 Principi di funzionamento dei fotodiode

Quando un fotone incide su un semiconduttore se ha un'energia $E \geq E_{GAP}$ allora genera una coppia elettrone lacuna libera[4]. Se tali portatori si trovano in una regione di svuotamento, il campo elettrico presente tende a far derivare le cariche; ciò si traduce in una corrente esterna al dispositivo proporzionale al numero di fotoni incidenti. Questa situazione si ottiene in un comune fotorivelatore $p-n$. L'efficienza quantica del dispositivo, cioè il rapporto tra il numero di fotoni incidenti e il numero di impulsi ottenuti in uscita, è proporzionale alla lunghezza della zona svuotata; per aumentarla è possibile applicare un campo elettrico molto elevato oppure drogare pesantemente le due regioni.

Una soluzione migliore consiste nell'introdurre una regione intrinseca tra le regioni P ed N: in questo modo è possibile estendere la regione svuotata indipendentemente dai drogaggi e dal campo elettrico applicato. Questo dispositivo prende il nome di fotodiode PIN.

Un'ulteriore modifica consiste nell'introdurre una regione (più sottile delle altre) fortemente drogata come in Fig. 1.1. Nonostante la tensione applicata sia

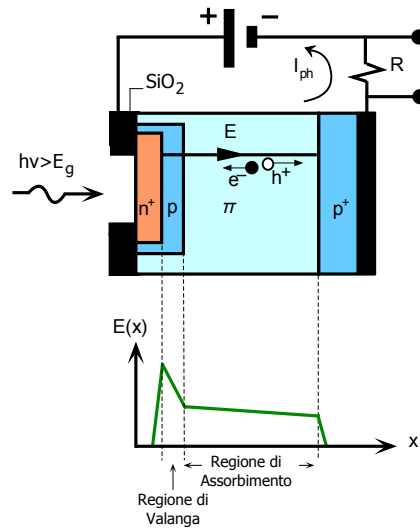


Figura 1.1: Rappresentazione schematica di un APD e campo elettrico al suo interno

inferiore a quella di breakdown, nei pressi della giunzione n^+p il campo elettrico è così elevato da accelerare gli elettroni fotogenerati sino al raggiungimento di un'energia tale da liberare nuovi portatori e quindi innescare un processo di moltiplicazione a valanga (vedi Fig. 1.2). Da ciò il nome di *Avalanche Photo-Diode* (APD). In questo modo la corrente generata è molto maggiore rispetto a quella di un PIN.

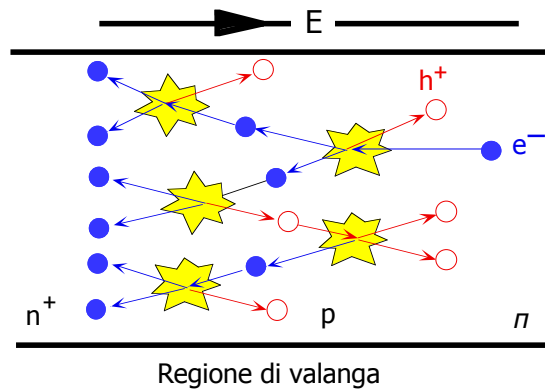


Figura 1.2: Processo di ionizzazione ad impatto e conseguente innesco della valanga

1.1.2 Il Single Photon Avalanche Diode

Uno SPAD è un tipo di fotodiode a valanga progettato per essere polarizzato al di sopra della tensione di breakdown, anziché al di sotto come nei classici APD.

Il superamento di questo limite fa sì che il fotone incidente sull'area attiva anziché dare origine ad una valanga che spontaneamente si esaurisce all'esaurimento del flusso fotonico, dà origine ad una valanga di tipo rigenerativo, ovvero con la proprietà di autosostenersi per un tempo indeterminato in mancanza di uno spegnimento esterno del rivelatore (Fig. 1.3).

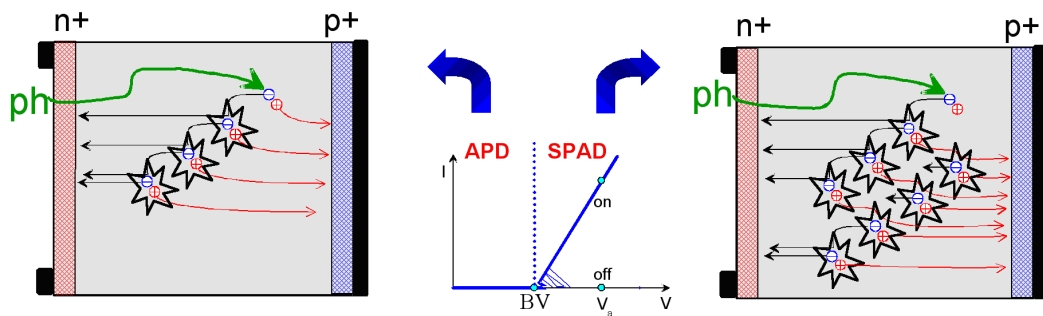


Figura 1.3: Confronto tra i principi di funzionamento degli APD e degli SPAD e caratteristica I-V

Il singolo fotone assorbito attiva una corrente nel dispositivo che viene limitata esternamente a qualche milliampère essendo il guadagno teoricamente infinito. A causa di ciò si perde la linearità in quanto una valanga prodotta da un solo fotone o da più fotoni sono, a livello macroscopico, assolutamente identiche. Occorre pertanto assicurarsi che il tasso di assorbimento di fotoni sia basso per evitare l'assorbimento di più fotoni contemporaneamente.

Gli SPAD in silicio possono essere suddivisi principalmente in due tipologie a seconda della zona svuotata della giunzione p-n. Esistono dispositivi “sottili” (*thin*) in cui essa non oltrepassa $1\ \mu\text{m}$ o “spessi” (*thick*) con zone di carica spaziale da $20\ \mu\text{m}$ fino a $150\ \mu\text{m}$. Le caratteristiche più importanti di uno SPAD di tipo *thin* sono:

- tensione di breakdown da 10 V a 50 V;
- area attiva “piccola” da $5\ \mu\text{m}$ a $100\ \mu\text{m}$ di diametro;
- buona efficienza quantica nel visibile (45% circa a 500 nm, 32% a 630 nm, 15% a 730 nm) e sufficiente nel vicino infrarosso (10% circa a 830 nm, 0,1% a 1064 nm).

Per quanto riguarda lo SPAD thick, invece, si tratta di dispositivi con:

- tensione di breakdown da 200 V a 500 V;
- area attiva “grande”, da 100 μm a 500 μm di diametro;
- ottima efficienza quantica nel visibile (pressoché costante al 50% a partire da 540 nm fino a 850 nm) e buona nell’infrarosso (circa 3% a 1064 nm).

Tali dispositivi in silicio divengono praticamente inutilizzabili a lunghezze d’onda superiori a 1,1 μm , escludendo l’importante terza finestra di trasmissione in fibra ottica.

Sempre per quanto concerne questi tipi di SPAD, l’innesco della valanga, dovuto al campo elettrico di svuotamento superiore al limite di 300 kV/cm, grazie alla polarizzazione ad una generica $V_a \geq V_B$ (tensione di breakdown), è un procedimento molto rapido, con un tempo di salita del fronte di corrente dell’ordine dei nanosecondi o anche solo di qualche centinaio di picosecondi. Tale fronte fornisce informazione sull’istante d’arrivo del fotone con pochi picosecondi di jitter. La corrente tuttavia continua a fluire nel dispositivo a meno di non ridurre la polarizzazione V_a a valori inferiori a V_B , laddove il funzionamento dello SPAD diventa il medesimo di un normale APD, al fine di consentirne lo spegnimento spontaneo in preparazione di un fotone successivo.

Negli SPAD non solo i fotoni assorbiti innescano la valanga, ma anche i portatori generati termicamente, per cui il fotodiode produce impulsi di corrente anche in assenza di illuminazione; tali impulsi sono chiamati conteggi di buio. Questo fenomeno è visibile anche nella caratteristica in Fig. 1.3 dove si notano dei repentini salti tra le due curve. La loro fluttuazione statistica (legge Poissoniana) rappresenta la sorgente interna di rumore del diode. Il tasso dei conteggi di buio aumenta al crescere della temperatura di lavoro del fotorivelatore e della sovratensione V_e , ovvero della differenza tra la tensione di polarizzazione e quella di breakdown. La sovratensione, inoltre, influenza il variare di due importanti parametri tipici dello SPAD, ovvero l’efficienza quantica e la risoluzione temporale,

ove per risoluzione temporale si intende la distribuzione statistica dei ritardi tra il vero istante di arrivo del fotone e l'attimo in cui è rivelato l'impulso di corrente.

Per la precisione, si dimostra che l'aumento della sovratensione migliora le prestazioni in termini di efficienza quantica e risoluzione temporale, ma peggiora quelle relative al rumore, determinando un aumento del numero di conteggi di buio. Tenendo conto di tutti questi fattori, si può ricavare un valore ottimale di V_e che deve essere mantenuto costante per ottenere le prestazioni migliori del fotodiode.

1.1.3 Modello SPAD

Viene ora proposto un modello circuitale per lo SPAD, ai fini di comprenderne meglio il suo funzionamento elettrico [2]. Come è possibile osservare in Fig. 1.4, l'arrivo di un fotone è modellizzato mediante la chiusura di un interruttore, il quale permette alla corrente di fluire attraverso il dispositivo. In serie a questo vi è R_D che è data dalla somma della resistenza di carica spaziale e della resistenza della zona neutra; il generatore di tensione V_B simula la tensione di breakdown del dispositivo.

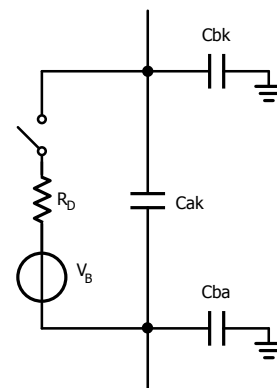


Figura 1.4: Modello base per SPAD

Siccome la rivelazione dei fotoni deve avvenire in tempi rapidi, le piccole capacità interne del dispositivo risultano molto importanti; perciò si individuano C_{ak} , che è la capacità della giunzione, inoltre C_{ba} e C_{bk} sono le capacità parassite tra anodo e catodo ed il substrato.

Esistono tuttavia modelli molto più complessi rispetto a quello qui illustrato, che rappresentano meglio la realtà, simulando il trigger, l'autosostenimento e l'autospegnimento della corrente fotogenerata mediante interruttori controllati. Per lo scopo del presente elaborato essi non sono necessari in quanto si è principalmente interessati alle capacità del dispositivo.

1.2 Circuito di spegnimento del fotorivelatore

Dal principio di funzionamento appena illustrato, si evince che per poter estrarre delle informazioni utili dallo SPAD, bisogna inserirlo in un apposito circuito che:

1. rilevi il fronte di salita della corrente di valanga;
2. generi un impulso in uscita sincronizzato con il fronte;
3. spenga la valanga, facendo scendere la tensione ai capi del diodo al di sotto della tensione di breakdown;
4. rimetta lo SPAD in condizioni di rilevare un nuovo fotone incidente, rialzando la tensione di bias agli opportuni valori superiori a V_B .

Questo circuito è usualmente chiamato *Quenching Circuit*; le caratteristiche di questo circuito influenzeranno molto le condizioni del fotorivelatore e quindi le sue performances[1].

Il più semplice circuito di controllo di un fotorivelatore è detto circuito di spegnimento passivo (*Passive Quenching Circuit*). Un esempio di tale circuito è mostrato in Fig. 1.5.

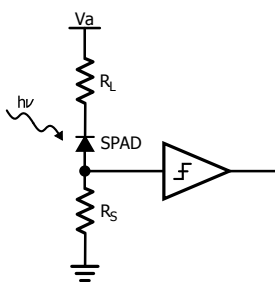


Figura 1.5: Circuito di spegnimento passivo

Lo SPAD è inversamente polarizzato da V_a attraverso una R_L di elevato valore (100 k Ω o più) al fine di limitare la corrente di valanga; questa normalmente è nulla fino a quando non arriva un fotone che innesca il processo di moltiplicazione e la fa salire rapidamente sino a qualche milliampère. Questa viene rivelata da R_S , resistenza di alcune decine di Ω , che la converte in tensione e permette quindi la lettura mediante un

comparatore veloce. I due stati della corrente appena descritti sono facilmente visualizzabili in Fig. 1.6 che rappresenta la caratteristica corrente-tensione tipica di uno SPAD polarizzato inversamente (dove I_D è la corrente che attraversa il diodo e V_D è d.d.p. tra catodo e anodo). Insieme a tale caratteristica viene tracciata la retta di carico per visualizzare graficamente quanto sta accadendo.

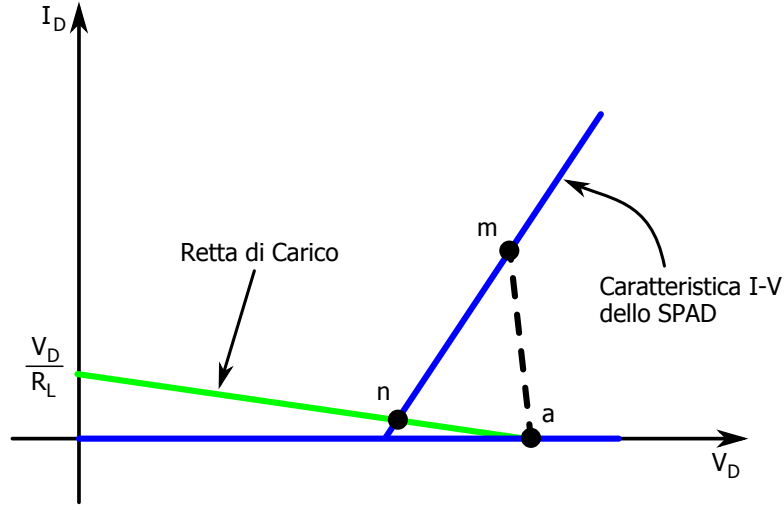


Figura 1.6: Caratteristica I-V dello SPAD e retta di carico del PQC

A riposo $I_D = 0$ e quindi lo SPAD è nel punto (a); in seguito all'arrivo del fotone si passa repentinamente in (m). L'elevata corrente che circola va a scaricare le capacità e I_D e V_D decrescono esponenzialmente al valore asintotico posto in (n). La corrente che attraversa il fotodiiodo nel tempo che va dall'inizio della valanga fino al valore asintotico è dato da

$$I_D(t) = \frac{V_D(t) - V_B}{R_D} = \frac{V_e(t)}{R_D} \quad (1.1)$$

Un esempio di questa scarica è dato in Fig. 1.8. Il valore di picco della corrente e il suo valore asintotico I_{fin} posso essere ricavati osservando il circuito equivalente in Fig. 1.7:

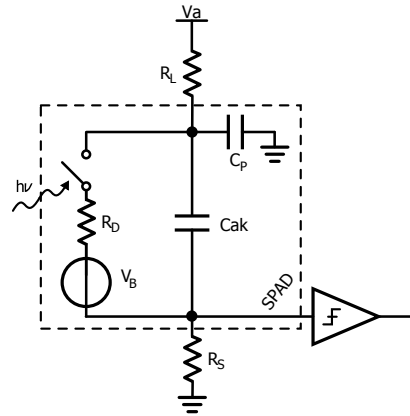


Figura 1.7: Circuito di spegnimento passivo con modello dello SPAD

$$I_D(0^+) = \frac{V_a - V_B}{R_D} \quad (1.2)$$

$$I_{fin} = \frac{V_a - V_B}{R_D + R_L} \simeq \frac{V_e}{R_L} \quad (1.3)$$

Il valore iniziale della tensione V_D è V_a , mentre il valore asintotico finale è:

$$V_{fin} = V_B + I_{fin} \cdot R_D \quad (1.4)$$

Il tempo impiegato dal sistema per portarsi su questi livelli di regime (*quenching time constant*) è data da:

$$\tau_q = (C_{ak} + C_p) \cdot R_L || R_D \simeq (C_{ak} + C_p) \cdot R_D \quad (1.5)$$

Per correnti $I_{fin} \geq 100 \mu\text{A}$ la probabilità che il processo di moltiplicazione a valanga si interrompa è molto basso. Scendendo sotto tale valore però, dato lo scarso numero di portatori, questo può succedere; in tal caso si ha una lenta ricarica delle C_{ak} e C_p sino a quando la tensione ai capi dello SPAD non raggiunge V_a . Questa costante di tempo (*recovery time constant*) è stimabile come:

$$\tau_r \simeq R_L \cdot (C_{ak} + C_p) \quad (1.6)$$

Il segnale che pilota il comparatore ha la stessa forma dell'onda dell'impulso di valanga ed il picco ha il seguente valore

$$V = \frac{V_a - V_B}{R_D} \cdot \frac{C_p}{C_p + C_{ak}} \cdot R_S \quad (1.7)$$

Si nota che esso è maggiore se $C_p \geq C_{ak}$; difatti solo la corrente proveniente da C_{ak} circola nell'anello interno al diodo. In questo caso V è nell'ordine di qualche mV.

In varie applicazioni è necessario utilizzare fotorivelatori molto sensibili sotto il controllo di un segnale di abilitazione (*Gated Mode*). Nel caso dello SPAD, ciò è facilmente ottenibile applicando un segnale di tipo impulsivo con ampiezza sufficientemente maggiore alla tensione di breakdown; questo tipo di comando è anche molto utile in quanto se lo SPAD è tenuto spento per un tempo sufficientemente lungo si hanno dei netti miglioramenti nelle sue prestazioni una volta riattivato.

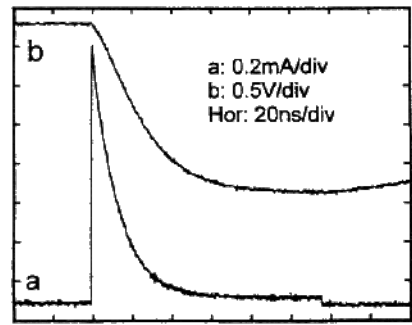


Figura 1.8: Andamenti della corrente che attraversa il diodo (a) e della tensione ai suoi capi (b) dall'innesco dalla valanga al suo spegnimento.

Una delle più semplici configurazioni circuitali adottate per questo scopo è mostrata in figura Fig. 1.9

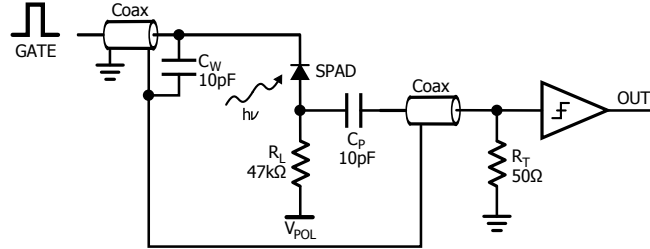


Figura 1.9: Circuito per spegnimento passivo in Gated-Mode

Questo circuito presenta un grosso problema in quanto ad accoppiarsi capacitivamente non è solo il segnale di valanga ma anche quello di gate che, sia sul fronte di salita che su quello di discesa, produce una variazione

$$\Delta V_{anode} = V_G \cdot \frac{C_p}{C_p + C_{ak}} \quad (1.8)$$

nella quale con V_G s'intende l'ampiezza effettiva del segnale di gate al catodo, C_{ak} è la capacità del diodo in inversa e C_p quella di prelievo. Il segnale di abilitazione, proveniente da un impulsatore con bassa impedenza di uscita, subisce dunque una riflessione a causa della grande impedenza vista al catodo, generando un'onda riflessa di pari ampiezza rispetto a quella incidente, raddoppiando quindi la tensione vista dallo SPAD. L'onda riflessa è poi terminata all'impulsatore senza dare luogo ad ulteriori riflessioni che potenzialmente altererebbero la sovratensione. La variazione della tensione anodica a causa della partizione capacitiva induce, sul terminale della capacità di prelievo verso il coassiale, dei picchi spuri, come visibile in Fig. 1.11.

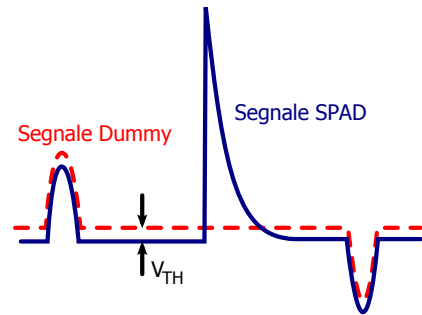


Figura 1.10: Forma d'onda dell'uscita di timing (blu-continuo) e del segnale dummy che rappresenta la soglia di discriminazione dinamica (rosso-tratteggiato)

Più il segnale di abilitazione è veloce, più l'ampiezza dei disturbi aumenta. Non si può semplicemente rendere il segnale di gate più lento in quanto è un requisito fondamentale e nemmeno alzare la soglia di comparazione in quanto

significherebbe rallentare la rilevazione della valanga e quindi il tempo di risposta totale del circuito.

L'idea innovativa introdotta consiste nella generazione di una soglia dinamica che possa seguire l'andamento degli impulsi di accoppiamento per evitare di sincronizzarsi su di essi, mentre permetta di mantenere un livello di soglia basso quando si deve leggere il segnale dovuto alla valanga. Al fine di ottenere un livello di soglia indipendente dall'ampiezza dell'impulso di accoppiamento del segnale di gate occorre generare un segnale aggiuntivo, chiamato *dummy*, che presenti unicamente gli impulsi di accoppiamento senza il segnale di valanga, come rappresentato in Fig. 1.10. Per tale scopo è necessario inserire in parallelo allo SPAD una rete che si comporta in modo analogo ad esso, ottenendo un ramo speculare. L'uscita qui prelevata rappresenta il corretto input della soglia dinamica interessata. Ciò si traduce nel circuito illustrato in Fig. 1.12

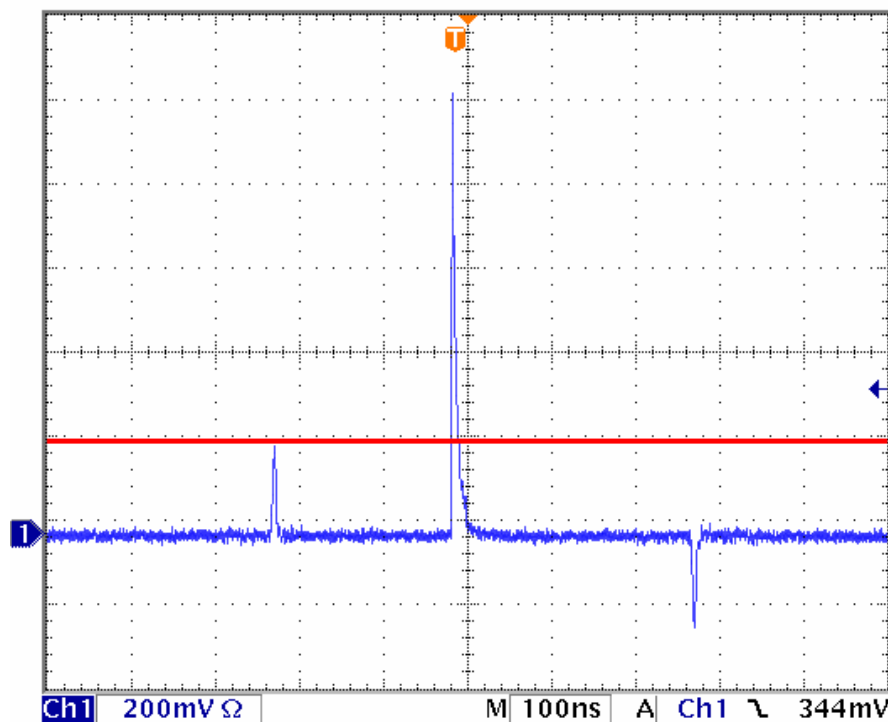


Figura 1.11: Segnale di valanga accoppiato capacitivamente (blu) e soglia minima di discriminazione statica del fronte (rosso).

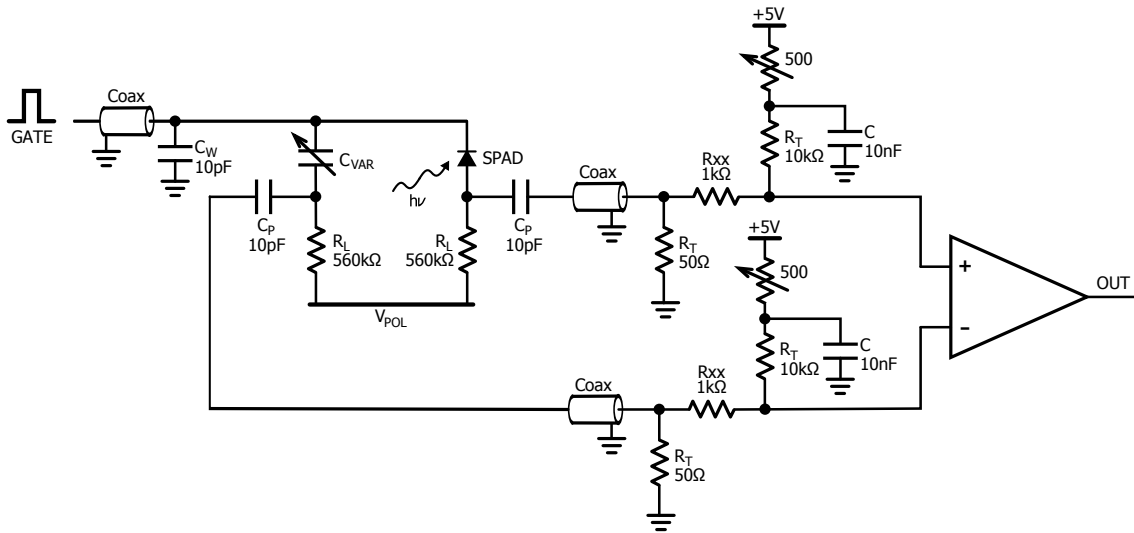


Figura 1.12: Circuito per spegnimento passivo in Gated-Mode con prelievo differenziale

La capacità variabile C_{var} deve essere regolata in modo tale da essere uguale alla capacità del diodo in inversa alla tensione di polarizzazione V_{pol} di utilizzo, per ottenere il dummy signal. Con questo tipo di circuito, l'arrivo del segnale di gate non produce alcuno scatto indesiderato del comparatore, mentre, all'innesco di una valanga da parte dello SPAD, la linea di dummy rimane ferma e la tensione del segnale generato dal fotodiode attraversa la soglia producendo lo scatto. Per produrre una replica perfetta dell'impulso spurio dovuto all'accoppiamento capacitivo sull'uscita di timing del segnale di gate è necessaria la conoscenza delle costanti di tempo in gioco. In particolare è necessario conoscere la capacità dello SPAD al variare della tensione di polarizzazione in quanto può essere necessario polarizzarlo a differenti tensioni al variare della temperatura, essendo la tensione di breakdown del dispositivo linearmente dipendente da quest'ultima.

1.3 Misura della capacità di giunzione

L'analisi fatta nel paragrafo precedente porta ad una conclusione: per valutare le applicazioni del fotorivelatore SPAD pilotato da un circuito di spegnimento, è necessario conoscere alcuni valori caratteristici del circuito. Alcuni di questi, tra cui le costanti di tempo in gioco, sono strettamente dipendenti dalla capacità di

giunzione C_{ak} . Ne deriva che non è sufficiente attribuire ad essa un valore medio (stimato, come già detto, intorno ad 1 pF), ma è indispensabile misurarla con precisione per ogni diodo. La determinazione esatta della C_{ak} , oltre ad essere richiesta per uno studio puramente teorico, permette anche di predisporre un modello che simuli correttamente il circuito considerato attraverso strumenti informatici.

1.3.1 Problematiche legate alla misurazione

La precisione nella misura della capacità di giunzione di uno SPAD è limitata da due principali fattori: il valore piccolo della capacità stessa e la sua dipendenza della tensione ai capi del diodo. Pur avendo stimato il valore della C_{ak} , c'è da sottolineare che tale valore può variare sia a seconda del dispositivo fisico in esame, che al variare della tensione di polarizzazione. Si è visto come, per mantenere costanti e ottimali le prestazioni dello SPAD in termini di efficienza quantica, risoluzione temporale e rumore, sia necessario tenere fissa su un valore ben preciso la sovratensione applicata. Per far ciò si deve però considerare che la tensione di breakdown non rimane invariata in qualsiasi condizione operativa, ma è fortemente dipendente dalla temperatura: il valore del coefficiente termico è legato alla struttura dello SPAD ed è attorno allo 0,3%/K. Per ottenere prestazioni simili a temperatura diversa occorre dunque modificare la tensione di polarizzazione: ciò determina una variazione dello spessore della regione di svuotamento del diodo (W), e quindi della capacità di giunzione che è legata ad esso dalla relazione:

$$C_D = \frac{\varepsilon A}{W} \quad (1.9)$$

dove A è la sezione trasversale della zona di svuotamento ed ε è la costante dielettrica del semiconduttore utilizzato. Segue che, per uno stesso dispositivo, è necessario effettuare più misure con differenti valori di polarizzazione.

Si deve prestare attenzione al fatto che misurando la capacità dello SPAD, leggendo semplicemente il valore fornito tra anodo e catodo del dispositivo, non si ottiene la misura desiderata, cioè C_{ak} . Questo è dovuto alla presenza di due capacità parassite tra i contatti anodo-substrato e catodo-substrato come precedentemente illustrato in Fig. 1.4.

La capacità misurata tra anodo e catodo (C_{AK}) è dunque:

$$C_{AK} = C_{ak} + \frac{C_{ba} \cdot C_{bk}}{C_{ba} + C_{bk}} \quad (1.10)$$

Analogamente si possono ricavare le formule di ciò che si ottiene misurando C_{BA} e C_{BK} . Unendo queste informazioni si arriva alla scrittura del seguente sistema:

$$\begin{cases} C_{AK} = C_{ak} + \frac{C_{ba} \cdot C_{bk}}{C_{ba} + C_{bk}} \\ C_{BA} = C_{ba} + \frac{C_{ak} \cdot C_{bk}}{C_{ak} + C_{bk}} \\ C_{BK} = C_{bk} + \frac{C_{ba} \cdot C_{ak}}{C_{ba} + C_{ak}} \end{cases} \quad (1.11)$$

la cui risoluzione fornisce C_{ak} , C_{bk} e C_{ba} ovvero gli effettivi valori della capacità di svuotamento e delle due capacità parassite all'interno dello SPAD.

E' importante notare che solo la capacità C_{ak} cambia al variare della tensione di polarizzazione, mentre le capacità parassite sono indipendenti da essa.

Capitolo 2

“Capacimetro per diodi polarizzati”

2.1 Requisiti dello strumento di misura

Da quanto illustrato nel capitolo precedente si ricava che un capacimetro adatto per SPAD deve:

- essere in grado di misurare con un’alta precisione le capacità tipiche che assume il dispositivo; per essere sicuri si prende un intervallo da 0,1 pF a qualche decina di pF.
- rispettare dei criteri progettuali rivolti a minimizzare i parassitismi capacitivi interni.
- consentire la corretta polarizzazione dello SPAD con una tensione che possa essere variata senza modificare per altri aspetti il funzionamento del circuito.

Partendo dalle considerazioni fin qui viste si può semplificare il problema cercando di realizzare un circuito che fornisca in uscita una grandezza elettrica, preferibilmente una tensione, per la sua semplicità di elaborazione, che sia strettamente legata al valore della capacità interessata[3].

2.2 Soluzione circuitale

Per fare questo è comodo utilizzare una configurazione circuitale ad operazionali simile ad un amplificatore invertente; siccome si ha a che fare con una capacità, si

devono sostituire le usuali resistenze con condensatori. In questo modo il guadagno dello stadio dipende direttamente dal valore della capacità in esame. Tutto ciò a patto di iniettare all'ingresso un segnale di stimolo variabile nel tempo (per esempio una sinusoide) ad un'adeguata frequenza.

Si ottiene quindi il circuito riportato in Fig. 2.1, dove si può studiare semplicemente il guadagno.

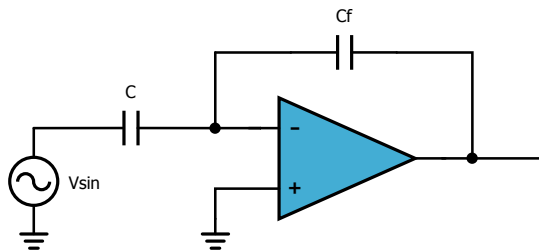


Figura 2.1: Circuito di principio "Capacimetro per diodi polarizzati"

$$G = \frac{V_o}{V_i} = -\frac{Z_f}{Z} = -\frac{\frac{1}{sC_f}}{\frac{1}{sC}} = -\frac{C}{C_f} \quad (2.1)$$

Da qui si evince che fissando opportunamente C_f e sostituendo lo SPAD a C si ottiene una tensione di uscita direttamente proporzionale alla capacità del fotodiode.

Tutto questo è valido usando un operazionale ideale; purtroppo nella realtà non si può fare totale affidamento a questo semplice modello. Innanzitutto esiste una corrente fuoriuscente dal morsetto non invertente dell'amplificatore, comunemente chiamata corrente di *bias*; questa influenza negativamente il corretto funzionamento del circuito in quanto, caricando costantemente la capacità C_f , fa saturare l'uscita. Per ovviare a tale problema è necessario aggiungere una resistenza R_f nella rete di reazione.

Come precedentemente affermato, si deve misurare la capacità dello SPAD al variare della tensione di polarizzazione. E' necessario quindi utilizzare un generatore di tensione continua per fornire la tensione di *bias* al catodo del fotodiode. Questa non può essere applicata direttamente al catodo del fotodiode in quanto ad esso arriva anche il segnale sinusoidale in ingresso; Perciò si usa una resistenza

R_p , con l'ulteriore compito di proteggere il generatore di bias da una eventuale corrente di cortocircuito. Ciò nonostante questa tensione essendo continua non viene trasferita in uscita, per via della C_{AK} (cioè la capacità dello SPAD che sostituisce il condensatore C). Per finire viene introdotta una capacità di disaccoppiamento C_p tra il segnale sinusoidale e il catodo dello SPAD.

Di conseguenza il circuito completo è illustrato in Fig. 2.2.

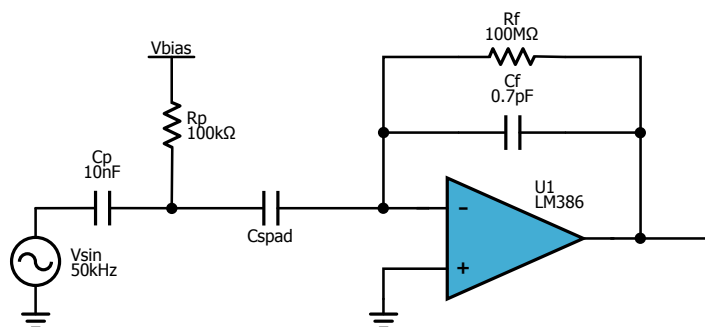


Figura 2.2: Circuito “Capacimetro per diodi polarizzati”

2.3 Analisi circuitale

Appurato che questa soluzione circuitale ha tutte le caratteristiche per fornire in uscita una tensione con le caratteristiche richieste, è opportuno uno studio dettagliato per verificare le proprietà del circuito in Fig. 2.2.

2.3.1 Guadagno ideale

Considerando a massa virtuale il terminale negativo dell'operazionale, si può dire che il generatore v_{in} genera una corrente pari a

$$i_{in} = \frac{V_{in}}{\frac{1}{sC_p} + R_p \parallel \frac{1}{sC_{AK}}}$$

della quale scorre nel ramo di retroazione la frazione

$$i_f = \frac{V_{in}}{\frac{1}{sC_p} + R_p \parallel \frac{1}{sC_{AK}}} \cdot \frac{R_p}{R_p + \frac{1}{sC_{AK}}}$$

questa moltiplicata per l'impedenza in retroazione $Z_f = R_f \parallel \frac{1}{sC_f}$, fornisce la tensione in uscita con segno opposto.

Di conseguenza il guadagno ideale del circuito è:

$$G_I = \frac{\left(R_f \parallel \frac{1}{sC_f}\right) \cdot R_p}{\left(\frac{1}{sC_p} + R_p \parallel \frac{1}{sC_{AK}}\right) \cdot \left(R_p + \frac{1}{sC_{AK}}\right)} = \frac{sC_p R_p \cdot sC_{AK} R_f}{[1 + sR_p(C_p + C_{AK})] \cdot (1 + sC_f R_f)} \quad (2.2)$$

Si nota che sono presenti due zeri nell'origine; considerando che $C_{AK} \ll C_p$, si ricava che la capacità di giunzione può essere trascurata nel calcolo dei poli, che hanno quindi in qualunque caso le costanti di tempo $\tau_{p1} = C_p R_p = 1 \text{ ms}$ e $\tau_{p2} = C_f R_f = 70 \text{ s}$, e frequenze $f_{p1} = 160 \text{ Hz}$ e $f_{p2} = 2,27 \text{ kHz}$.

La capacità C_{AK} influisce però sul valore del guadagno ad alta frequenza, che è:

$$G_{I,HF} = -\frac{C_{AK}}{C_f} \quad (2.3)$$

2.3.2 Guadagno reale

Ovviamente, come dice la parola stessa, il guadagno ideale non spiega esattamente le cose come sono nella realtà. Per determinare l'effettivo guadagno dello stadio bisogna innanzitutto individuare il guadagno d'anello:

$$\begin{aligned} G_L &= -A(s) \cdot \frac{\frac{1}{sC_{AK}} + R_p \parallel \frac{1}{sC_p}}{\frac{1}{sC_{AK}} + R_p \parallel \frac{1}{sC_p} + R_f \parallel \frac{1}{sC_f}} = \\ &= \frac{-A(s) \cdot [1 + sR_p(C_p + C_{AK})] \cdot (1 + sR_f C_f)}{1 + s[R_p(C_p + C_{AK}) + R_f(C_f + C_{AK})] + s^2 R_f R_p [C_f(C_p + C_{AK}) + C_p C_{AK}]} \end{aligned} \quad (2.4)$$

dove $A(s)$ è il guadagno dell'amplificatore operazionale.

Il G_L a bassa frequenza è pari, in modulo, al guadagno in continua dell'operazionale ($A_0 = 100 \text{ dB} = 10^5$), ha due zeri le cui frequenze corrispondono a quelle dei poli di G_I , e tre poli, di cui uno dato dall'operazionale a 50 Hz e gli altri due ricavabili risolvendo l'equazione di secondo grado ottenuta dall'annullamento del denominatore.

Si hanno ora a disposizione tutti gli elementi per ricavare il guadagno d'andata:

$$\begin{aligned} G_A &= -G_I(s)G_L(s) = \\ &= \frac{-A(s) \cdot (s^2 R_p C_p R_f C_{AK})}{1 + s[R_p(C_p + C_{AK}) + R_f(C_f + C_{AK})] + s^2 R_f R_p [C_f(C_p + C_{AK}) + C_p C_{AK}]} \end{aligned} \quad (2.5)$$

Non si possono disegnare questi grafici poichè le frequenze dei poli variano a seconda del valore di C_{AK} . Al posto di studiare analiticamente tutti i casi possibili, si può intraprendere un ragionamento veloce ma efficace.

Come precedentemente anticipato i poli del guadagno ideale non vengono influenzati da piccole variazioni della capacità dello SPAD. Il guadagno d'andata tende a introdurre poli ad alte frequenze, limitando la banda del guadagno ideale. Quindi il guadagno reale sarà identico a quello ideale finchè non interviene il guadagno d'andata, ovvero si ha un polo che attenua il guadagno reale. Tale polo si ha alla frequenza (f_r) tale che $G_I = G_A$. Anche per il guadagno d'andata i poli non subiscono variazioni dovute a modifiche del valore di C_{AK} . Tuttavia, sia G_I che G_A hanno un guadagno in banda piatta strettamente dipendente dalla capacità dello SPAD. Più precisamente all'aumentare di tale capacità il guadagno ideale in banda piatta aumenta molto più velocemente di quello di andata. Questo porta a spostare il punto d'incontro tra i due guadagni, causando una diminuzione di f_r , e quindi il restringimento della banda dove il guadagno vale effettivamente $-C_{AK}/C_f$.

Si può quindi chiarire il discorso illustrando, come in Fig. 2.3, gli andamenti di questi guadagni, disegnando la f_r in un punto arbitrario ma sensato. Dall'analisi sperimentale si può successivamente capire nel caso specifico come varia la banda piatta in funzione della capacità dello SPAD.

2.4 Osservazioni

Bisogna notare che l'operazionale presenta una capacità d'ingresso e nel circuito sono presenti dei parassitismi che rendono impreciso lo studio precedentemente svolto. Pur quanto si cerca di minimizzare tali effetti, non si possono eliminare completamente. Di conseguenza risulta scomodo svolgere calcoli sapendo già che i risultati ottenuti non sono coerenti con la realtà. L'approccio più sensato è effettuare un'analisi sperimentale, facendo uno sweep in frequenza al variare della C_{AK} . Evitando di effettuare misure inutili, si sceglie una capacità del valore

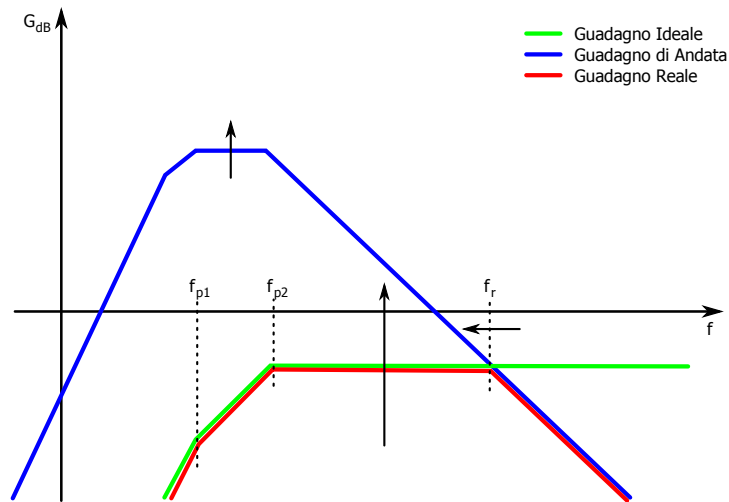


Figura 2.3: Diagrammi di Bode del modulo di G_I , G_A e G_R all'aumentare di C_{AK}

minimo e una del valore massimo che può assumere uno SPAD. Ciò porta ai seguenti due casi.

- con $C_{AK} = 0,1 \text{ pF}$, si ha una banda da 3 kHz a 950 kHz;
- con $C_{AK} = 20 \text{ pF}$, si ha una banda da 3 kHz a 150 kHz;

Le condizioni più stringenti si hanno con la capacità più alta. Si cerca quindi di stare a metà di questa banda. Tuttavia, si tiene un margine di sicurezza nel caso (improbabile) che la capacità sia superiore a 20 pF scegliendo la frequenza d'ingresso di 50 kHz.

Ricapitolando si necessita di una tensione sinusoidale in ingresso con frequenza fissata e ampiezza tale da non modificare la tensione di polarizzazione. Ciò porta a stabilire il segnale seguente d'ingresso:

$$V_{in} = 0,1 \sin(2\pi 50 \text{ k } t) \quad (2.6)$$

Di conseguenza si ottiene il seguente segnale d'uscita:

$$V_{in} = -\frac{C_{AK}}{0,7 \text{ pF}} \cdot 0,1 \sin(2\pi 50 \text{ k } t) \quad (2.7)$$

Ovvero il modulo dell'ampiezza minima in uscita (corrispondente a $C_{AK}=0,1 \text{ pF}$) di 15 mV e una massima (corrispondente a $C_{AK}=20 \text{ pF}$) di 3 V.

Capitolo 3

SpadCapMeter

In questo capitolo vengono analizzati i diversi blocchi che compongono lo SPAD-CAPMETER , ovvero il circuito che permette di elaborare fisicamente il segnale ottenuto dallo stadio precedente. Il miglior modo per la gestione e la visualizzazione dei dati è l'elaborazione numerica. Per fornire il corretto segnale di ingresso per il “Capacimetro per diodi polarizzati”, e per poter rilevare la capacità si individuano quindi le seguenti parti in cui viene strutturato il circuito:

1. Front-end analogico;
2. Convertitore A/D;
3. Microcontrollore;
4. Generatore di sinusoide;
5. Alimentazione.

Inoltre prima di iniziare a descrivere il circuito attualmente utilizzato in tutte le sue parti, si fa un breve riepilogo sui passi effettuati nella progettazione dello SPADCAPMETER .

3.1 Storia ed evoluzione del progetto

Analizzando il “Capacimetro per diodi polarizzati”, si nota che il valore della capacità è proporzionale al rapporto tra il modulo della sinusoide in ingresso e

il modulo di quella in uscita. Conoscendo l'ampiezza della sinusoidale iniettata all'ingresso, l'idea più immediata per conoscere il valore della capacità è quella di realizzare un sistema in grado di rilevare l'ampiezza di picco del segnale d'uscita.

Ci sono svariati metodi per effettuare una misura di questo genere, una prima idea è quella di sfruttare un microcontrollore che integri in un solo chip sia la parte analogica di acquisizione/conversione, che quella digitale per l'elaborazione del segnale. Questo dispositivo è il PSoC™ della Cypress Semiconductor™.

3.1.1 Il PSoC™

Un PSoC™ è, come lascia sottintendere il nome, un *System on Chip*, ovvero un dispositivo che include molte funzionalità in un unico chip. Esso infatti è composto da blocchi configurabili analogici e di logica digitale interconnessi mediante una rete programmabile. E' composto principalmente da quattro zone:

1. Core PSoC™, include una CPU, memoria, un generatore di clock e dei pin di ingresso/uscita configurabili;
2. Sistema Digitale, composto da 16 blocchi basati su architettura a 8 bit;
3. Sistema Analogico, composto da 12 blocchi ciascuno costruito attorno ad un amplificatore operazionale;
4. Risorse di Sistema, con decimatore, MAC hardware, riferimenti di tensioni.

Il clock che governa la CPU è generato internamente a 24 MHz.

Programmando il dispositivo si stabiliscono le interconnessioni tra queste parti del chip, creando il proprio sistema personalizzato. I principali blocchi che si possono usare sono:

- Convertitori analogico/digitale;
- Filtri;
- Amplificatori, amplificatori da strumentazione;
- Comparatori;

- Convertitori digitali/analogici e DAC moltiplicatori;
- PWMs e PWM con tempo morto;
- Contatori e timers;
- UART, SPI e I2C slave e master.

I blocchi analogici sono organizzati in quattro colonne da tre, ciascuna delle quali include un blocco a tempo continuo (CT) e due a capacità commutate (SC).

3.1.2 Rilevatore di picco con PSoC™

La soluzione più immediata è quella di realizzare un rilevatore di picco sfruttando tali blocchi a disposizione. La configurazione circuitale ottimale per fare ciò è illustrata in Fig. 3.1.

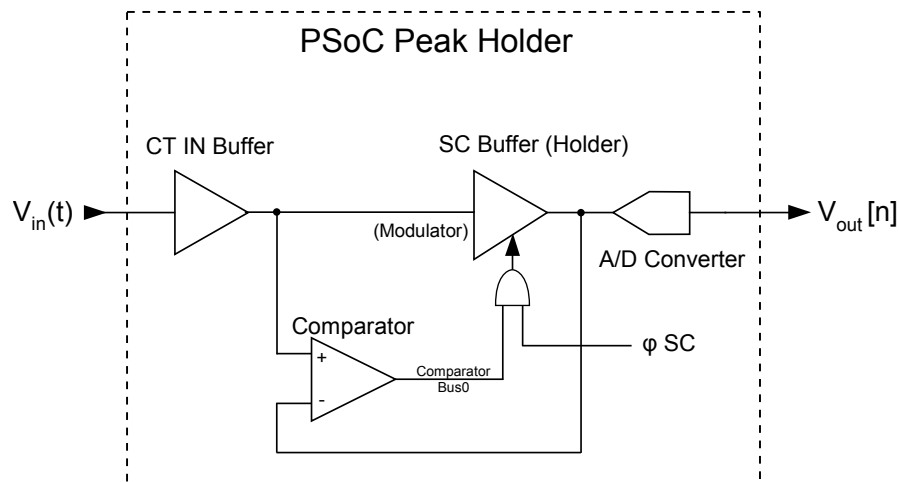


Figura 3.1: Schema di principio del rilevatore di picco con PSoC™

La tensione sinusoidale in ingresso viene memorizzata nel blocco SC buffer. L'uscita di tale holder deve essere convertita in digitale per una successiva elaborazione numerica. Perciò bisogna realizzare un campionatore affinché la tensione da acquisire sia esattamente il massimo della sinusoide in ingresso. Un semplice sampler si ottiene usando un blocco comparatore, che confronti la sinusoide in ingresso con la tensione memorizzata nel blocco SC buffer. Ciò funziona in quanto, grazie ad un ritardo introdotto dall'holder, si compara il valore della sinusoide

con lo stesso segnale nell'istante immediatamente precedente. Conseguentemente quando la differenza di queste due tensioni è positiva l'ingresso sta aumentando, mentre è negativa quando l'ingresso sta diminuendo. Questo si traduce in un fronte del comparatore in corrispondenza del massimo e del minimo della sinusoide. Si può quindi utilizzare questo segnale per abilitare o disabilitare il comando di clock del blocco a capacità commutate, che quindi trasferisce la tensione in uscita quando il segnale sta aumentando mentre la mantiene quando sta diminuendo.

Qui si riscontrano però dei problemi, ovvero:

- la tensione memorizzata non è costante a causa della scarica delle capacità presenti nel buffer a capacità commutate e delle tensioni di alimentazioni non perfettamente stabili nei blocchi;
- frequenze di campionamento degli ADC interni non sufficienti (massimo 10kHz);
- precisione degli ADC più basse delle tolleranze richieste.

Tutto ciò porta ad utilizzare un sistema di acquisizione esterno, seguito poi da una elaborazione digitale. Per evitare di usare il PSoC™ solo per l'elaborazione digitale, in quanto si sprecherebbero tutte le risorse analogiche, si è preferito usare il microcontrollore PIC.

3.1.3 Rilevatore di picco con Amplificatori Operazionali

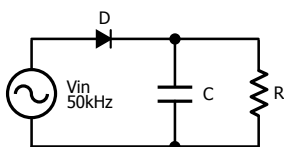


Figura 3.2: *Semplice rilevatore di picco diodo condensatore resistore*

Per quanto riguarda il front-end analogico, l'idea più immediata per trovare il massimo valore di una sinusoide è usare il rilevatore di picco, che può essere sviluppato con una semplice rete diodo, condensatore, resistore come mostrato in Fig. 3.2.

In questo modo il segnale in uscita segue la tensione in ingresso, finché quest'ultima non raggiunga l'ampiezza massima. Dopo di che tale valore viene mantenuto ai capi del condensatore. A causa di un'inevitabile

costante di scarica tale tensione non viene mantenuta costante e non è ottimale in quanto i segnali sinusoidali in gioco hanno ampiezze basse (centinaia di mV).

Alternativamente si può realizzare una linea di ritardo per avere una sinusoide opportunamente sfasata da comparare con l'originale, ottenendo quindi un segnale logico, in particolare lo Start of Conversion dell'ADC. Esso ha le stesse funzioni e caratteristiche di quello ottenuto nel caso del PSoC™, con la differenza che non viene mantenuto il valore di picco della sinusoide. Quindi il ritardo deve essere tale da far commutare il comparatore in corrispondenza del massimo della sinusoide sfasata. Tale rete si può ottenere con un filtro passa tutto realizzato ad amplificatori operazionali, dove lo sfasamento dipende dalla costante di tempo RC . Tuttavia questo è uno stadio invertente, quindi per confrontare il segnale in uscita con quello originale, si deve aggiungere un ulteriore amplificatore in configurazione invertente. Ciò si traduce nel circuito in Fig. 3.3.

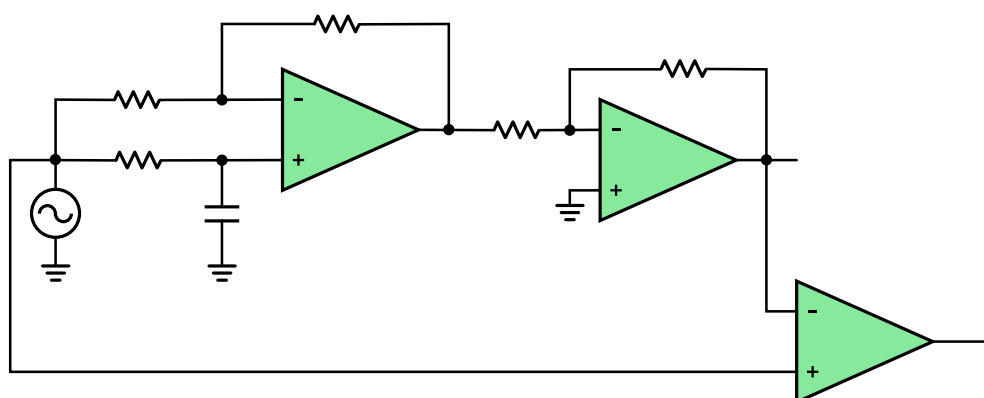


Figura 3.3: Rilevatore di picco con rete di ritardo

Come si può notare tale schema ha degli svantaggi poichè è complesso in quanto presenta un numero elevato di componenti e richiede una taratura della costante di tempo precisa e dipendente dal tempo di risposta del comparatore. Inoltre il segnale da inviare all'ADC non è la sinusoide in uscita del “Capacimetro per diodi polarizzati” bensì quella in uscita dal secondo operazionale. Il passo consecutivo è quindi quello di semplificare tale circuito.

Si può sfruttare il fatto che si conosce esattamente la frequenza del segnale d'ingresso, ciò vuol dire che prendendo un punto qualsiasi della sinusoide si può

calcolare esattamente il ritardo necessario per spostarsi nel punto corrispondente al massimo. In particolare si sa che tra il passaggio per lo zero e il punto in corrispondenza del massimo c'è uno sfasamento di 90° . Trovare l'istante di zero è però molto semplice utilizzando un comparatore. La soluzione che evince da questo ragionamento è mostrata in Fig. 3.4.

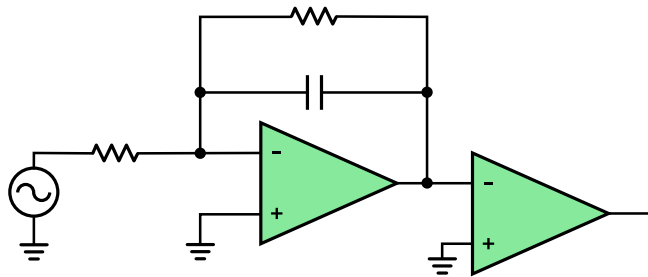


Figura 3.4: Rilevatore di picco sfasatore di 90°

Esso è un filtro passa basso attivo del 1° ordine. Per ottenere un ritardo di un quarto di periodo bisogna posizionare il polo almeno una decade prima di 50 kHz. Per compensare l'attenuazione introdotta, si modifica il rapporto R_f/R_i . In questo caso si nota che non interessa ottenere un segnale di ampiezza identica all'ingresso in quanto interessa solo il passaggio per lo zero. Quindi il guadagno di questo stadio deve tale da fornire un segnale facilmente discriminabile dal comparatore.

All'uscita del comparatore si ottiene un'onda quadra con un fronte di salita in corrispondenza del minimo del segnale non sfasato e un fronte di discesa quando tale segnale raggiunge il valore massimo. Di conseguenza per acquisire la tensione desiderata l'ADC deve iniziare la conversione sul fronte di discesa dell'onda in uscita dal comparatore.

Anche questa configurazione presenta un difetto, che non può essere trascurato: all'aumentare dell'ampiezza del segnale sinusoidale, l'amplificatore operazionale della rete ritardante introduce offset non controllabile. Di conseguenza il fronte di discesa del SOC non si verifica nell'istante in cui la sinusoide raggiunge il valore massimo, acquisendo una tensione sbagliata.

Per ovviare a questo problema si può pensare di realizzare una rete tale da rilevare il passaggio per lo zero della sinusoide in ingresso e successivamente ri-

tardare dello stesso tempo (un quarto di periodo) il segnale logico in uscita dal comparatore. Realizzando il circuito si ottiene un'onda quadra con le stesse caratteristiche della rete precedente, ed inoltre con i fronti che non risentono l'influenza dell'ampiezza dell'ingresso.

3.2 Front-end analogico

Riprendendo i risultati ottenuti nel paragrafo precedente, si raggiunge la soluzione circuitale ottimale per la rilevazione del massimo del segnale d'ingresso. Non bisogna dimenticare che tale tensione potrebbe avere un offset indesiderato. Il metodo migliore per evitare errori introdotti da questa componente del segnale d'ingresso è introdurre un filtro. A questo punto ci sono tutti gli elementi per progettare il circuito definitivo del front-end analogico. Ricapitolando occorre un filtro in ingresso, un comparatore che rilevi il passaggio per lo zero e una rete digitale che sfasi il segnale logico in uscita dal comparatore. Tutto ciò è rappresentato schematicamente in Fig. 3.5.

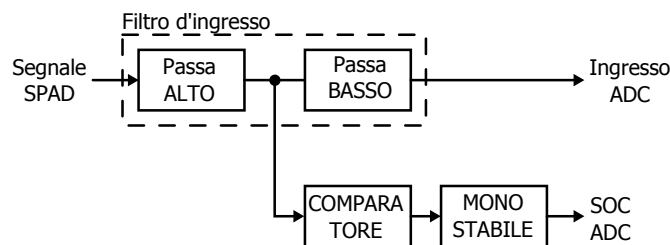


Figura 3.5: Schema a blocchi del circuito di ingresso SPADCAPMETER

Nel dettaglio si studiano le singole parti del progetto.

3.2.1 Filtro d'ingresso

Come anticipato, serve un filtro passa alto per eliminare la componente continua del segnale. Inoltre è necessario un filtro anti-alias, cioè passa basso con frequenza di taglio uguale alla banda del segnale, nel caso ideale, in ingresso all'ADC. Quindi il segnale, prima di essere convertito, subisce un filtraggio passa banda. Tuttavia il filtraggio anti-alias serve solo per la conversione analogica digitale, quindi da tale rete bisogna ottenere un segnale filtrato in banda e uno senza componenti a basse

frequenze. I filtri più semplici sono quelli passivi del primo ordine. Mettendo in cascata un passa alto e un passa basso di questo tipo si riesce ad avere una funzione di trasferimento tipica del filtro passa banda e ad estrarre facilmente il segnale a valle del solo filtro passa alto. Di conseguenza la rete scelta è mostrata in Fig. 3.6.

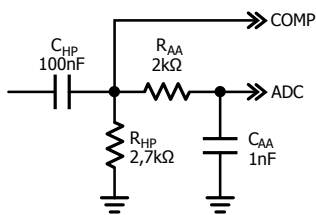


Figura 3.6: Filtro passa banda passivo

I due filtri passivi si possono studiare indipendentemente tenendo conto che le frequenze di taglio non siano vicine. Infatti se esse hanno valori simili le capacità sono dello stesso ordine di grandezza e lo studio della funzione di trasferimento si complica. Dato che non serve avere una banda stretta si può fare lo studio

separato delle due frequenze. Per prima cosa viene fissata la frequenza di taglio superiore f_{AA} (frequenza anti-alias), cioè la frequenza in cui il filtro anti-alias inizia ad attenuare. Dato che l'ingresso è a 50 kHz è stata scelta la frequenza di taglio di 80 kHz per evitare di ridurre l'ampiezza del segnale. Fissato questo valore e sapendo che $f_{AA} = 1/(2\pi R_{AA}C_{AA})$ i valori commerciali di resistenza e condensatore che più si avvicinano sono $R_{AA} = 2\text{ k}\Omega$ e $C_{AA} = 1\text{ nF}$ che danno una f_{AA} reale pari a 79,5 kHz. Ora si può determinare la frequenza di taglio del filtro passa alto, $f_{HP} = 1/(2\pi R_{HP}C_{HP})$, sapendo che deve essere molto minore di f_{AA} , ma sufficientemente alta da attenuare bene la componente continua. A questo punto si può introdurre un nuovo vincolo progettuale. Sapendo che l'alimentazione di rete introduce disturbi sinusoidali a 50 Hz, è importante scegliere f_{HP} tale da attenuare tali componenti indesiderate, stando almeno due decadi prima di f_{AA} per le motivazioni precedentemente indicate. Si può decidere di fissare tale frequenza nell'intorno di 500 Hz. Di conseguenza i valori di resistenza e condensatore che forniscono un buon risultato sono $R_{HP} = 2,7\text{ k}\Omega$ e $C_{HP} = 100\text{ nF}$ (in corrispondenza dei quali $f_{HP} = 590\text{ Hz}$ che garantisce un'attenuazione a 50 Hz di 21,43 dB).

3.2.2 Comparatore

Ottenuto il segnale senza offset, si deve rilevare il relativo passaggio per lo zero. Il modo più semplice per ottenere tale risultato è usare comparatore, più precisa-

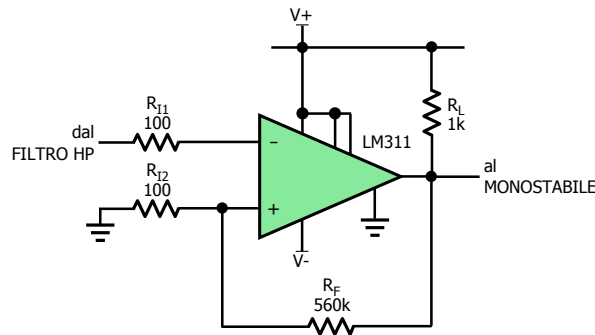


Figura 3.7: Comparatore

mente la configurazione a trigger di Schmitt per evitare commutazioni indesiderate dovute al rumore. Siccome si vuole ottenere un fronte negativo in corrispondenza del punto massimo dell'ingresso, si sceglie la configurazione invertente. In tale modo quando il segnale è positivo l'uscita assume valore logico basso, mentre quando è negativo l'uscita si porta al valore logico alto. Di conseguenza si ottiene un fronte di discesa in corrispondenza del passaggio dello zero che è, temporalmente parlando, un quarto di periodo prima del valore massimo. Allora lo schema scelto è mostrato in Fig. 3.7.

Per realizzare ciò viene scelto l'integrato LM311, per le seguenti caratteristiche ritenute vantaggiose in questo progetto:

- tensione di alimentazione duale, da 2,5 V a 18 V;
- tensione di offset tipica all'ingresso di 2 mV;
- uscita tipo open collector;
- tempo di risposta di 115 ns.

Queste garantiscono il vantaggio di non aver bisogno di specifici valori di alimentazione. Sapendo inoltre che i segnali in gioco possono anche assumere valori di qualche decina di mV (dipende infatti dalla valore della C_{ak} dello SPAD utilizzato) è indispensabile una buona sensibilità in ingresso. In più tale dispositivo, grazie all'uscita open collector, è compatibile con tutte le famiglie logiche integrate. Come ultimo aspetto, siccome la sinusoide ha una frequenza $f_{sin}=50$ kHz e

quindi $T_{sin} = 1/f_{sin} = 20 \mu s$, il tempo di risposta di 115 ns permette di avere una commutazione in uscita praticamente istantanea.

Dall'analisi del circuito si possono trovare le soglie del comparatore che sono:

$$V_{H+} = V_{SAT+} \cdot \frac{R_{i1}}{R_{i1} + R_F} \quad (3.1)$$

$$V_{H-} = V_{SAT-} \cdot \frac{R_{i1}}{R_{i1} + R_F} \quad (3.2)$$

Si deve ricordare che i segnali hanno ampiezze molto piccole, e quindi le soglie non possono essere dello stesso ordine di grandezza. Di conseguenza vengono scelte almeno un ordine di grandezza inferiore alla minima ampiezza d'ingresso. Come anticipato, il segnale minimo assume valori di qualche decina di mV. La scelta diventa quindi di impostare le soglie a qualche mV. Dato che l'alimentazione non è ancora definita si possono dimensionare R_{i1} e R_F affinché, indifferentemente da V_+ e V_- , le soglie assumono il valore desiderato. In più si nota che, avendo un'uscita open collector, $V_{SAT+} \simeq V_+$ mentre $V_{SAT-} \simeq 0$. In tal modo si ottiene:

$$V_{H+} = V_+ \cdot \frac{R_{i1}}{R_{i1} + R_F} \quad (3.3)$$

$$V_{H-} = 0 \cdot \frac{R_{i1}}{R_{i1} + R_F} \quad (3.4)$$

Quindi il rapporto $R_{i1}/(R_{i1}+R_F)$ deve essere minore di 0,001 (perchè deve essere successivamente moltiplicato per la tensione di alimentazione). Questo vincolo si traduce in R_F maggiore di $1000R_{i1}$. Per non ottenere valori di resistenze che commercialmente non esistono, viene scelta R_{i1} nell'ordine del centinaio di Ω con conseguente R_F nell'ordine del centinaio di $k\Omega$, al più di qualche $M\Omega$. Più precisamente viene scelta $R_{i1}=100 \Omega$ $R_F=560 k\Omega$, ottenendo:

$$V_{H+} = V_+ \cdot \frac{R_{i1}}{R_{i1} + R_F} = 0,00018 \cdot V_+ \quad (3.5)$$

$$V_{H-} = 0 \cdot \frac{R_{i1}}{R_{i1} + R_F} = 0 \quad (3.6)$$

Si può verificare che questo dimensionamento è corretto calcolando la soglia minima e quella massima, corrispondenti alla minore e alla maggiore alimentazione supportata dall'LM311. Quindi:

- $V_+=+2,5\text{ V}$, $\rightarrow V_{H+}= 0,45\text{ mV}$;
- $V_+=+18\text{ V}$, $\rightarrow V_{H+}= 3,2\text{ mV}$.

Infine, per avere simmetria all'ingresso del comparatore, viene messa la resistenza $R_{i2}=R_{i1}$, mentre per portare l'uscita al valore logico alto desiderato si deve mettere una resistenza di pull-up (R_L) sull'uscita. Per evitare che il comparatore assorba una corrente elevata, tale resistenza è stata scelta di $1\text{ k}\Omega$.

3.2.3 Ritardatore

Per realizzare digitalmente un ritardatore un'idea è quella di usare un circuito monostabile. Esso sostanzialmente ha la caratteristica di fornire in uscita un valore logico stabile e perturbarlo esclusivamente a fronte di un impulso in ingresso per un periodo (T_M) fissato. Bisogna far in modo che tale rete abbia come livello stabile lo zero logico e che a seguito del fronte negativo in uscita dal comparatore si porti al valore non stabile, cioè livello logico alto, tornando al valore stabile (fornendo quindi un fronte di discesa) esattamente $T_{sin}/4=5\text{ }\mu\text{s}$ dopo la perturbazione ricevuta in ingresso.

Il circuito che rispecchia in modo ottimale le caratteristiche appena esposte è la configurazione di Fig. 3.8 che usa l'integrato NE555 come monostabile.

Questo devices presenta le seguenti caratteristiche principali:

- tensione di alimentazione da $4,5\text{ V}$ a 16 V ;
- tempo di risposta di 100 ns ;
- corrente massima assorbita dall'alimentazione di 15 mA ;

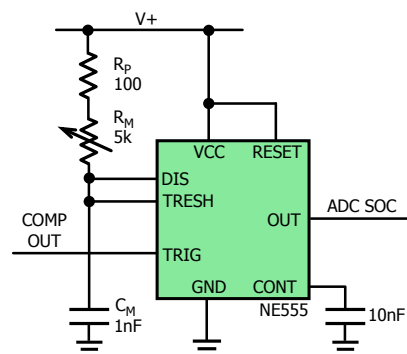


Figura 3.8: Monostabile con 555

- errore di temporizzazione contenuto entro l'1% del valore teorico.

Tale integrato, nella configurazione illustrata in Fig. 3.8, necessita di un impulso (spike) negativo in ingresso e perturba l'uscita (da livello logico basso a quello alto) per un tempo fissato dalla rete composta da R_M e C_M . Più precisamente:

$$T_M = -\ln(1/3) \cdot R_M \cdot C_M = 1,1 \cdot R_M \cdot C_M \quad (3.7)$$

Tale tempo deve essere quindi pari a $5 \mu\text{s}$. Fissando $C_M=1 \text{ nF}$, si ottiene quindi $R_M= 4545 \Omega$. Ovviamente tale valore non esiste commercialmente, ma può essere ottenuto con un trimmer da $5 \text{ k}\Omega$. Ciò si traduce inoltre nella possibilità di tarare il circuito in maniera da rilevare il massimo nel modo ottimale. Per evitare che in fase di taratura, accidentalmente, il trimmer sia nella posizione corrispondente a resistenza nulla si aggiunge in serie una resistenza di protezione $R_P= 100 \Omega$.

Come precedentemente anticipato, l'integrato richiede uno spike negativo istantaneo in ingresso per funzionare correttamente. L'uscita del comparatore invece è un'onda quadra. Di tale segnale bisogna quindi risaltare il fronte di discesa, ovvero mantenere sempre a livello logico alto il segnale tranne quando si verifica un fronte negativo. Per prima cosa serve quindi un filtro passa alto che elimina la componente continua dell'onda quadra, lasciando passare solo il fronte (poichè è una frequenza infinita). Ciò si realizza con una rete CR. Tuttavia se tale rete è riferita a massa l'uscita sarà costante a zero, presentando uno spike positivo in corrispondenza del fronte di salita e uno spike negativo in corrispondenza del fronte di discesa. Per avere invece l'uscita sul livello logico alto, si deve riferire il filtro composto da C_{S1} e R_{S1} alla tensione di alimentazione. A tal punto si nota che gli spike positivi superano il valore della tensione di alimentazione. Per eliminare questi spike che per giunta non servono, si aggiunge un diodo come mostrato in Fig. 3.9.

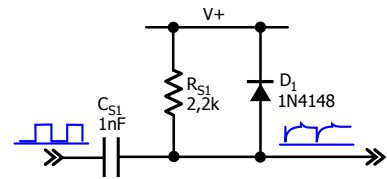


Figura 3.9: Generatore di impulsi per trigger

In tal modo quando si verifica uno spike positivo la tensione all'anodo è maggiore di quella al catodo, il diodo D_1 è polarizzato direttamente e quindi si comporta

come un corto circuito, portando la tensione al valore di alimentazione. E' proprio tale rete che bisogna apporre tra il comparatore e l'ingresso del monostabile.

Bisogna però dimensionare la frequenza del polo affinché la costante di scarica del condensatore sia sufficiente da garantire una larghezza dello spike tale da essere rilevato, e allo stesso tempo inferiore al semiperiodo dell'onda quadra. Fissato $C_{S1}=1\text{ nF}$, si può misurare sperimentalmente la resistenza che garantisce il miglior risultato. Nel caso specifico da noi esposto $R_{S1}=2,2\text{ k}\Omega$.

In definitiva la rete completa che svolge la funzione del ritardatore digitale è riportata in Fig. 3.10.

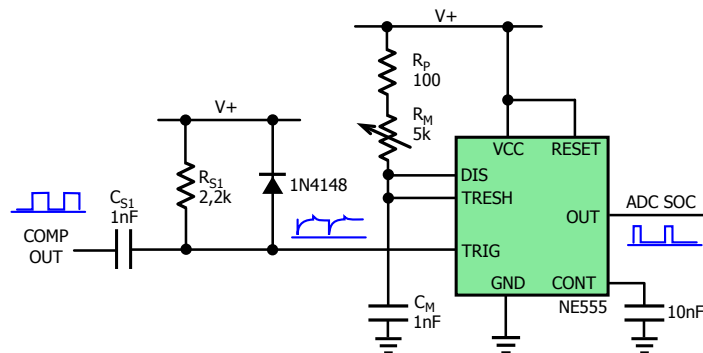


Figura 3.10: Circuito per sfasatore digitale

3.3 Convertitore A/D

Focalizzato il funzionamento del front-end analogico e note le proprietà del segnale da convertire, si devono tradurre tutte queste informazioni nelle caratteristiche essenziali necessarie per scegliere l'ADC.

3.3.1 Requisiti di conversione

Bisogna cercare un convertitore che sia alimentato con una tensione analoga a quella del resto del circuito e abbia una frequenza di campionamento superiore a quella del segnale (in questo caso non si richiede il doppio della banda in quanto nel progetto si richiede un solo punto per periodo).

Coerentemente con il front-end analogico, deve essere in grado di iniziare la conversione a seguito della lettura di un fronte di discesa all'ingresso dedicato al SOC.

Riprendendo le informazioni ottenute studiando il "Capacimetro per diodi polarizzati", e sapendo che la misura necessita di un errore non superiore a 5 fF, si possono calcolare altre caratteristiche. Importante è la risoluzione. Il segnale da convertire, V_{sin} si ottiene da questo semplice conto:

$$V_{sin} = 0,1 \cdot \sin(2\pi \cdot 50 \text{ kHz } t) \cdot \frac{C_{AK}}{700 \text{ fF}} \text{ V} \quad (3.8)$$

Il massimo di questa funzione è dunque:

$$V_{Msin} = 0,1 \cdot \frac{C_{AK}}{700 \text{ fF}} \text{ V} \quad (3.9)$$

Per avere una risoluzione richiesta di 5 fF, vuol dire che il minimo incremento di tensione dell'ingresso, che cambia di un LSB la conversione, sia esattamente pari a:

$$\Delta V = 0,1 \cdot \frac{5 \text{ fF}}{700 \text{ fF}} = 0,7 \text{ mV} \quad (3.10)$$

Si può quindi riassumere che l'ADC necessario deve:

1. poter essere alimentato con una tensione compresa tra i 4,5 V e i 16 V;
2. avere una frequenza minima di campionamento di 50 kHz;
3. avere lo Start Of Conversion attivo basso;
4. avere dinamica d'ingresso (D) di 3V, cioè da 0 V a 3 V poichè interessa solo la parte positiva del segnale;
5. avere una risoluzione minima pari a ΔV .

3.3.2 LTC1417A

Un ADC che rispecchia tali esigenze è l'LTC1417. Le sue caratteristiche sono:

- tensione di alimentazione singola o duale, 5V;

- frequenza massima di campionamento, 400Ksps;
- dinamica d'ingresso, 4V;
- risoluzione di 14 bit;
- Start Of Conversion attivo basso;
- uscita seriale SPI.

Per controllare che il numero di bit è sufficiente bisogna tradurre il ΔV in bit, secondo la relazione:

$$n = \log_2 \left(\frac{D}{\Delta V} \right) = \log_2 \left(\frac{4 \text{ V}}{0,7 \text{ mV}} \right) = 12,5 \text{ bit} \quad (3.11)$$

Tutte le specifiche di progetto sono rispettate. Per poter acquisire ed elaborare il dato digitale bisogna conoscere il funzionamento dell'ADC. E' un convertitore ad approssimazioni successive, la cui struttura è mostrata in figura Fig. 3.11.

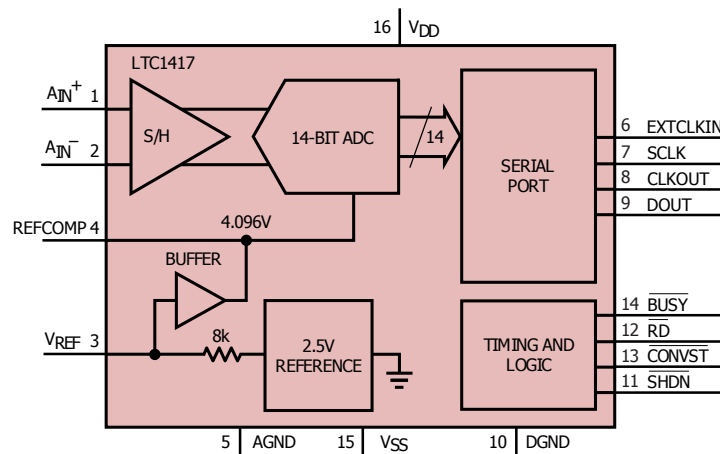


Figura 3.11: Circuito Interno dell'LTC1417A

3.3.3 Circuito di conversione

Dal momento che nell'istante della conversione la tensione è sicuramente positiva, si sceglie di usare l'alimentazione singola. In tal modo il segnale massimo in ingresso può arrivare fino a 4 V, coprendo il range richiesto. Tuttavia, essendo il segnale sinusoidale, in ingresso all'ADC arrivano anche tensioni negative. Seppur

esse non vengono convertite, introducono il rischio di danneggiare l'integrato. Per protezione si introduce il diodo D_{Z1} . E' praticamente impossibile che il convertitore fornisca in uscita un dato a 14 bit, in quanto è inevitabilmente presente del rumore che peggiora tale risoluzione; difatti già a 50 kHz il datasheet segnala un ENOB¹ di 13,5 bit. Per minimizzare tale rumore, oltre a mettere il filtro anti-alias in ingresso, si deve stabilizzare la tensione di alimentazione. Infatti, essendo un segnale che deve raggiungere ogni parte del circuito, raccoglie inevitabilmente disturbi quali, per esempio, le commutazioni del comparatore e del monostabile. Esse si traducono in fluttuazioni dell'alimentazione che portano inevitabili errori nella conversione. Il miglior filtro per togliere questi disturbi è la rete a pi-greco CLC, dove in particolare l'induttanza prende il nome di *Choke* che significa strozzamento. Precisamente tale rete è formata da $L_C = 100$ H, $C_{C1} = 100$ nF, $C_{C2} = 10$ F e $C_{C3} = 100$ nF.

Per stabilizzare le tensioni di riferimento interne al convertitore si aggiungono i condensatori C_{R1} , C_{R2} e C_{R3} dei valori consigliati nel datasheet.

L'ADC richiede SOC attivo basso di durata limitata (come il monostabile). Di conseguenza bisogna realizzare nuovamente una rete che fornisca in uscita uno spike corrispondente al fronte di discesa del segnale in uscita dall'integrato NE555. Come nel paragrafo riguardante il ritardatore digitale, si sceglie la rete formata da C_{S2} , R_{S2} e D_2 mostrata in Fig. 3.12.

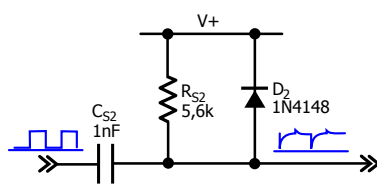


Figura 3.12: Generatore di impulsi per SOC

Anche qui bisogna scegliere la frequenza del polo affinché la costante di scarica del condensatore sia sufficiente da garantire una larghezza dello spike tale da essere rilevato, e allo stesso tempo inferiore al semiperiodo dell'onda quadra. Fissato $C_{S2}=1$ nF, si può misurare sperimentalmente la resistenza che garantisce il miglior risultato. Nel caso specifico da noi esposto $R_{S2}=5,6$ k Ω .

Questo ADC dispone 4 modalità di gestione del clock. Esso infatti integra al suo interno un timer di riferimento che è utilizzato durante le operazioni di conversione. Questo però potrebbe anche essere utilizzato per trasferire il dato

¹Effective Number Of Bit

digitale. Per avere un miglior controllo sul dispositivo si è scelto di utilizzare un clock esterno.

Infine i segnali digitali che bisogna gestire sono:

- BUSY, indica se l'ADC è occupato a convertire o è libero;
- RD, serve per abilitare l'uscita a trasferire i dati;
- DOUT, è il dato seriale in uscita dall'ADC;
- SCLK, è il clock sincrono a DOUT impostato dal dispositivo di lettura del dato.

Questi segnali vengono gestiti dall'elaboratore digitale.

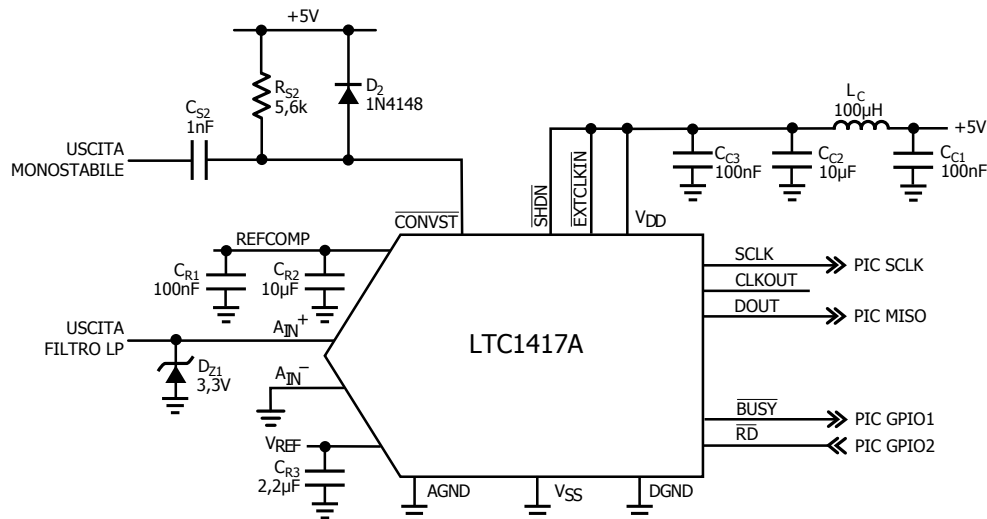


Figura 3.13: Circuito del convertitore Analogico/Digitale

3.4 Microcontrollore

Un microcontrollore è un sistema a microprocessore completo, integrato in un solo chip, progettato per ottenere la massima autosufficienza funzionale ed ottimizzare il rapporto prezzo-prestazioni per una specifica applicazione. Esso infatti integra al suo interno una CPU (generalmente a 8-16 bit), delle memorie per il salvataggio del programma e dei dati, ed altre periferiche di uso generico quali timer, UART²,

²Universal Asynchronous Receiver-Transmitter

I/O.... Per avere delle prestazioni superiori esistono dispositivi quali DSP ed FPGA, che sono dedicati per le applicazioni che richiedono un'elevata capacità di calcolo ed elevate velocità. Ovviamente con tali dispositivi la circuiteria di supporto diventa molto più complessa aumentando notevolmente il costo della realizzazione del progetto.

Dal momento che il sistema in esame non richiede particolari prestazioni, la scelta ricade su un microcontrollore. Per individuare tale dispositivo, le caratteristiche minime richieste per soddisfare le esigenze dello strumento sono:

- hardware per la comunicazione seriale SPI per interfacciarsi con l'ADC;
- bus di comunicazione veloce con PC;
- timer per generare la frequenza di 50 kHz;
- programmazione IN-Circuit.

3.4.1 Il PIC18F4550

Tra tutti i possibili microcontrollori esistenti rispecchianti le caratteristiche richieste, è stato scelto il PIC 18F4550 della Microchip Technology. Esso integra un vasto numero di periferiche, tra le quali gioca un'importante ruolo in questo elaborato il blocco hardware che gestisce la comunicazione USB. Per confermare che tale dispositivo rappresenta una valida scelta in questo progetto, si espongono ed analizzano in dettaglio le sue caratteristiche principali.

Universal Serial Bus

Come appena detto, la caratteristica principale del 18F4550 è il *controller USB* hardware, in quanto la maggior parte dei microcontrollori non possiedono questo blocco.

La parte fondamentale di questa periferica è il *Serial Interface Engine* (SIE) che si occupa di tutta la gestione del protocollo USB, il quale non è una semplice comunicazione asincrona come una normale RS232, ma è molto più articolata; questo giustifica l'importanza della presenza del blocco hardware dedicato, che

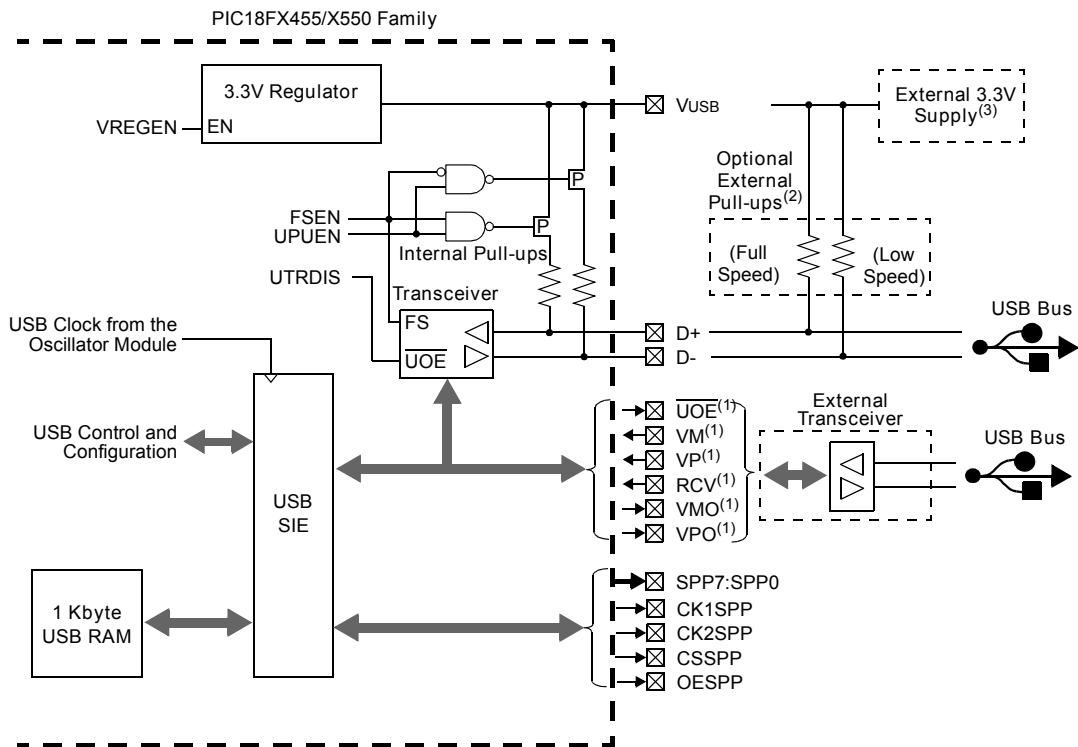


Figura 3.14: Blocco USB del PIC18F4550

gestisce la comunicazione, in quanto altrimenti sarebbe impensabile realizzare il tutto via software. Un accenno al protocollo USB sarà effettuato in Chap. 4.1.1.

A fianco del SIE ci sono alcune sotto-periferiche di supporto, che semplificano notevolmente l'utilizzo di quest'ultimo. Per primo il *transceiver* interno, il quale si occupa di convertire il *livello di trasporto* del SIE, in *livello fisico* da mandare sul cavo. Questo è supportato da un regolatore di tensione che fornisce l'alimentazione di 3,3 V richiesti dal livello fisico.

Un altro sotto-blocco molto importante per aumentare la velocità di trasferimento, punto di forza della comunicazione USB rispetto agli altri standard diffusi in campo *embedded*, è la RAM *dual-port*. Questa è mappata nella memoria del microcontrollore e permette la condivisione di dati tra il Core e il SIE. La CPU può quindi caricare dei dati in memoria e successivamente, mentre il SIE li trasferisce, essere libera di fare altre operazioni.

Il blocco USB è anche fonte di molti segnali di Interrupt, i quali sono gestiti da un modulo specifico che va poi a generare un unico interrupt alla CPU; questi

sono generati all'arrivo di nuovi pacchetti e/o a seguito di eventi speciali quali errori o condizioni trappola. Essi vengono poi processati nel software in modo opportuno.

Un'altra opzione fornita dal PIC18f4550 è lo *Streaming Parallel Port* (SPP), il quale consente di trasferire molti dati ad una memoria esterna, in modalità parallela.

Queste ed altre funzionalità sono illustrate in Fig. 3.14.

CPU e Memorie

Il PIC18F4550 è un microcontrollore di tipo RISC³ con architettura Harvard. L'architettura Harvard ha la memoria dei dati e del programma distinte, accessibili mediante due bus indipendenti; questo permette un notevole risparmio di tempo in fase di esecuzione delle operazioni rispetto alla tradizionale architettura von Neumann, la quale prevede una sola memoria contenente dati e programma. Ciò unito alla presenza di un'unità di prefetch del codice e all'utilizzo di istruzioni RISC, consente di evitare di avere dei momenti nei quali la CPU è in stallo a causa della mancanza di istruzioni da eseguire o di dati su cui operare. Le istruzioni RISC sono infatti un numero molto esiguo ma consentono di fare delle operazioni base che, combinate opportunamente tra loro, riescono comunque a far fronte a tutte le istruzioni tipicamente presenti in un CISC⁴.

Le istruzioni possono però anche essere *long word*, ovvero la CPU deve caricare più di un byte di istruzione prima di eseguirla; queste sono particolarmente utili nel caso di salti del programma in quanto viene memorizzato direttamente l'indirizzo di arrivo all'interno dell'istruzione stessa.

Ogni istruzione dell'ALU è composto da 4 *Q-Cycles*:

1. decodifica dell'operazione;
2. lettura degli operandi;
3. esecuzione dell'istruzione;
4. scrittura dei risultati.

³Reduced Instruction Set Computer

⁴Complex Instruction Set Computer

La ALU è provvista di un solo registro di accumulazione (**WREG**), mappato in memoria, e di un registro di stato (**STATUS**). Lo **STATUS** Register prevede 5 flag:

1. **N** - Negative bit, indica se il risultato in **W** è negativo;
2. **OF** - OverFlow bit, indica se il 7°bit ha fatto un overflow;
3. **Z** - Zero bit, indica se $W=0$;
4. **DC** - Digit Carry/ $\overline{\text{Borrow}}$ bit, indica un eventuale riporto nel 1°nibble di **W**;
5. **C** - Carry/ $\overline{\text{Borrow}}$ bit, indica un eventuale riporto in **W**.

Per quanto riguarda le memorie, il PIC18F4550 è dotato di 2 kbyte di memoria RAM per i dati, di 32 kbyte di memoria FLASH per il programma e di 256 byte di EEPROM per la memorizzazione di dati non volatili. La memoria RAM è mappata in 16 banchi da 256 byte cadauno. I primi 8 sono dedicati alla memoria per l'utente mentre i restanti sono occupati dai registri interni al sistema. La memoria FLASH a differenza della RAM sfrutta celle da 16 bit e può quindi contenere 16 k-istruzioni; questo è possibile grazie all'architettura Harvard come citato prima. Essa è puntata dal registro **TBLPTR**. Infine la EEPROM, che è capace di sostenere fino a 1 milione di cicli di scrittura e mantenere i dati memorizzati per un periodo non inferiore ai 40 anni, è accessibile mediante alcuni registri mappati in memoria.

Clock e PLL

Un'altra importante sezione del microcontrollore è quella che si occupa della generazione e della successiva distribuzione del clock all'interno del dispositivo. E' noto infatti che la maggior parte dei sistemi digitali, per poter funzionare al meglio (specialmente nel trasferimento di informazioni su bus paralleli), necessita di una linea di clock per la sincronizzazione. E' necessario predisporre quindi un quarzo esterno al PIC che viene opportunamente stimolato al fine di instaurare in esso delle oscillazioni, particolarmente stabili in frequenza viste le caratteristiche meccaniche del quarzo. Esse vengono poi trasformate in un segnale digitale che rappresenta il riferimento temporale del microcontrollore.

Internamente il PIC necessita di svariate frequenze per funzionare; giusto per dare un esempio, il modulo USB ha bisogno di un clock a 96MHz per la sincronizzazione con l'Host. Sfortunatamente però quarzi a frequenze così elevate non esistono⁵; da qui l'utilità del modulo PLL⁶ il quale è sostanzialmente un moltiplicatore, che permette di generare frequenze multiple di un clock di riferimento. Si può quindi utilizzare un quarzo esterno di 4MHz e, mediante prescalers e PLL, generare tutte le frequenze utili al dispositivo.

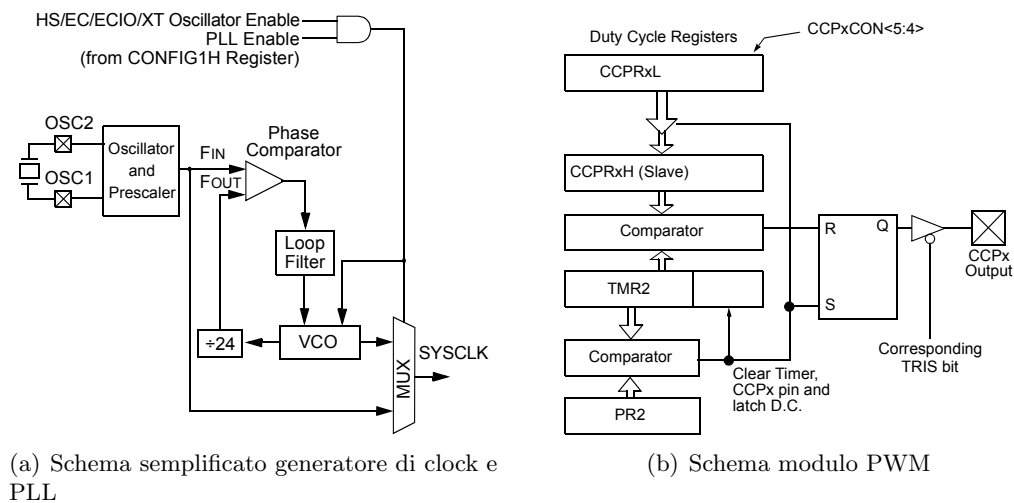


Figura 3.16: Moduli clock e PWM

Timer

Il PIC18F4550 dispone di ben 4 moduli di Timer, ognuno con particolari caratteristiche. Tutti e quattro sono a 16 bit, possono funzionare con un'ampia scelta di segnali di clock e possono generare una serie di segnali di interrupt. Inoltre alcuni di essi sono in grado di lavorare come contatori di eventi esterni al PIC o di generare delle forme d'onda. In particolare questa è una funzione molto utile per il progetto presentato in questo elaborato. Per svolgere ciò, i timer si devono appoggiare al modulo denominato *Capture\Compare\PWM* (CCP).

Analizziamo qui di seguito la struttura del Timer 2 e del modulo PWM, riportata in Fig. 3.16(b), che sono quelli utilizzati nel progetto. Il clock di sistema o

⁵a meno di utilizzare quarzi *overtone*

⁶Phase Locked Loop

un suo sottomultiplo va ad incrementare il registro TRM2, il quale viene confrontato con il contenuto di PR2, che è il registro di periodo, per generare un segnale. Questo viene poi utilizzato dal modulo PWM per settare un latch SR. Il reset di tale latch viene invece generato dalla comparazione tra TRM2 e CCPPR nel quale è memorizzato un valore corrispondente al duty-cycle (DC) dell'uscita.

SPI

La comunicazione SPI viene gestita hardware dal modulo MSSP, mostrato in Fig. 3.17. Esso consente di serializzare un dato posto nel registro SSPBUF, che viene automaticamente trasferito in SSPSR quando questo è libero, sul pin SDO oppure di memorizzare un dato seriale in arrivo sul pin SDI.

In questo tipo di comunicazione un ruolo molto importante è svolto dal clock. Infatti il dato deve essere sincrono a questo segnale per avere una corretta lettura. Di conseguenza il modulo hardware integra una sezione dedicata a ciò.

La trasmissione avviene tra un dispositivo detto *master* e uno o più *slaves*. Il master controlla il bus, emette il segnale di clock, decide quando iniziare e terminare la comunicazione.

Perciò in questo blocco hardware sono previsti entrambi i funzionamenti. Il pin SCK può essere di lettura o scrittura: se il PIC è impostato come master genera un clock su tale pin e nello stesso tempo lo usa per lo shift del registro SSPSR, mentre se è slave riceve il clock dall'esterno e lo utilizza solamente per lo shift.

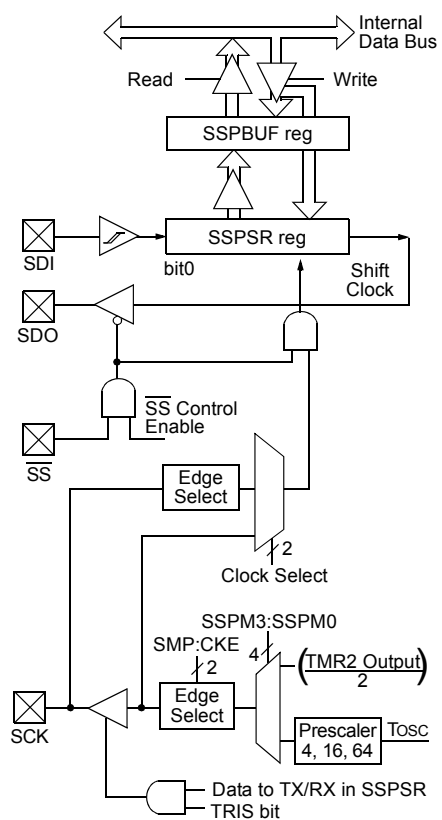


Figura 3.17: Schema modulo SPI

Programmazione mediante bootloader

Un'altra interessante caratteristica di questo microcontrollore è la possibilità di programmazione con il bootloader. Esso consiste in un piccolo programma caricato a bordo del PIC, il quale all'accensione controlla la presenza di un segnale in ingresso. Se esso è presente avvia una comunicazione con il PC attraverso l'interfaccia USB. Mediante un opportuno software è possibile inviare il programma al microcontrollore il quale provvede a scriverlo in FLASH, partendo da un indirizzo tale da non sovrascrivere la parte di codice del bootloader. Ciò consente di realizzare una facile ed efficiente programmazione IN-CIRCUIT del dispositivo senza utilizzare programmatori esterni. Bisogna sottolineare la necessità di caricare la prima volta tale bootloader sul dispositivo tramite un apposito programmatore.

3.4.2 Configurazione hardware

Visto il funzionamento del microcontrollore e le esigenze del sistema in progetto, serve sviluppare la circuiteria necessaria per la temporizzazione, la programmazione IN-CIRCUIT e i collegamenti verso le altre periferiche.

Per quanto riguarda il clock, come precedentemente detto, bisogna utilizzare un quarzo da 4 MHz collegato ai pin `OSC1` e `OSC2`. Per l'innesco delle oscillazioni, come consigliato dal datasheet, si aggiungono due condensatori verso massa da 33 pF.

Il circuito regolatore di tensione per l'USB necessita di un condensatore esterno da 470 nF che viene collegato al pin `VUSB`.

Inoltre per il caricamento del bootloader, operazione che si esegue solo al primo utilizzo, si necessita di un connettore per il collegamento con il programmatore ICD2. Questo prevede cinque linee, due di alimentazione, una di reset ($\overline{\text{MCLR}}$), una per il dato (`PGD`) e una per il clock (`PGC`).

Inoltre per il corretto funzionamento del PIC, il pin `PGM` deve essere forzato a massa.

Il segnale necessario per avviare la programmazione mediante bootloader si fornisce attraverso un pulsante collegato al pin `RB4`, in particolare esso chiude a

massa ed è tenuto normalmente alto da una resistenza di pull-up. Per assicurare che questo segnale venga letto all'avvio del microcontrollore si aggiunge un pulsante per forzare il reset attraverso il pin $\overline{\text{MCLR}}$.

I pin dedicati alla comunicazione USB sono D+ e D-, direttamente collegati al connettore USB di tipo B.

Per quanto riguarda la lettura dei dati forniti dall'ADC sono previsti i seguenti pin:

- RB1/SCK, segnale di clock;
- RB0/SDA, dato seriale;
- RD5, segnale di read;
- RD4, segnale di busy;

Infine i pin RC2 e RC0 sono utilizzati per generare onde quadre, utilizzate nei successivi paragrafi.

3.5 Generatore sinusoidale

A tal punto risulta importante riuscire a generare il segnale sinusoidale da usare nel primo circuito analizzato, cioè il “Capacimetro per diodi polarizzati”, sempre con l'obiettivo di ridurre al minimo la strumentazione esterna richiesta. Tale segnale deve avere una frequenza di 50 kHz. Il PIC tuttavia è in grado di fornire in uscita segnali logici, con livello basso corrispondente a 0 V e livello alto V_+ , di frequenza impostabile. Si ricorda che l'onda quadra si può vedere, grazie all'analisi e alla trasformata di Fourier, come somma di componenti armoniche sinusoidali con la caratteristica fondamentale di aver frequenza multipla di quella del segnale più una componente continua. Dal momento che si conosce precisamente la frequenza del segnale che si vuole ottenere, si può generare un'onda quadra e mandarla in un filtro passa banda. Ovvero, con il PIC è semplice generare un segnale logico periodico con DC del 50% a 50 kHz, però poi si deve realizzare un circuito analogico in grado di eliminare tutte le componenti del segnale tranne l'armonica

fondamentale ed in più di attenuarla o amplificarla. Un filtro passa banda come quello illustrato in Chap. 3.2.1 non rappresenta minimamente la soluzione richiesta in quanto ha una selettività ($Q = f_0/\Delta f$) molto bassa. Con selettività si intende semplicemente il rapporto tra la frequenza centrale del filtro passa banda e la differenza delle due frequenze corrispondenti a -3dB del guadagno (vedi Fig. 3.18).

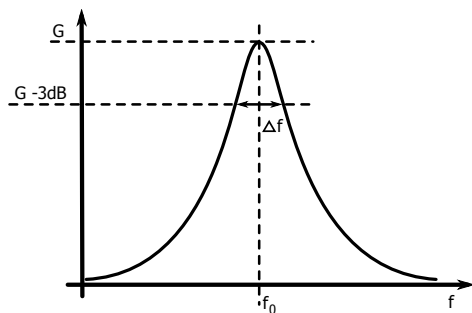


Figura 3.18: Risposta in frequenza di un filtro passa banda e punti salienti

Si passa quindi a cercare un filtro attivo, con amplificatori operazionali, che abbia una selettività sufficiente da attenuare le armoniche diverse dalla fondamentale, affinché il segnale ottenuto sia sinusoidale[7].

Tra tutte le tipologie di filtri attivi, la configurazione che presenta il miglior compromesso tra elevata selettività e semplicità è quella dei filtri a *reazione multipla* del secondo ordine, illustrata in Fig. 3.19.

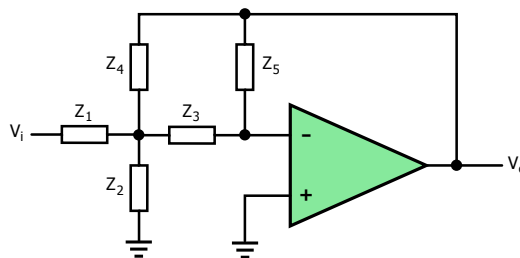


Figura 3.19: Filtro a reazione multipla generico

La funzione di trasferimento di tale schema si ottiene facilmente, osservando che:

- il nodo B è a massa virtuale;
- la corrente assorbita dall'operazione è nulla;
- le correnti del nodo A devono bilanciarsi.

Ciò si traduce nelle seguenti relazioni:

$$I_1 = I_2 + I_3 + I_4 - \frac{V_i - V_A}{Z_1} = \frac{V_A}{Z_2} + \frac{V_A}{Z_3} + \frac{V_A - V_o}{Z_3} \quad (3.12)$$

$$I_3 = I_5 \frac{V_A}{Z_2} = -\frac{V_o}{Z_5} \quad (3.13)$$

$$V_A = -V_o \frac{Z_3}{Z_5} \quad (3.14)$$

Da queste equazioni si riesce a ricavare la funzione di trasferimento $T(S)$ generale di tutti i filtri a reazione multipla del secondo ordine, ricordando che le Z_j sono le impedenze.

$$T(S) = \frac{V_o}{V_i} = -\frac{Z_4/Z_1}{1 + (Z_3 + Z_4 + Z_3Z_4/Z_1)/Z_5 + (Z_3Z_4)/(Z_2Z_5)} \quad (3.15)$$

La configurazione del filtro passa banda si ottiene usando come Z_1, Z_2 e Z_5 delle resistenze, mentre al posto di Z_3 e Z_4 dei condensatori come mostrato in figura Fig. 3.20.

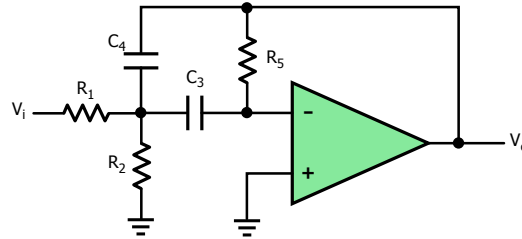


Figura 3.20: Filtro Passa Banda a reazione multipla generico

In tal modo la funzione di trasferimento che si ottiene è:

$$T(s) = -\frac{\frac{s^2}{R_1 C_4}}{s^2 + \frac{s}{R_5 C_{//}} + \frac{1}{C_3 C_4 R_{//} R_5}} = -\frac{\frac{s^2 G \omega_0}{Q}}{s^2 + \frac{s \omega_0}{Q} + \omega_0^2} \quad (3.16)$$

dove $R_{//} = R_1 R_2 / (R_1 + R_2)$, $C_{//} = C_3 C_4 / (C_3 + C_4)$ e le caratteristiche del filtro sono:

- guadagno

$$G = \frac{R_5 C_{//}}{R_1 C_4} \quad (3.17)$$

- selettività

$$Q = \omega_o C_{//} R_5 \quad (3.18)$$

- frequenza centrale

$$f_0 = \frac{\omega_0}{2\pi} = \frac{1}{2\pi \sqrt{R_{//} R_5 C_3 C_4}} \quad (3.19)$$

Fissando queste caratteristiche si ottengono i valori di R_1, R_2, R_5, C_3 e C_4 . Nel caso in esame si conosce sicuramente $f_0=50$ kHz. Per quanto riguarda il guadagno non si ha un vincolo preciso, in quanto si sa che la sinusoide deve avere un'ampiezza di 100 mV ma non si conosce l'ampiezza della prima armonica dell'onda quadra. Tuttavia si presuppone che essa sia dell'ordine di qualche V, quindi si sceglie $G=0,2$ per ottenere un'ampiezza in uscita nell'ordine delle centinaia di mV o di 1 V. Ultimo parametro è la selettività. Anche qui bisogna definire un vincolo. Per avere una sinusoide pulita bisogna attenuare notevolmente la seconda armonica, cioè 100 kHz. Si deve far attenzione al fatto che aumentando troppo il fattore Q il sistema rischia di oscillare da solo. Si sceglie allora che tale frequenza venga attenuata di un fattore 500 rispetto a G. Sostituendo tutti i dati impostati nell'equazione 3.16 si ottiene $Q=8$.

Ottenuti tutti i vincoli progettuali e fissati $C_3=C_4=220$ pF, ecco cosa si ottiene:

$$R_1=103 \text{ k}\Omega, R_2=630 \Omega, R_5=118 \text{ k}\Omega.$$

In particolar modo si nota che $R_{//}$, ottenuta come parallelo tra R_1 e R_2 , è praticamente uguale a quest'ultima. Di conseguenza, dalla (3.19), si evince che variando R_2 si modifica la frequenza ma di conseguenza, dalla (3.18), anche la selettività. Si sceglie usare al posto della resistenza commerciale più prossima al valore ottenuto, una resistenza di 560Ω in serie ad un trimmer di 200Ω che consentono una taratura. Infine per le altre due resistenze si scelgono i valori commerciali che più si avvicinano a quello teorico. Per la precisione, $R_1=100 \text{ k}\Omega$ e $R_5=120 \text{ k}\Omega$.

Tuttavia si deve prestare attenzione al fatto che un carico resistivo in uscita può alterare il funzionamento del filtro fino a renderlo instabile e facendolo oscillare. Per ovviare a tale problema si aggiunge un buffer, che fornisce in uscita ciò che

legge in ingresso, indipendentemente dal carico. Questo viene garantito in quanto l'operazionale in tale configurazione non assorbe corrente e ha una resistenza d'ingresso infinita, e quindi come se fosse un circuito aperto.

Inoltre bisogna fare in modo che l'ampiezza della sinusoide in uscita sia esattamente 0,1 V. La modifica di R_1 , R_2 ed R_5 è sconsigliata in quanto si variano le altre caratteristiche essenziali del filtro. In più, come appena illustrato, è preferibile evitare di modificare l'uscita del passa banda. Da com'è scelto il guadagno in tale progetto, si ottiene sicuramente una tensione superiore al valore desiderato. Di conseguenza è comodo inserire una rete di attenuazione in ingresso.

Più precisamente tale rete, mostrata in Fig. 3.21 è composta da un condensatore C_B per eliminare la componente continua e centrare l'onda quadra in 0 V, un diodo per diminuire l'ampiezza dell'onda e un partitore resistivo (R_a e R_b) per definire precisamente l'attenuazione. Siccome si deve eliminare solo la componente continua, la capacità deve avere un valore tale da non filtrare anche frequenze nell'ordine di 50 kHz. Per tale motivo viene scelta $C_B = 1$ nF. Per il dimensionamento del partitore, risulta più efficiente svolgere una taratura sperimentale. Da queste misurazioni si trova, come valori ottimali per ottenere 100 mV in uscita, $R_a = 2$ k Ω e $R_b = 33$ k Ω .

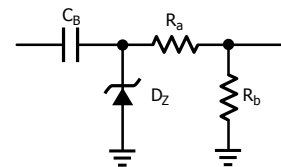


Figura 3.21: Schema attenuatore

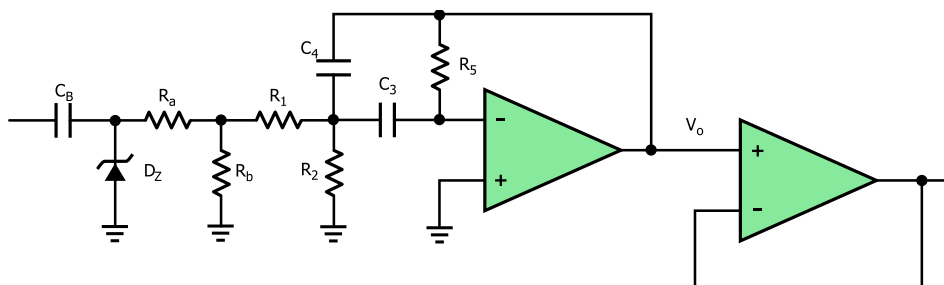


Figura 3.22: Schema completo generatore 50 kHz

Riassumendo, il segnale sinusoidale a 50 kHz di ampiezza 0,1 V si ottiene con il circuito illustrato in Fig. 3.22.

3.6 Alimentazione

Dal momento che si vuole usare la comunicazione USB, provvista di una linea di alimentazione, che può fornire 5 V erogando una corrente massima di 500 mA, si può progettare la sezione di alimentazione per rendere il circuito indipendente dalla rete elettrica. Purtroppo questa tensione non basta, in quanto per generare la sinusoide, la circuiteria necessita di alimentazione negativa in quanto l'onda in uscita dal filtro passa banda precedentemente descritto deve essere centrata in zero e varierà da un valore massimo di 100 mV e un valore minimo di -100 mV. Un altro motivo è che il comparatore deve rilevare il passaggio per lo zero, quindi anche lui necessita di un'alimentazione duale. Per evitare di complicare l'utilizzo di tale sistema, si è cercato un metodo diverso per alimentare il circuito da quello classico trasformatore+stabilizzatore. Dal momento che si ha a disposizione una tensione continua (i 5 V dell'USB) una soluzione efficiente è quella di usare un circuito DC-DC, cioè una rete elettrica in grado di convertire una tensione continua in ingresso in un'altra in uscita.

La configurazione più semplice per realizzare questo tipo di sistema di alimentazione è usare una *pompa di carica* mostrata in Fig. 3.23.

Per semplicità, si può immaginare che tale rete sia composta da un circuito fissatore seguito da un rettificatore. Data un'onda quadra in ingresso, più precisamente generata dal PIC e quindi di ampiezza 0-5 V, la prima sezione modifica questo segnale spostando il suo valor medio da 2,5 V a -1,8 V. Ciò è possibile in quanto durante il T_{ON} dell'onda il diodo è polarizzato direttamente e fissa la tensione ai suoi capi, nel punto (1), a 0,7 V. Di conseguenza sul condensatore si ha una d.d.p. di $5-0,7=4,3$ V. Durante il successivo T_{OFF} il condensatore non può scaricarsi istantaneamente e quindi la tensione in (1) si porta al valore -4,3 V dato che in ingresso la tensione si è portata a 0 V. Tale valore viene mantenuto in quanto D_1 risulta polarizzato inversamente. In questo caso si nota che D_2 risulta polarizzato direttamente e in uscita trasferisce la tensione in (1) a meno della c.d.t. ai suoi capi, cioè -3,6 V. Durante il T_{ON} invece è spento e C_2 mantiene la tensione costante. In realtà la corrente prelevata da questo stadio tende a scarica-

re il condensatore C_2 . Per mantenere la tensione d'uscita costante bisogna quindi usare un segnale d'ingresso con un periodo inferiore alla costante di scarica del condensatore.

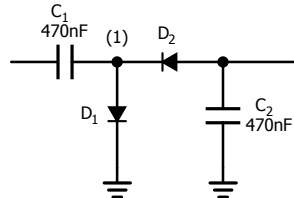


Figura 3.23: Schema pompa di carica

Riassumendo, il sistema usa un sistema di alimentazione dove $V_+ = 5\text{ V}$ e $V_- = -3,6\text{ V}$. Ciò non comporta nessun problema in quanto il segnale che si deve generare ha ampiezza massima $0,1\text{ V}$ e minima $-0,1\text{ V}$, mentre la tensione in ingresso ha valore massimo di 3 V e minimo -3 V . L'operazionale riesce quindi a non saturare, mentre il comparatore anche se satura riesce a rilevare il passaggio per lo zero. In ogni caso si può concludere che tale alimentazione è compatibile con il funzionamento del circuito. Come ultima analisi bisogna controllare se la corrente assorbita dal sistema sia inferiore a 500 mA (corrente massima fornita dalla porta USB). Per tale calcolo si può controllare l'assorbimento di ogni singolo blocco, anche attraverso le informazioni fornite dai datasheet. Ecco una stima degli assorbimenti.

- filtro d'ingresso del front-end: 0 A poichè è una rete passiva;
- comparatore: massimo 10 mA ;
- monostabile: massimo 15 mA ;
- convertitore A/D: massimo 6 mA ;
- LED: 4 mA ;
- operazionale: 15 mA ;
- microcontrollore: massimo 25 mA .

Di conseguenza la massima corrente assorbita dal circuito è 75 mA e quindi la scelta di tale sistema di alimentazione risulta un'ottima soluzione.

3.7 Realizzazione progetto hardware

Definiti e progettati i vari blocchi, bisogna unirli per realizzare il circuito che svolge tutte le funzioni illustrate. Tuttavia non è immediato questo passaggio. In particolare bisogna notare che ci sono parti digitali e parti analogiche, e quindi una massa digitale *DGND* e una analogica *AGND*. Esse dovranno essere unite in un solo punto del circuito.

Inoltre il cavo USB è schermato per evitare che eventuali disturbi si accoppino sulla linea di trazione dei dati. Tali disturbi vengono quindi catturati dalla schermatura, traducendosi in un accumulo di carica. Il connettore USB prevede due pin per questa schermatura, che bisogna collegare a *DGND*. Questi accumuli di carica possono modificare il livello di massa e portare disfunzionalità nel sistema, per ciò bisogna inserire un filtro passa basso che elimini questa componente continua realizzando la rete consigliata in [6]

Per evitare collegamenti sbagliati o cortocircuiti si preferisce usare un fusibile autorigenerante di 350 mA, che protegge sia il circuito sia la porta USB utilizzata.

Inoltre, è indispensabile filtrare tutte le alimentazioni in prossimità di ogni integrato per eliminare i disturbi che si accoppiano alle tensioni di alimentazione. Solitamente tale filtraggio si effettua con una coppia di condensatori in parallelo, uno dell'ordine dei μF e uno nell'ordine del centinaio di nF in particolare con quest'ultimo praticamente attaccato al piedino di alimentazione del device.

In particolar modo l'ADC e l'operazionale attorno al quale è progettato il filtro passa banda richiedono un'alimentazione estremamente pulita da rumori, nel primo caso per non sbagliare la conversione e nel secondo caso per evitare che disturbi perturbino la stabilità del sistema. Per quanto riguarda il convertitore, tale filtraggio viene discusso nel capitolo 3.2.1, mentre

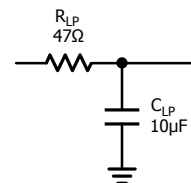


Figura 3.24: Filtro alimentazione operazionale

per l'operazionale si sceglie un filtro passa basso passivo del primo ordine, come mostrato in Fig. 3.24.

Precisamente ciò è indispensabile per l'alimentazione negativa, in quanto è generata partendo da un'onda quadra. Per ridurre la banda passante in bassa frequenza si scelgono $R_{LP}=47\ \Omega$ e $C_{LP}=10\ \mu\text{F}$. In tal modo si ottiene una frequenza di taglio $f_{LP} = 1/(2\pi R_{LP}C_{LP}) \simeq 300\ \text{Hz}$

Tutto il progetto dell'hardware dedicato e le considerazioni aggiuntive fin qui analizzate si traducono nello schema realizzato con il programma OrCad Capture mostrato in Fig. 3.26 e Fig. 3.27.

Grazie al programma OrCad Layout si passa al disegno vero e proprio del circuito stampato. Per arrivare a ciò bisogna tener conto dei seguenti accorgimenti:

- raggruppare i vari componenti di ogni blocco per semplificare lo sbroglio e le operazioni in fase di saldatura;
- tenere i condensatori di riferimento dell'ADC e quelli di filtraggio delle alimentazioni il più possibile vicino all'integrato;
- evitare che segnali digitali con commutazioni veloci passino vicino al segnale da convertire, alle linee di alimentazione, sotto e in prossimità degli integrati;
- usare un piano di massa per far sì che la corrente rientrante veda una resistenza pressochè nulla e quindi non si trasformi in una caduta di tensione;
- collegare AGND e DGND in un unico punto.

Conseguentemente il circuito stampato ottenuto è mostrato in Fig. 3.28 e Fig. 3.29. Infine il master realizzato e saldato è mostrato in Fig. 3.30, mentre la 3.31 mostra lo strumento inscatolato.

Per finire bisogna ricordare che dopo aver saldato tutti i componenti la prima cosa da fare è caricare il programma corretto nel PIC. Non da meno è essenziale regolare il trimmer per selezionare precisamente la frequenza centrale del filtro passa banda e quindi controllare che non ci siano parassitismi o collegamenti non corretti che portino all'instabilità di tale rete. Per quanto riguarda l'acquisizione

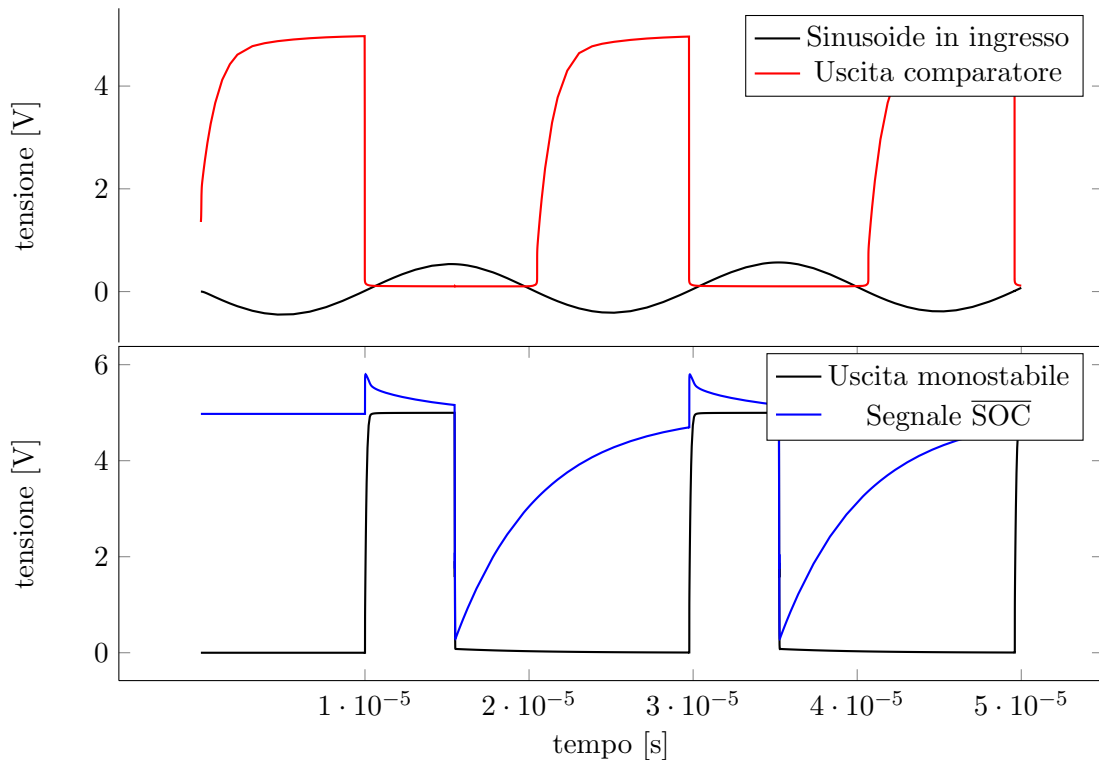


Figura 3.25: Schermate dell'Oscilloscopio

del massimo segnale della sinusoide bisogna effettuare rigorosamente la regolazione del ritardo del monostabile. Ciò si può svolgere iniettando in ingresso la sinusoide a 50 kHz visualizzandola su un canale dell'oscilloscopio. Conseguentemente con un secondo canale di tale strumento si controlla che l'uscita del comparatore sia effettivamente un'onda quadra con T_{OFF} quando la sinusoide è positiva. Infine con un terzo canale dell'oscilloscopio si legge il segnale di inizio conversione dell'ADC (SOC). In particolare si deve tarare la costante di tempo del monostabile affinché il fronte di discesa di tale segnale sia esattamente in corrispondenza del massimo del segnale da convertire. Una volta regolato il trimmer che consente ciò si può controllare con il quarto canale dell'oscilloscopio anche il dato in uscita dell'ADC, rigorosamente seriale con tempo di trasferimento inferiore al periodo della sinusoide. La schermata che si ottiene è raffigurata in Fig. 3.25.

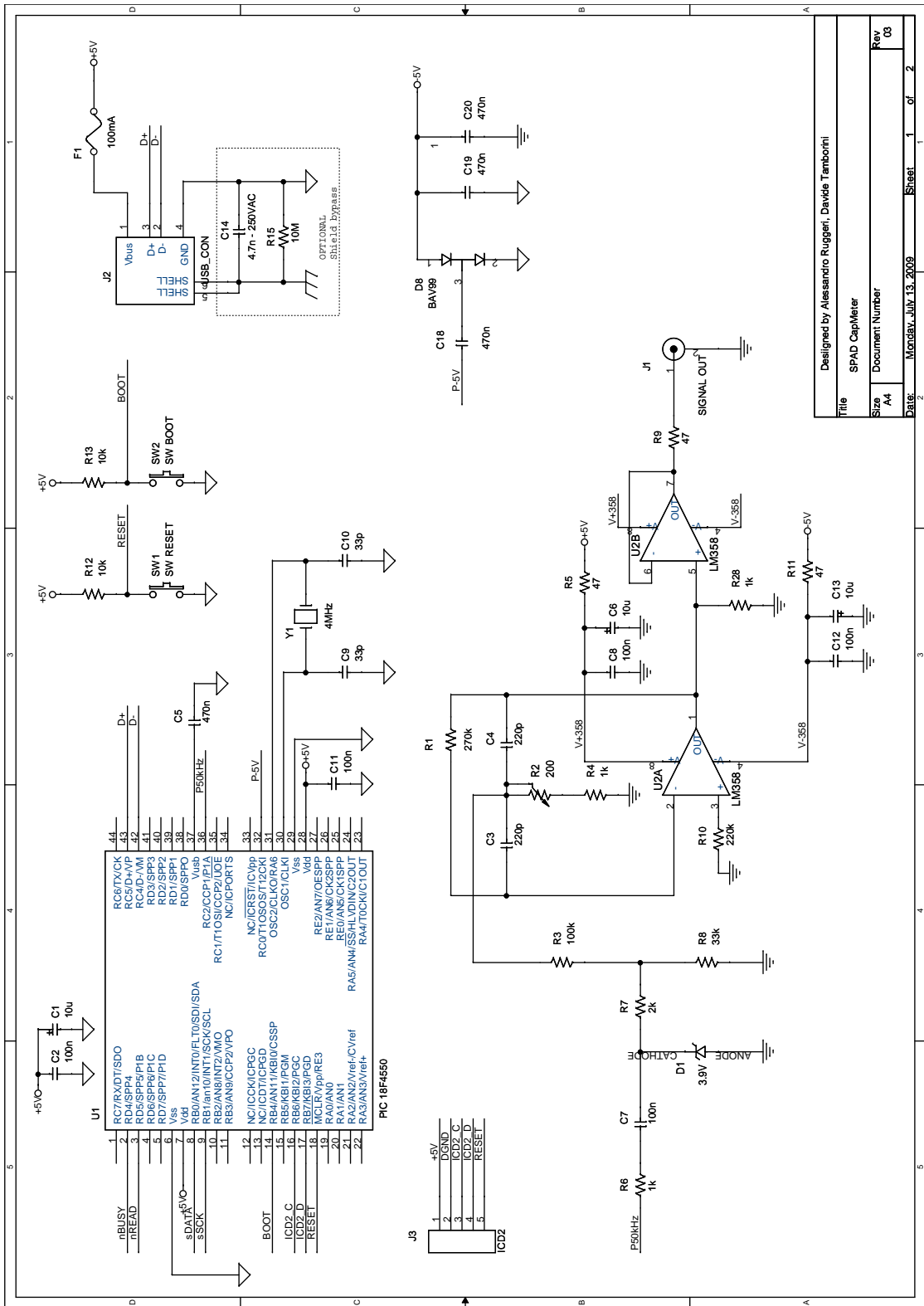
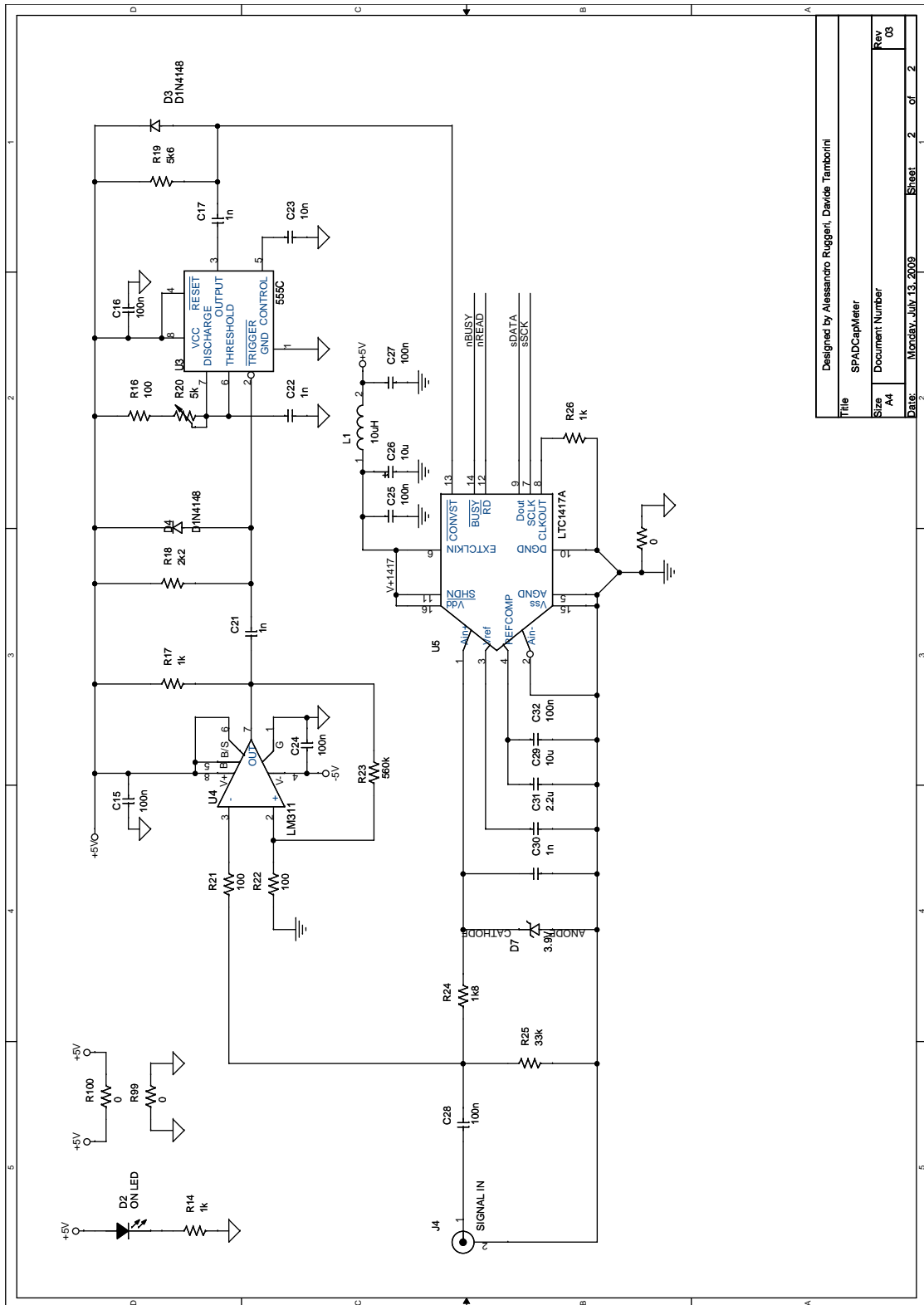


Figura 3.26: Schema Orcad con PIC, USB, alimentatore e filtro 50kHz



Designed by Alessandro Ruggieri, Davide Tamborini	
Title SPADCapMeter	
Size A4	Document Number
Date: Monday, July 13, 2009	Sheet 2 of 2
Rev 03	

Figura 3.27: Schema Orcad catena di acquisizione

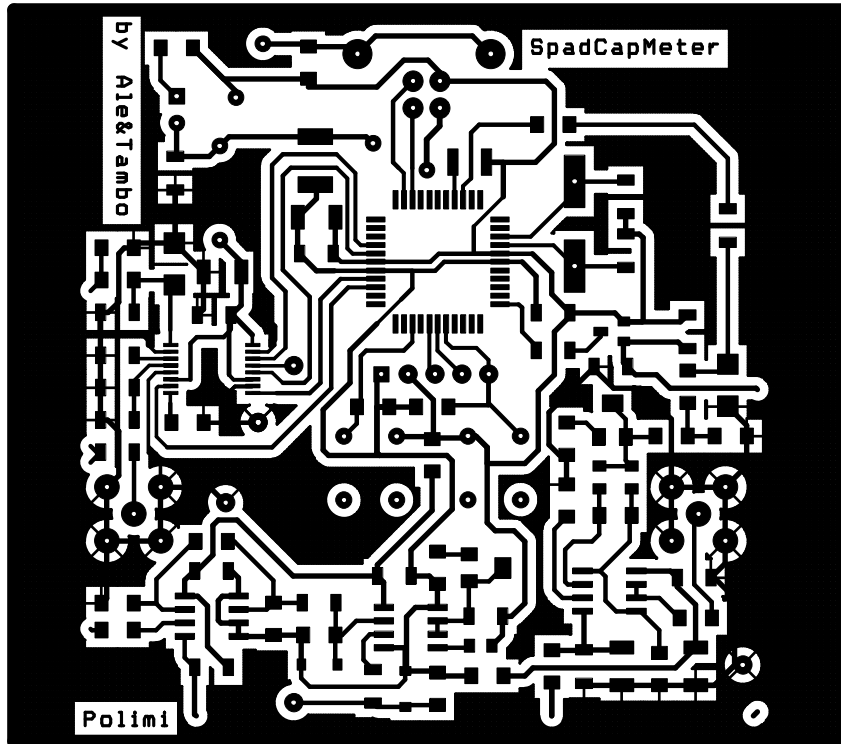


Figura 3.28: Top del Layout

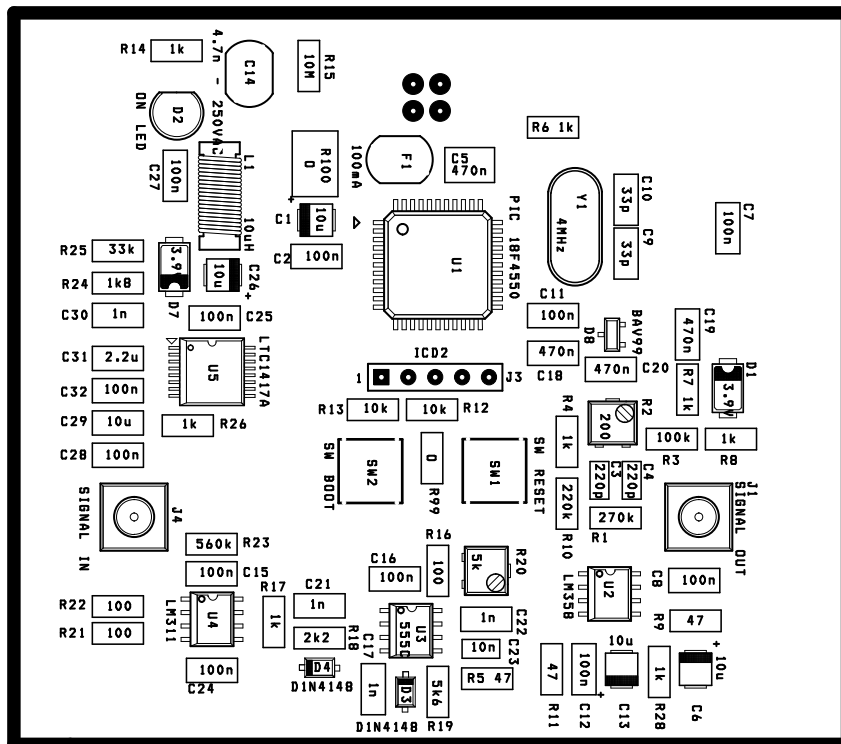


Figura 3.29: Assembly Top del Layout

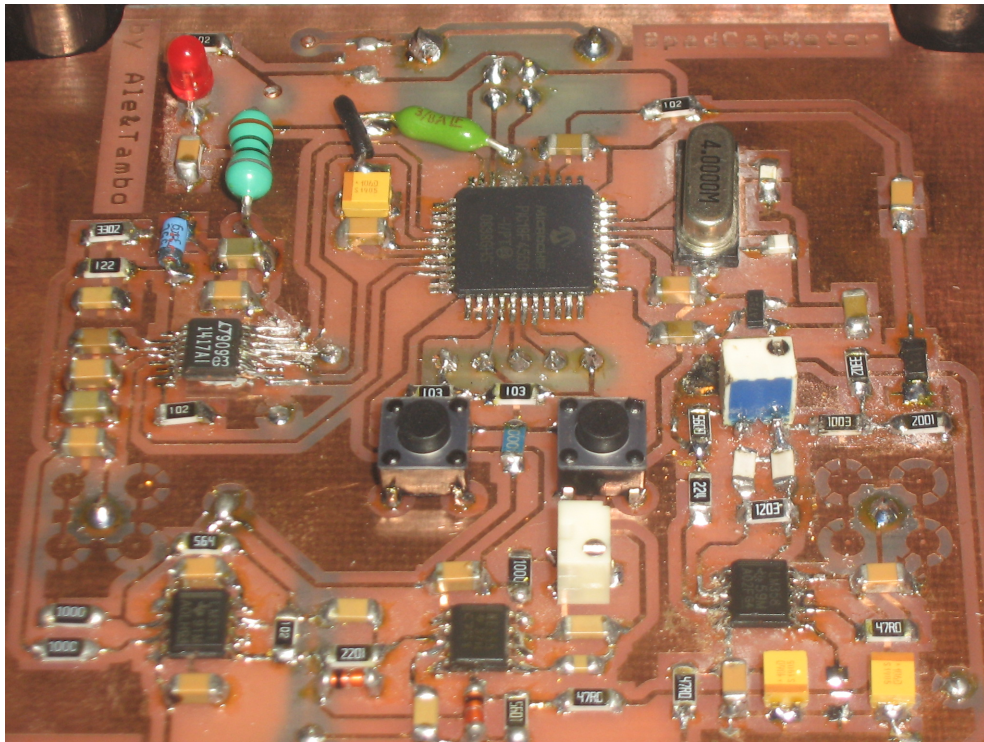


Figura 3.30: Scheda realizzata vista da sopra

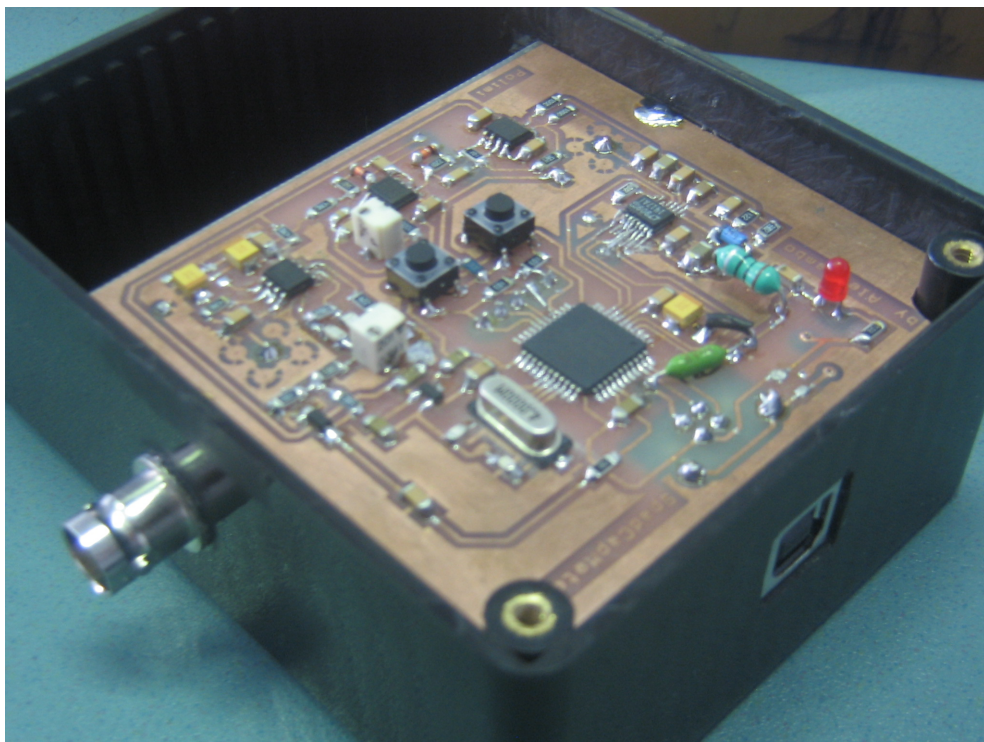


Figura 3.31: Scheda realizzata vista da davanti

Capitolo 4

Software

Terminato il progetto della sezione hardware, si deve sviluppare il firmware del microcontrollore e un opportuno software per la gestione dello strumento da PC.

4.1 Microcontrollore

Per prima cosa viene sviluppato il programma per la gestione del PIC18F4550. Esso infatti deve occuparsi delle seguenti operazioni:

- gestire la comunicazione verso il computer tramite l'interfaccia USB;
- leggere il dato dal convertitore A/D via SPI;
- elaborare digitalmente il segnale;
- configurare opportunamente i moduli hardware interni al PIC.

Per scrivere il firmware si preferisce usare il linguaggio C al posto dell'assembler, sia per la sua semplicità che per vantaggi intrinseci al dispositivo scelto. Esso infatti, nonostante sia RISC, presenta delle istruzioni ottimizzate per l'esecuzione di un codice generato da un compilatore C. Il tool di sviluppo utilizzato è l'MPLAB, mentre il compilatore è il C18[9], entrambi della Microchip Technology. Inoltre tale azienda, distribuisce dei codici di base per lo sviluppo di applicazioni USB basate sui microcontrollori della famiglia 18Fxxxx; queste verranno utilizzate in modo proficuo durante la scrittura del firmware.

4.1.1 Protocolli di comunicazione

USB

A questo punto prima di illustrare il funzionamento del software che gestisce la comunicazione col PC, si presentano alcune peculiarità dell'interfaccia USB.

Come spiegato in [5], le comunicazioni seriali di tipo RS232 largamente utilizzate in passato, stanno oramai scomparendo a favore dell'USB. Dal momento che si vuole realizzare uno strumento utilizzabile su qualsiasi calcolatore, la scelta ricade appunto su questa tecnologia più recente, sebbene più complessa da gestire.

Essa consiste ancora in una comunicazione di tipo seriale ma al posto di essere punto a punto è organizzata a stella, dove una serie di dispositivi *slaves* sono connessi ad un unico *master* (o *host*). Questo significa che una periferica non può di sua iniziativa trasmettere dati ma deve essere l'host a richiederli[10]. Nel caso preso in esame il dispositivo master è il PC che può gestire anche altre periferiche USB oltre allo SPADCAPMETER. La comunicazione tra i dispositivi è effettuata mediante *pipes* (canali logici) ai quali corrispondono dei relativi *end-point* nelle periferiche. Ciascuno di essi può trasferire dati in una sola direzione. Quando un dispositivo è connesso ad un host, parte una procedura di enumerazione; questa consiste nell'assegnazione da parte del master di un indirizzo per la periferica. Se il dispositivo è supportato dall'host allora avviene un'installazione dei driver e infine esso è abilitato a trasmettere.

Un vantaggio consiste nel fatto di essere *plug and play* ovvero è possibile connettere nuove periferiche mentre il PC è in funzione.

Inoltre sul cavo sono presenti fili che permettono di alimentare periferiche a bassa potenza (5 volt con al massimo 500 mA). In questo modo si possono evitare gli alimentatori esterni.

Esistono attualmente tre tipi di connessione USB:

- connessione *low speed* (USB 1.0), con velocità di 1.5 Mbit/s;
- connessione *full speed* (USB 1.1), con velocità di 12 Mbit/s;
- connessione *high-speed* (USB 2.0), con velocità di 480 Mbit/s.

ovviamente tutti questi standard sono retrocompatibili. Il microcontrollore scelto è in grado di lavorare in modalità full speed. Il segnale trasmesso sul canale trasmissivo è di tipo differenziale e con livelli logici di 3.3 V. La scelta della velocità di trasmissione è selezionata variando l'impedenza di terminazione della linea.

La comunicazione sul bus avviene mediante pacchetti di dati. Inizialmente i pacchetti sono inviati dall'host a tutti i dispositivi ma solo il dispositivo interessato alla comunicazione risponde. Dopo questa prima sincronizzazione parte la comunicazione vera e propria. Tutti i pacchetti sono composti da bytes da 8 bit salvati come *little endian*. I primi quattro bytes di un pacchetto consistono nel *Packet Identifier* (PID). Esistono diversi tipi di pacchetti:

- *handshake packets*, contiene solo il PID ed è usato in risposta ai *data packet*;
- *token packets*, contengono il PID, l'indirizzo del dispositivo e una parte di controllo di errore, sono mandati solo dall'host per far trasmettere allo slave dei dati oppure per configurarlo;
- *data packets*, usati per trasmettere dati.

Esistono infine quattro modalità di trasferimento dei dati:

- *control transfers*, usato tipicamente per i comandi, consiste nell'inviare un token, un solo pacchetto di dati e un handshake. E' indicato per il trasferimento di pochi dati.
- *interrupt transfers*, nel quale ciclicamente l'host richiede alla periferica se ha qualche dato da trasmettere.
- *isochronous transfers*, usato per trasferire un flusso di dati in modo continuo.
- *bulk transfers*, usato per il trasferimento di grandi volumi di dati a scatti.

SPI

Il Serial Peripheral Interface o SPI è un bus di comunicazione seriale sincrono full-duplex, utilizzato soprattutto nella comunicazione tra microcontrollore ed al-

tri circuiti integrati, a breve distanza. La trasmissione è bidirezionale ed è controllata da un master che emette il segnale di clock e decide quando iniziare la comunicazione. Questo sistema è definito anche a quattro fili in quanto oltre all'alimentazione e massa ci sono quattro linee di dati. Essi sono:

- SCLK: Serial Clock;
- SDI (o MISO): ingresso dei dati per il master, uscita per gli slaves;
- SDO (o MOSI): uscita dati per il master, ingresso per gli slaves;
- SS: slave select, segnale di abilitazione dello slave, per fare in modo che quando esso è disabilitato abbia la linea SDI in tristate.

La comunicazione avviene semplicemente mediante uno shift dei dati attraverso un registro a scorrimento il quale permette la conversione di un dato da parallelo a seriale e viceversa. I pin SDI e SDO sono quindi connessi al primo e all'ultimo flip-flop di questo registro che sarà peraltro accessibile in parallelo dall'interno della periferica.

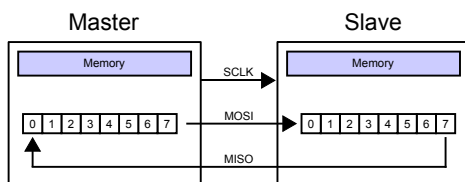


Figura 4.1: Collegamento tra due periferiche SPI

Lo standard prevede due possibili impostazioni: una (CPHA) sul fronte del clock in cui avviene lo shift dei dati e l'altra (CPOL) sulla polarità del clock, ovvero sul suo stato di riposo. Queste opzioni, in genere, sono impostabili sul dispositivo ma-

ster e permettono di adattarlo a tutte le possibili varianti di dispositivi slaves.

Esistono due tipologie principali di collegamento:

1. slaves controllati indipendentemente, con linee SCLK, SDI e SDO che vanno a tutte le periferiche le quali vengono poi singolarmente abilitate dal master con tante linee di SS quanti sono gli slaves.
2. slaves in catena, la linea SCLK va a tutti gli slaves, mentre SDO va solo al primo, la sua SDI va all'SDO del secondo slaves e così via, per formare un anello.

4.1.2 Programma PIC

Prima di iniziare a scrivere il firmware è bene realizzare uno schema a blocchi che focalizzi i punti principali su cui lavorare. In questo progetto ciò si traduce nel seguente elenco di operazioni:

- Inizializzazione, suddivisa in:
 - Inizializzazione USB, dove si impostano i parametri di configurazione del modulo hardware dedicato alla comunicazione col PC.
 - Altre inizializzazioni, dove vengono settati tutti i registri degli altri moduli utilizzati.
- Ciclo infinito, nel quale si eseguono:
 - USBTask, il quale si occupa della parte software della gestione del protocollo USB.
 - Process IO, nella quale:
 - * StateMACHINE, è una macchina a stati che gestisce l'acquisizione e l'elaborazione dei dati dall'ADC.
 - * Service Request, risponde ai comandi ricevuti tramite USB.

In particolare le parti di codice di maggiore interesse sono riportate qui di seguito.

Innanzitutto si analizza il codice del `main()`, il quale rispecchia fedelmente la struttura riportata nello schema appena visto:

Codice 4.1: *main*

```
void main(void)
2 {
    InitializeSystem();
4
    while(1){
6        USBDeviceTasks();
        ProcessIO();
8    } //end while
}
```

InitalizeSystem

La funzione `InitializeSystem()` richiama al suo interno `USBDeviceInit()` e `UserInit()`; la prima è una funzione inclusa nel pacchetto Microchip per la gestione dell'USB e permette, come anticipato, di configurare il blocco hardware USB per la corretta ricezione/trasmissione dei dati. La seconda invece è riportata in 4.2.

Codice 4.2: UserInit

```

void UserInit(void){
2   stato = NOP;
   M.Val = 0;
4   Mcompleted = TRUE;
   //---- SPI -----/
6   SSPSTAT = 0b01000000;
   SSPCON1 = 0b00100000;
8   // Enabling SPI I/O:
   DIR_SCLK = OUTPUT_PIN;
10  DIR_SDATOut = OUTPUT_PIN;
   DIR_LTC1417_nBUSY = INPUT_PIN;
12  DIR_LTC1417_nRD = OUTPUT_PIN;
   //---- -5V GENERATOR -----/
14  DIR_CLKxM5V = OUTPUT_PIN;
   //---- TIMER2 & PWM FOR 50kHz GENERATOR -----/
16  PR2 = 59;
   SetDCPWM1(120);
18  DIR_SIN50kHz = OUTPUT_PIN;
   OpenTimer2(T2_POST_1_1 & T2_PS_1_4);
20  CCP1CON |= 0b00001100;           // CCP1=PWM1
}

```

`stato` è la variabile che controlla la macchina a stati che gestisce l'acquisizione dei dati; essa è impostata inizialmente a `NOP`, ovvero `NoOperation`. `M.Val` è la variabile in cui si salvano i risultati delle misure e `Mcompleted` è un flag che segnala il termine della misura.

Si passa quindi alla configurazione dei moduli[8]. Per quanto riguarda la configurazione del modulo SPI si devono impostare due registri: `SSPSTAT` e `SSPCON1`. Per quanto riguarda il primo si focalizza l'attenzione sui bit 7 e 6 che permettono di impostare la polarità di clock (`CPOL`) e la fase (`CPHA`); queste devono essere le stesse sia per il PIC che per l'ADC.

- `SMP=0`: il dato è campionato al centro del tempo di bit;
- `CKE=1`: la trasmissione avviene sul fronte di discesa del clock.

in seguito si impostano i pin corrispondenti all'SPI settandoli come ingressi o uscite.

Per creare il segnale ad onda quadra a 50kHz e duty-cycle del 50% da inviare poi al filtro che genera la sinusoide di eccitazione per il “Capacimetro per diodi polarizzati”, si devono impostare il `Timer2` e il modulo `PWM`. Come anticipato nella sezione hardware, il timer gestisce il periodo dell'onda mentre il `CCP` il duty-cycle.

Il periodo è dato da questa formula:

$$PWM_{period} = [PR2 + 1] \cdot 4 \cdot T_{OSC} \cdot TMR2 \text{ Prescale Value} \quad (4.1)$$

Siccome serve una frequenza di 50 kHz e quindi un periodo di 20 μ s, avendo a disposizione un $T_{OSC} = 1/48 \text{ MHz} = 20,83 \text{ ns}$, scegliendo un prescaler di 4, si ottiene il valore di `PR2` ovvero il registro di periodo del `Timer2`:

$$PR2 = \frac{PWM_{period}}{4 \cdot T_{OSC} \cdot TMR2 \text{ Prescale Value}} - 1 = 59 \quad (4.2)$$

Siccome serve un duty-cycle del 50%, il tempo T_{ON} dell'onda quadra sarà di 10 μ s. Per impostare il modulo `PWM` si usa la seguente formula:

$$T_{ON} = CCPRxL:CCPxCON<5:4> \cdot T_{OSC} \cdot TMR2 \text{ Prescale Value} \quad (4.3)$$

da cui si ricava che

$$CCPRxL:CCPxCON<5:4> = \frac{T_{ON}}{T_{OSC} \cdot TMR2 \text{ Prescale Value}} = 120 \quad (4.4)$$

Per impostare questo valore è stata utilizzata la funzione `SetDCPWM1()`; ora si può impostare il pin di uscita del modulo ed abilitare il timer tramite la funzione `OpenTimer2()` a cui bisogna passare i valori di postscale e di prescale (1 e 4). Infine attraverso il registro `CCPxCON` si configurano i bit `CCPxM3:CCPxM0` per selezionare il tipo di modalità di funzionamento del modulo `CCP`; in particolare il codice 1100 rappresenta la modalità `PWM`.

SPAD StateMachine

Per acquisire i dati dall'ADC ed elaborarli si è pensato di creare una macchina a stati; questa è costituita da una serie di stati, che rappresentano una specifica operazione da eseguire, e da collegamenti che indicano il flusso del programma.

Il principale vantaggio di questa struttura è quella di evitare l'uso di "funzioni bloccanti", cioè funzioni che interrompono l'esecuzione del programma finchè non si verifica uno specifico evento. Ciò comporterebbe che durante questa attesa il microcontrollore diventerebbe insensibile agli altri avvenimenti. Inoltre nel caso specifico di questa applicazione, il software per la comunicazione USB necessita che la funzione `USBDeviceTasks()` sia richiamata circa ogni 10 ms. La struttura a macchina a stati è invece dinamica, in quanto controlla ciclicamente il verificarsi degli eventi.

Lo stato base di questa macchina si chiama `NOP` e rappresenta lo stato ozioso. Per la lettura dell'ADC si è predisposto lo stato `ADC_START_MEASURE` che effettua semplicemente la lettura dell'ultimo valore acquisito dall'ADC. Questa misura può risultare talvolta inaccurata perciò, per evitare di leggere valori spuri e per aumentare la risoluzione apparente del convertitore, si è pensato di creare lo stato `ADC_IMP` e tutta una serie di stati ad esso collegati. Essi leggono più di un valore dall'ADC e ne fanno prima la moda tra `MAX_MODE` valori e poi la media su `MAX_MEAN` risultati. I due parametri massimi sono lasciati liberi al fine di poter scegliere il miglior compromesso tra risultati ottenuti e tempo per effettuare la misura completa.

Qui di seguito viene riportato il corrispondente codice C che fa da scheletro alla macchina a stati.

Codice 4.3: *StateMachine*

```
void SPAD_StateMachine(void){
2   switch(stato){
      case NOP:
4       //No Operation State
          break;
6       case ADC_START_MEASURE:
          ...
8       break;
      case ADC_IMP:
```

```

10     ...
11     break;
12 case ADC_IMP2:
13     ...
14     break;
15 case ADC_IMP_M:
16     ...
17     break;
18 default:
19     stato=NOP;
20     break;
21 }
22 }

```

Nel codice 4.4 è presentato lo stato che effettua una singola lettura dell'ADC. Innanzitutto si controlla che il convertitore non sia occupato. Se il test ha esito positivo si può abbassare il segnale di LTC1417_nRD per indicare una lettura all'ADC. Dopo una breve attesa per assicurarsi che i livelli logici si siano assestati e il convertitore sia effettivamente pronto ad inviare i dati, si esegue la lettura della parte alta del valore a 16 bit¹ risultante dalla conversione mediante la funzione ReadSPI e si salva il risultato nella parte alta della variabile M, definita come intero senza segno a 16 bit. Poi si procede con la medesima funzione a leggere la parte bassa e a memorizzarla nella stessa variabile. Infine si riporta il segnale di LTC1417_nRD alto e poi si controlla che il LTC1417_nBUSY sia ancora alto. Difatti se questo è basso vuol dire che l'ADC ha ricevuto dal circuito di front-end un nuovo StartOfConversion e quindi il dato presente nel suo registro di conversione potrebbe essere corrotto. Se questo non è accaduto la lettura del dato ha avuto buon fine, è quindi possibile settare il flag Mcompleted e ritornare allo stato NOP.

Codice 4.4: Stato ADC START MEASURE

```

case ADC_START_MEASURE:
2   if(LTC1417_nBUSY){
3       LTC1417_nRD=0;
4       Nop();          // wait data access time after nRD?
5       M.byte.HB=ReadSPI();
6       M.byte.LB=ReadSPI();
7       LTC1417_nRD=1;
8       if(LTC1417_nBUSY){ //nBUSY is still high
9           Mcompleted=TRUE;
10          stato=NOP;
11          }/*elseif          //there is a new conversion running

```

¹In realtà il convertitore è a 14bit ma gli ultimi due sono riempiti con zeri

```

12     */
14     }
    break;

```

Per quanto riguarda la misura più performante, come anticipato, si ha bisogno di più di uno stato. Il primo di questi è denominato `ADC_IMP` e si occupa principalmente dell'inizializzazione delle variabili utilizzate negli altri stati. Vengono svuotati gli array `M_mean` e `M_mode` e i due puntatori `k_mean` e `k_mode` vengono azzerati. Si passa quindi allo stato `ADC_IMP2`.

Codice 4.5: Stato ADC IMP

```

case ADC_IMP:
2     //clear temp array
    for(k_mean=0;k_mean<MAX_MEAN;k_mean++){
4         M_mean[k_mean]=0;
    }
6     for(k_mode=0;k_mode<MAX_MODE;k_mode++){
    M_mode[k_mode]=0;
8     }

10    k_mean=0;
    k_mode=0;
12    stato=ADC_IMP2;
    break;

```

Lo stato del codice 4.6 fa in modo di produrre un valore da salvare nella variabile `M_res` che è calcolato come la somma di `MAX_MEAN` valori, ottenuti a loro volta come la moda di `MAX_MEAN` acquisizioni dall'ADC. Questo permette di eliminare eventuali campioni inesatti (filtro mediano) e di fare un filtraggio passa basso (media). Se il contatore del numero di mode effettuate è minore di `MAX_MODE` allora si passerà allo stato `ADC_IMP_M`, analogo a `ADC_START_MEASURE`, il quale legge una singola misura dall'ADC, la salva in `M_mode[k_mode]`, e poi ritorna allo stato attuale avendo incrementato il contatore `k_mode`. Una volta raggiunto il massimo, il vettore `M_mode` viene ordinato mediante l'algoritmo *Insertion Sort* e viene salvato il valore di mezzo nell'array `M_mean` alla posizione `k_mean`, che viene successivamente incrementata. Una volta che anche `k_mean` raggiunge il suo valore massimo `MAX_MEAN`, allora si possono sommare tutti gli elementi del vettore `M_mean` e salvare il risultato in `M_res`. Viene settato il flag `Mcompleted` e lo stato passa a `NOP`.

Codice 4.6: *Stato ADC IMP2*

```

case ADC_IMP2:
2   if(k_mode<MAX_MODE){
      //launch another conversion
4   stato=ADC_IMP_M;
      }else{
6   //sort array and find the mode
      InsertionSort(M_mode,MAX_MODE);
8   M_mean[k_mean++]=M_mode[(MAX_MODE-1)/2];

10  k_mode=0;
      }

12  if(k_mean>=MAX_MEAN){
14  //calculate mean
      M_res=0;
16  for(k_mean=0;k_mean<MAX_MEAN;k_mean++){
          M_res+=M_mean[k_mean];
18  }
      Mcompleted=TRUE;
20  stato=NOP;
      }
}

```

Lo stato ADC_IMP_M del codice 4.7 come anticipato ha la stessa funzione di ADC_START_MEASURE, però al posto di salvare il risultato in M lo salva in M_mode[k_mode] e incrementa quest'ultimo contatore. Infine ritorna allo stato ADC_IMP2 da cui è stato richiamato.

Codice 4.7: *Stato ADC IMP M*

```

case ADC_IMP_M:
2   if(LTC1417_nBUSY){
      LTC1417_nRD=0;
4   Nop(); // wait data access time after nRD?
      M.byte.HB=ReadSPI();
6   if(LTC1417_nBUSY){
          M.byte.LB=ReadSPI();
8   LTC1417_nRD=1;
          if(LTC1417_nBUSY){
10  M_mode[k_mode]=M.Val;
          k_mode++;

12  stato=ADC_IMP2;
          }
14  }
      }
16  }
break;

```

ServiceRequests

La presente funzione (codice 4.8) serve per rispondere ai comandi inviati dal PC attraverso l'interfaccia USB. Il software implementato da Microchip nella funzione USBDeviceTasks() insieme al modulo hardware si occupa infatti di gestire tutto il protocollo e restituisce all'utente direttamente i dati ricevuti. Con l'istruzione `if(!USBHandleBusy(USBGenericOutHandle))` si controlla se sono arrivati nuovi dati. Se si, questi vengono analizzati nel successivo blocco `switch-case` e si fornisce una risposta a ciascuna richiesta da parte del PC.

Codice 4.8: Funzione ServiceRequest

```

void ServiceRequests(void){
2   BYTE index;

4   //Check to see if data has arrived
   if(!USBHandleBusy(USBGenericOutHandle)){
6       //if the handle is no longer busy then the last
       //transmission is complete

8       counter = 0;

10      INPacket.CMD=OUTPacket.CMD;
12      INPacket.len=OUTPacket.len;

14      //process the command
       switch(OUTPacket.CMD){
16
18         case READ_IMP:
           if(Mcompleted){
20             //conversion completed
             INPacket._byte[2]=0xAA;

22             //send previous measure
             M.Val=(unsigned int)(M_res/MAX_MEAN);
24             INPacket._byte[3]=M.byte.HB;
             INPacket._byte[4]=M.byte.LB;
26             counter=0x05;

28             //read new measure
             Mcompleted=FALSE;
30             stato=ADC_IMP;
           }else{
32             //conversion is still in progress
             INPacket._byte[2]=0xBB;
34             counter=0x03;
           }
36       break;

```

```

38     case READ_SINGLE:
39         if(Mcompleted){
40             //conversion completed
41             INPacket._byte[2]=0xAA;
42
43             //send previous measure
44             INPacket._byte[3]=M.byte.HB;
45             INPacket._byte[4]=M.byte.LB;
46             counter=0x05;
47
48             //read new measure
49             Mcompleted=FALSE;
50             stato=ADC_START_MEASURE;
51         }else{
52             //conversion is still in progress
53             INPacket._byte[2]=0xBB;
54             counter=0x03;
55         }
56         break;
57
58     case RESET:
59         Reset();
60         break;
61
62     default:
63         Nop();
64         break;
65 }//end switch()
66
67 if(counter != 0){
68     if(!USBHandleBusy(USBGenericInHandle)){
69         USBGenericInHandle = USBGenWrite(USBGEN_EP_NUM, __
70             (BYTE*)&INPacket, counter);
71     }
72 }//end if
73
74 //Re-arm the OUT endpoint for the next packet
75 USBGenericOutHandle = USBGenRead(USBGEN_EP_NUM, __
76     (BYTE*)&OUTPacket, USBGEN_EP_SIZE);
77 }//end if
78 }//end ServiceRequests

```

Si commenta in particolare il caso `READ_SINGLE` (riga 40): se il precedente ciclo di misura è stato completato (`Mcompleted==1`) allora si imposta il secondo byte del pacchetto da spedire al PC (`INPacket`) con la costante `0xAA` scelta arbitrariamente per indicare l'avvenuta conversione. Se questa non è ancora terminata si imposta

a `0xBB`. Il valore corrispondente alla misura viene poi spezzato in due parti e messo nel terzo e nel quarto byte del pacchetto. La variabile `counter` viene impostata a 5 (lunghezza del pacchetto + 1) e questa è utile al software per l'invio dei pacchetti. A questo punto si può resettare il flag di avvenuta conversione e impostare la macchina a stati per effettuare una nuova conversione `stato=ADC_START_MEASURE`.

Tutto quello appena detto per `READ_SINGLE` si può estendere a `READ_IMP`, visto che i due casi sono praticamente identici tra loro tranne per il dato trasmesso e lo stato caricato. In questo caso infatti viene trasmesso `M_res/MAX_MEAN`, in quanto `M_res` è la somma di più valori misurati e si impone `stato=ADC_IMP`.

Esiste poi il caso `RESET`, di immediata comprensione e il caso di default utile nel caso il pacchetto ricevuto non sia uno di quelli previsti.

A questo punto il pacchetto costruito può essere passato nella *Dual Port RAM* e spedito al PC mediante l'uso della funzione integrata `USBGenWrite()`. Infine l'endpoint di ricezione può essere riarmato per ricevere nuovi pacchetti in arrivo.

4.2 LabVIEW™

Per quanto riguarda il software di gestione dello strumento, un ottimo programma è LabVIEW™. Esso infatti presenta delle caratteristiche di elevata comodità e versatilità nell'interfacciamento con periferiche di ogni tipo. Prima di parlare dello sviluppo vero e proprio di tale software, è necessario capire cos'è e come funziona tale programma.

4.2.1 Introduzione a LabVIEW™

LabVIEW™ (Laboratory Virtual Instrument Engineering Workbench) è un ambiente di sviluppo per applicazioni principalmente orientate all'acquisizione di dati ed alla gestione di strumentazione e elettronica, all'analisi ed elaborazione dei segnali.

La particolarità di LabVIEW™ è quella di fornire un ambiente di programmazione di tipo grafico ad oggetti denominato "*G - Language*", il quale consente di realizzare algoritmi utilizzando icone ed altri oggetti grafici uniti da linee di col-

legamento in modo da realizzare una sorta di diagramma di flusso. Tutto questo sostituisce il testo del codice degli altri linguaggi di programmazione conservando comunque molte similitudini ad essi: difatti presenta tutti i tipi di dati e gli operatori predefiniti di uso comune, permette di generare nuovi tipi di dati combinando tra loro i tipi di dati elementari e di controllare l'esecuzione dei programmi ricorrendo a strutture di controllo di flusso.

Un'ulteriore differenza dai tradizionali linguaggi di programmazione consiste nel fatto che è un programma *data-driven*, ovvero non segue il flusso temporale delle operazioni, bensì il flusso dei dati, enfatizzando come i dati si muovono tra i vari blocchi operativi.

LabVIEW™ mette inoltre a disposizione del programmatore una serie di librerie di funzioni che possono essere richiamate ed utilizzate all'interno dei programmi, che comprendono funzioni di uso comune, funzioni specializzate per l'acquisizione e l'elaborazione dei segnali, controllo di strumentazione numerica, trasmissione di dati mediante vari protocolli di comunicazione. E' possibile inoltre definire nuove funzioni ed arricchire le librerie preesistenti.

LabVIEW™ è quindi principalmente pensato per il controllo di schede di acquisizione e di strumentazione connessa al calcolatore. L'ambiente di sviluppo consente di costruire programmi i quali prendono il nome di *Virtual Instrument*, VI, che permette l'interazione tra calcolatore e strumentazione fornendo contemporaneamente all'utente un opportuno pannello frontale grafico per il dialogo con il VI stesso.

4.2.2 Programma LabVIEW™

Per gestire al meglio lo strumento di misura vengono individuate cinque sezioni principali.

Misura Singola

E' la sezione dedicata all'acquisizione continua del valore della capacità ottenuta mediante il diagramma illustrato in Fig. 4.2. A seguito di un comando di *START*, scegliendo su quanti valori acquisiti si vuole effettuare una media, il blocco Get-

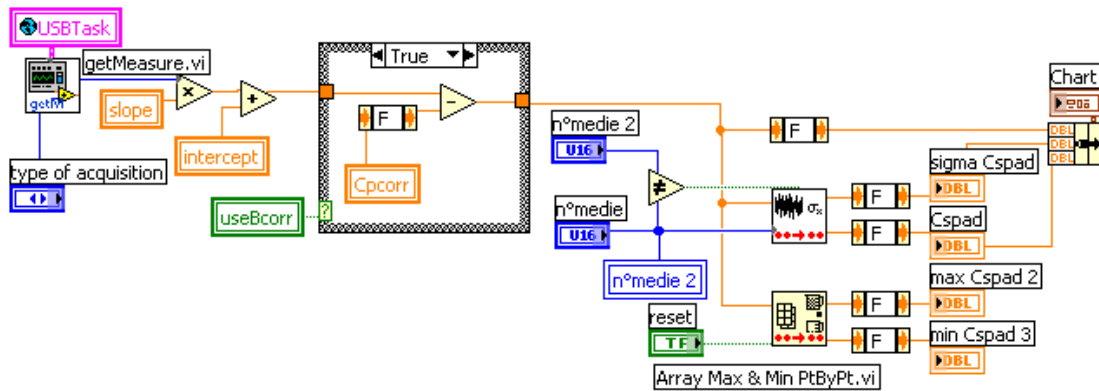


Figura 4.2: Diagramma LabVIEW™ acquisizione singola misura

Measure.vi acquisisce il dato dal PIC che attraverso i coefficienti ottenuti nella sezione *TARATURA* converte in capacità. Si può abilitare un comando che sottrae al valore acquisito la capacità di correzione dovute ai parassitismi. Tale valore viene mandato ad un blocco che calcola la media e la varianza su un numero di dati impostati da un ulteriore controllo, ed entrambi poi visualizzati in un grafico (blocco *Chart*). Contemporaneamente il blocco *Array Max & Min PtByPt* memorizza il massimo e il minimo dei valore acquisiti. Questi valori, insieme alla media e alla deviazione standard, vengono visualizzati su label.

Cparassite

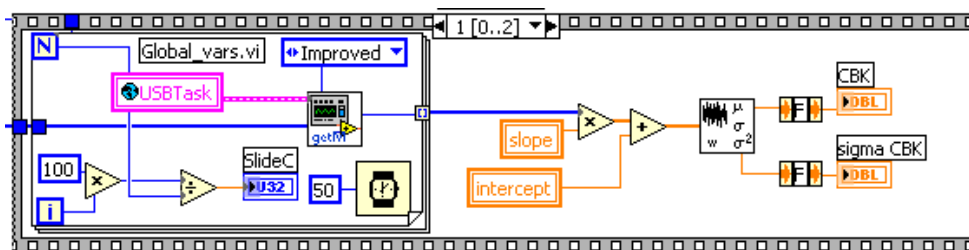


Figura 4.3: Diagramma LabVIEW™ misura capacità parassite

E' la sezione dedicata alla misurazione delle capacità parassite. Si possono misurare le tre capacità C_{AK} , C_{BA} e C_{BK} (tenendo fissa la tensione di polarizzazione) seguendo il diagramma di flusso illustrato in Fig. 4.3. Qui vengono ricevuti dati dal PIC in modalità Improved tante volte quante impostate dall'utilizzatore.

Attraverso i risultati ottenuti dalla taratura si converte questo dato esadecimale in valore di capacità. Si effettua la media di tutti questi dati acquisiti ottenendo il valore della capacità sotto esame e la relativa standard deviation. Una volta terminate le misurazioni, bisogna risolvere il sistema illustrato in eq. 1.11 per determinare i valori dei parassitismi C_{ba} e C_{bk} . Per fare questo è stato risolto il sistema in forma simbolica attraverso questi comandi in MATLAB:

Codice 4.9: *Risoluzione sistema capacità*

```

2 | syms CAK CBA CBK Cak Cba Cbk
  |
  | A=-CAK+Cak+Cba*Cbk/(Cba+Cbk);
  |
4 | B=-CBA+Cba+Cak*Cbk/(Cak+Cbk);
  |
  | C=-CBK+Cbk+Cak*Cba/(Cak+Cba);
  |
6 | S=solve(A,B,C,Cak,Cba,Cbk)

```

I risultati ottenuti vengono inseriti in un VI che ricevuti in ingresso i tre valori misurati, restituisce i valori delle tre capacità reali.

In particolare si nota che determinati C_{ba} e C_{bk} , si può ottenere:

$$C_{ak} = C_{AK} - \frac{C_{ba} \cdot C_{bk}}{C_{ba} + C_{bk}} \quad (4.5)$$

e quindi si può calcolare il coefficiente di correzione da usare nella sezione della caratteristica dello SPAD.

$$C_{pcorr} = \frac{C_{ba} \cdot C_{bk}}{C_{ba} + C_{bk}} \quad (4.6)$$

Tali dati, oltre ad essere visualizzati nell'interfaccia si possono salvare in un apposito file. Inoltre si possono caricare dati ottenuti da misurazioni precedenti.

Misura Caratteristica

Questa sezione è dedicata alla rilevazione della curva caratteristica dello SPAD. Per far ciò bisogna impostare dei controlli. In particolare bisogna definire tutti i valori di tensione in corrispondenza dei quali si vuole conoscere la capacità dello SPAD. Una volta settati tutte queste impostazioni, prima di iniziare l'acquisizione, viene visualizzato un messaggio per consentire all'operatore di verificare che il

fotodiode sia polarizzato alla stessa tensione indicata dal programma. La prima parte di acquisizione ed elaborazione del singolo dato è uguale a quella usata nella sezione *Misura Singola*, ma si differenzia dal fatto che come misura si ottiene la media su un numero prestabilito dall'utente di dati acquisiti. Tale valore si ottiene a seguito di altre impostazioni. Alla fine di ogni misura si salva in un array il valore acquisito e la corrispondente tensione impostata. Questi dati vengono mandati al successivo blocco con la funzione di visualizzarli su un grafico. Infine tali dati possono essere salvati in file di testo oppure come file immagine.

Taratura

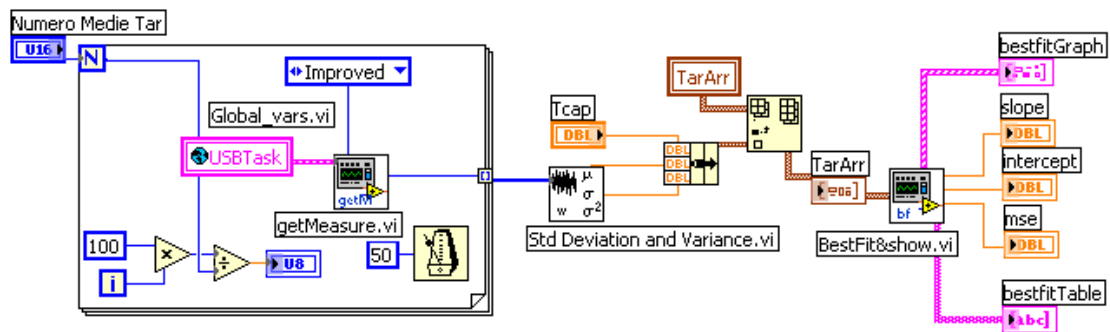


Figura 4.4: Diagramma LabVIEW™ taratura

E' la sezione dedicata alla taratura dello strumento. Essa è realizzata inserendo il valore (noto) della capacità che si sta misurando. Avviando l'acquisizione, cioè premendo il tasto *Tara*, il VI attribuisce al codice letto la capacità imposta seguendo il diagramma di flusso in Fig. 4.4. In particolare *USBTask* rappresenta la porta USB usata, il blocco *Get measure.vi* invia la richiesta del tipo di dato che vuole ottenere dal PIC (Normal o Improved) e lo acquisisce. Tale operazione viene ripetuta tante volte quanto è fissato da *Numero Medie Tar*, poi i dati ottenuti vengono elaborati da *Std Deviation and Variance.vi* che calcola media e deviazione standard. Successivamente tali valori ottenuti vengono memorizzati in un array insieme al corrispondente valore della capacità nota *Tcap*. Ripetendo tale operazione con nuovi valori di capacità, i dati contenuti nell'array vengono mandati al blocco *BestFit&show.vi* che li visualizza in una tabella, calcola i coef-

ficienti della retta interpolante e mostra su un grafico i dati e tale retta. Ci sono inoltre dei controlli aggiuntivi, cioè un tasto per cancellare tutte le misure, un tasto per cancellare un solo valore, un tasto per salvare i valori e un tasto per caricare valori precedentemente salvati. Come risultato si ottiene una label con i coefficienti della retta che interpola i dati ottenuti. Tali coefficienti vengono usati nelle altre sezioni per visualizzare il dato come capacità.

Opzioni

Infine viene prevista una sezione dedicata all'impostazione di settaggi generali, quali i percorsi del file di taratura e del file contenente i valori dei parassitismi che il programma deve automaticamente caricare all'avvio, la scelta del formato con cui si vogliono esportare le immagini. In più si ha possibilità di testare il corretto funzionamento del programma, attivando la demo mode, ovvero una particolare modalità in cui, senza collegare lo SPADCAPMETER , il programma legge dati di prova simulati internamente.

Capitolo 5

Istruzioni per l'uso

Realizzato il dispositivo sia nella parte hardware che in quella software, si hanno quasi tutti gli strumenti per effettuare le misure.

5.1 Collegamento e driver

Come prima cosa si deve collegare lo SPADCAPMETER al PC mediante un cavo USB maschio-maschio con una connettore tipo B e uno tipo A. Come prima verifica della corretta connessione si nota che il LED si accende, inoltre compare su PC messaggio **Trovato nuovo hardware** ed automaticamente parte l'installazione guidata dell'hardware. I driver per far riconoscere la periferica al PC sono forniti insieme all'eseguibile del programma sul CD in allegato all'elaborato. A termine dell'installazione dello SPADCAPMETER si può controllare che tale periferica sia correttamente riconosciuta nella sezione **Pannello di controllo** → **Sistema** → **Hardware** → **Gestione periferiche** che visualizza una porta con il nome **SPAD CapMeter**.

A tal punto, con il doppio click sul file **setup.exe** che si trova nella cartella **SCM\Installer**, si installa il programma d'interfaccia. Si seguono le istruzioni e si riavvia il computer come richiesto dal programma. Appena si riaccende viene terminata l'installazione. Nell'elenco dei programmi si trova **SPADCapMeter- PC Interface** → **SCM** e si avvia. A questo punto se la scheda non è collegata viene dato un messaggio d'errore ma cliccando su **ok** si può usare ugualmente il programma.

5.2 Come effettuare la misura

Completate le operazioni precedentemente illustrate si ottiene lo strumento pronto a rilevare la capacità dello SPAD.

Il software si articola in più sezioni, ottenendo diverse modalità di misurazione. Per prima cosa, se non si dispone dello SPADCAPMETER, si deve andare nella sezione opzioni dove si può attivare la Demo Mode che permette di controllare il funzionamento del software, nonché di prendere confidenza con lo strumento.

5.2.1 Taratura strumento

La sezione *Taratura* è fondamentale in quanto fornisce tutti i parametri necessari per convertire il codice ricevuto dal PIC in valore di capacità in Farad. Per fare questo sono necessari alcuni condensatori di capacità nota e di ridotta tolleranza da inserire negli appositi tulipani del “Capacimetro per diodi polarizzati”. In Fig. 5.1 si mostra lo screenshot di questa parte.

Per prima cosa si imposta il numero di valori sul quale si deve effettuare la media per ottenere un valore il più possibile preciso attraverso il controller (2). Successivamente si inserisce, attraverso il controller (1), il valore del condensatore di prova inserito nel circuito. Premendo il pulsante (3) si acquisisce il valore, lo si salva in tabella e lo si visualizza in un grafico. Durante tale operazione una barra di stato ne visualizza il progresso. Nel caso un valore sia stato acquisito in modo scorretto si può cancellare la riga nel quale è contenuto attraverso il pulsante (4). Oppure si possono cancellare tutti i valori per fare una nuova taratura con il controller (5). Una volta acquisiti i valori, vengono calcolati i coefficienti della retta che interpola i dati. Essi vengono visualizzati nelle label *slope* e *intercept*. Per finire i pulsanti (6) e (7) permettono rispettivamente di caricare i dati salvati in precedenti misure e salvare i dati ottenuti con la taratura appena effettuata.

5.2.2 Cparassite

In questo pannello, mostrato in Fig. 5.2 si effettua la misura delle due capacità parassite C_{bk} e C_{ba} . La prima operazione è quella di fissare un valore della tensione

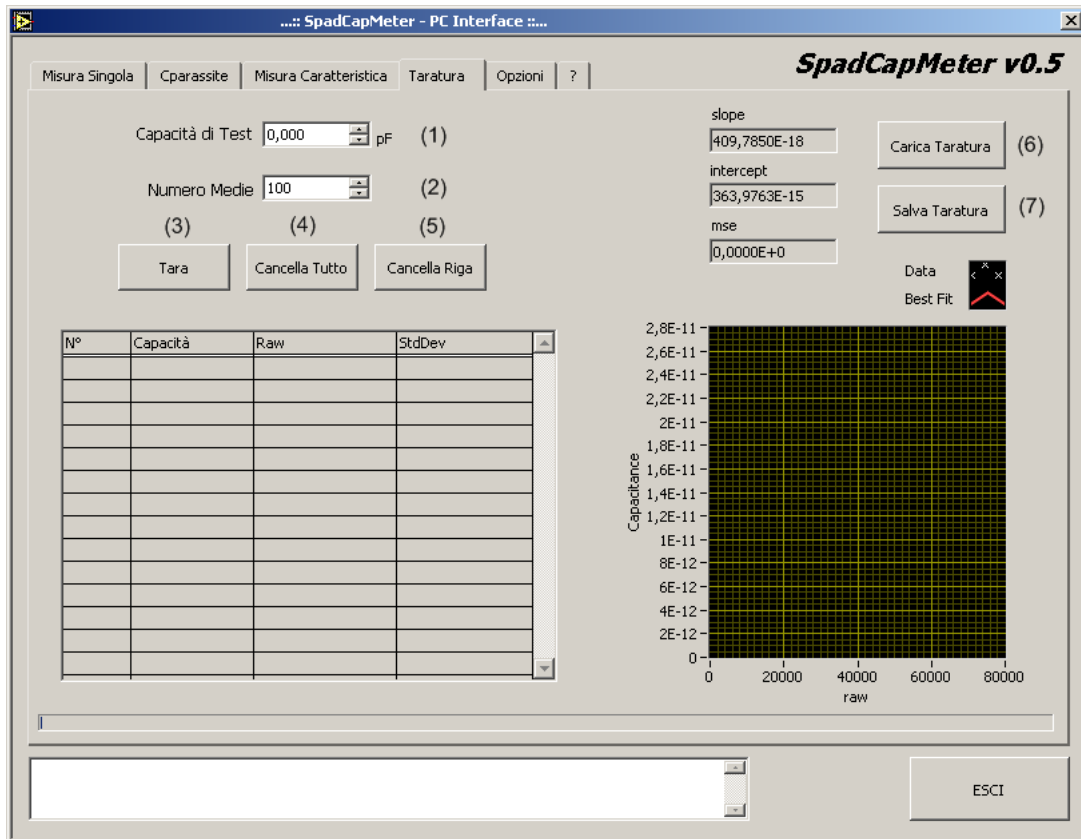


Figura 5.1: Schermata taratura

di polarizzazione e non modificarlo fino al termine di questa sessione di misure. Tramite il controller (4) si imposta il numero dei dati da mediare per ottenere il valore desiderato. Successivamente si inserisce nel circuito lo SPAD collegando i piedini B e K, poi si preme il tasto (1). Finita l'acquisizione vengono visualizzati nelle label *CBK* e *sigma CBK* il valore misurato e la sua incertezza. In seguito si svolgono le altre due acquisizioni misurando *CBA* mediante il controller (2) e *CAK* attraverso il pulsante (3), avendo cura di inserire lo SPAD in modo opportuno. Al termine di ciò vengono mostrate nelle label *Cbk* e *Cba* le capacità parassite. Questi due valori servono nelle sezioni successive. Infine ci sono i controller (5) che permettono di usare dati precedentemente salvati e il (6) per salvare quelli appena acquisiti e calcolati. L'uso di queste informazioni può essere disabilitato tramite un'opzione nella scheda Opzioni.

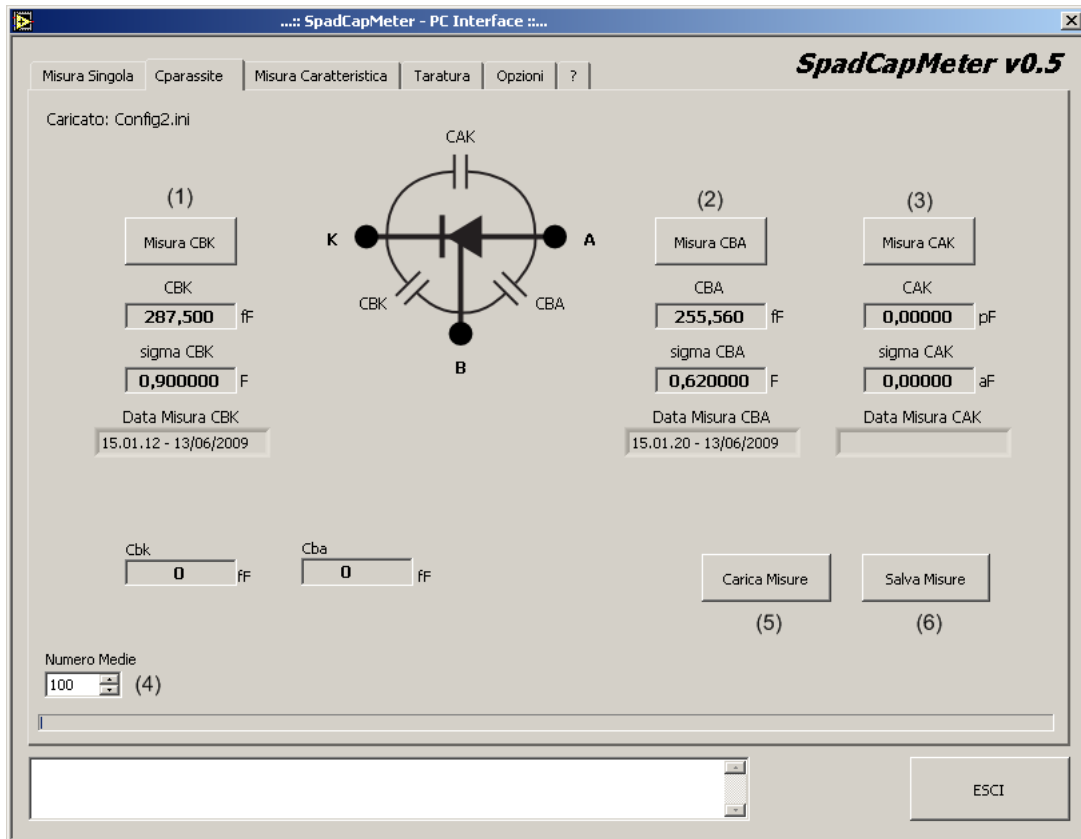


Figura 5.2: Schermata capacità parassite

5.2.3 Acquisizione singolo valore

Nella sezione *Misura Singola*, mostrata in Fig. 5.3 per prima cosa si deve impostare con il controller (1) il numero di valori N sul quale fare la media. Premendo il tasto (2) si fa partire l'acquisizione. Sul grafico vengono visualizzati come punti i singoli valori acquisiti, con una linea rossa la media degli ultimi N e in blu la loro deviazione standard. Infine nella label C_{spad} viene visualizzata la capacità media, in $\sigma_{C_{spad}}$ la sua standard deviation ed in $max C_{spad}$ e $min C_{spad}$ i valori massimi e minimi acquisiti durante tutta la sessione di acquisizione. Si possono resettare questi ultimi due valori premendo il pulsante (3). Quando si vuole fermare l'acquisizione basta premere il pulsante (2).

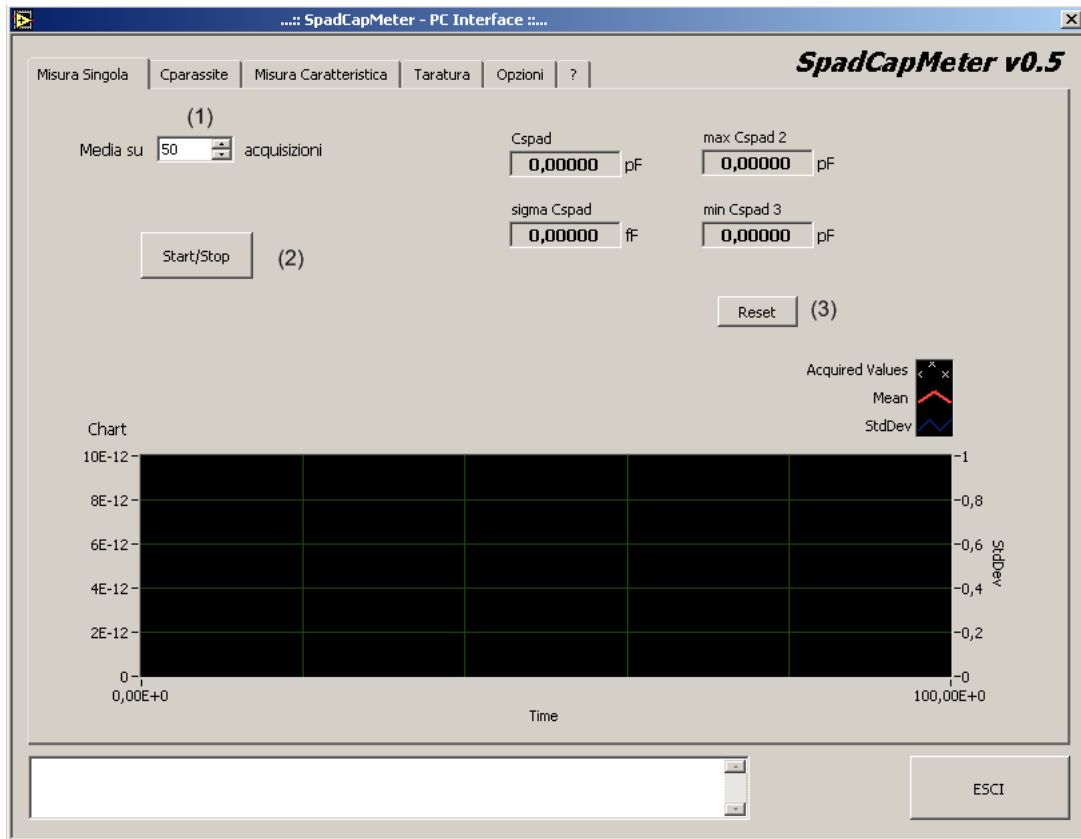


Figura 5.3: Schermata misura singola

5.2.4 Rilevazione caratteristica $C_{ak}-V_a$

A questo punto si passa al pannello *Misura Caratteristica* di cui uno screenshot è mostrato in Fig. 5.4. Per prima cosa si devono impostare i valori di tensione in corrispondenza dei quali effettuare le misure. Ciò si fa mediante il controller (1) dove si sceglie come devono venire spaziate i valori tra il valore minimo in (2) e il massimo in (3). In particolare si può scegliere una distribuzione lineare o logaritmica che suddivide automaticamente l'intervallo nel numero di passi desiderati, impostati dal controller (4). Se invece si sceglie la modalità user si devono inserire i valori di tensione che si desidera usare separati dal “;” nel controller (6). Attraverso il controller (5) si sceglie il numero di valori sui quali effettuare la media per trovare il valore preciso di capacità. Dopo questi settaggi si preme il pulsante (7) e l'utente deve seguire le operazioni fornite dal programma. Una volta terminata l'acquisizione di tutti i valori, essi vengono visualizzati sul grafico.

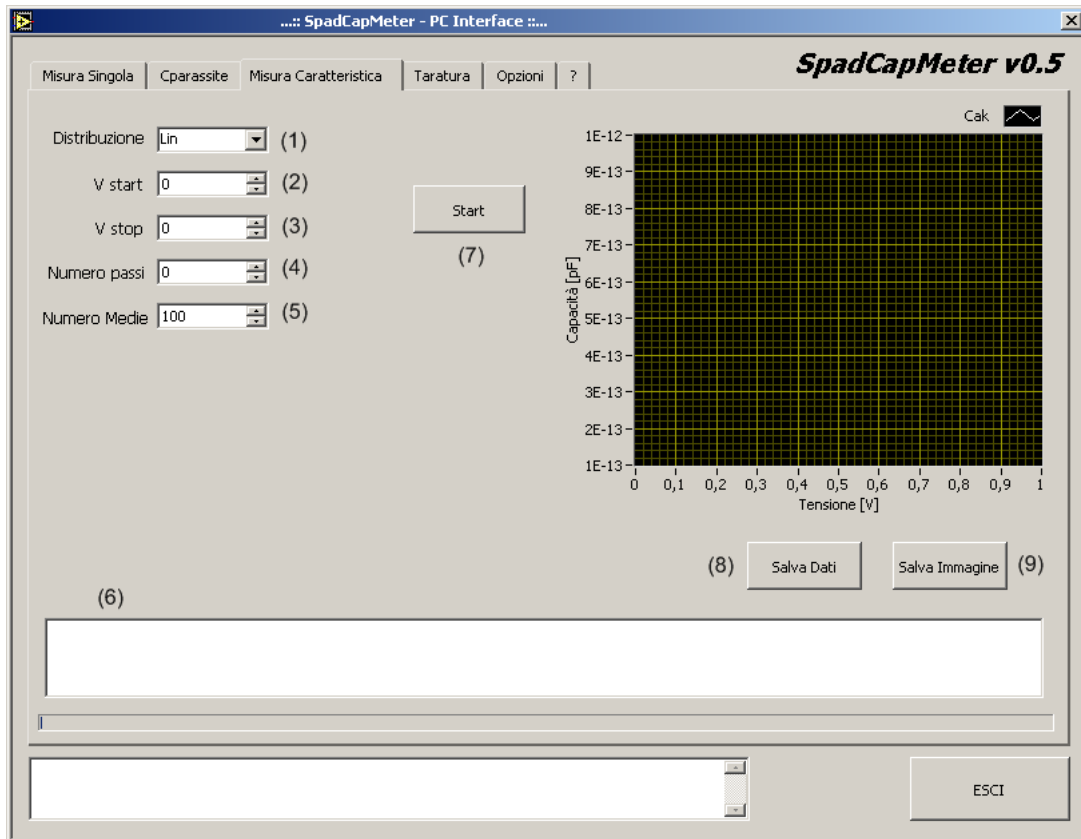


Figura 5.4: Schermata misura caratteristica

Infine tale grafico può essere esportato in formato immagine mediante il pulsante (8) oppure si possono salvare tutti i dati dei punti acquisiti premendo il tasto (9).

Capitolo 6

Misure sperimentali

Prima di concludere e per avere la certezza del corretto funzionamento del progetto, è stata svolta una misura della caratteristica $C_{ak}-V_a$ di uno SPAD al silicio.

In particolare si è utilizzato un fotodiodo di tipo thin, con tensione di breakdown di 17 V.

Di questo dispositivo la capacità interessata è stimata attorno a qualche pF. Per la taratura dello strumento sono stati utilizzati condensatori di precisione con valori di 0,7 pF, 1,1 pF e 4,3 pF. I risultati ottenuti in questa sezione sono mostrati in Fig. 6.2(a).

Una volta ottenuti i coefficienti della retta interpolante i risultati di questa sezione, si è preferito controllare, con una quarta capacità di valore intermedio a quelle utilizzate per la taratura, la bontà della misura. In particolare si è scelto un condensatore da 2.2 pF ed è stata impostata la modalità di misura singola, ottenendo la schermata riportata in Fig. 6.2(b). Si nota che il valore visualizzato non si discosta dal valore noto e quindi si è ritenuta la taratura adeguata per effettuare la misura della caratteristica.

Necessitando della relazione tra C_{ak} e V_a , bisogna prima calcolare il valore delle capacità parassite, utilizzando il pannello Cparassite. Sono state misurate le capacità C_{AK} , C_{BK} e C_{BA} , tutte con $V_a = 0$ e il programma ha calcolato il valore delle due capacità tra i contatti di anodo e catodo ed il substrato.

A tal punto, nel pannello delle opzioni, è stata abilitata la correzione delle

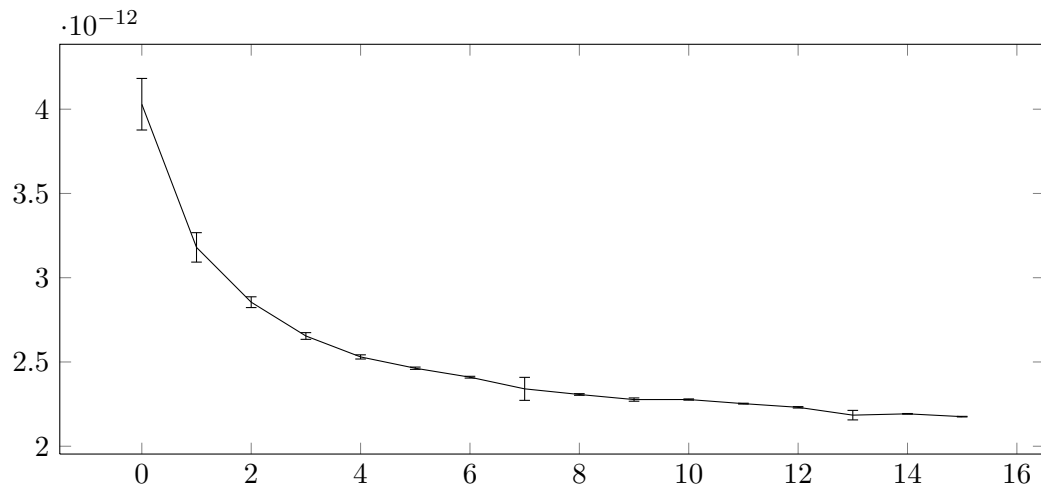
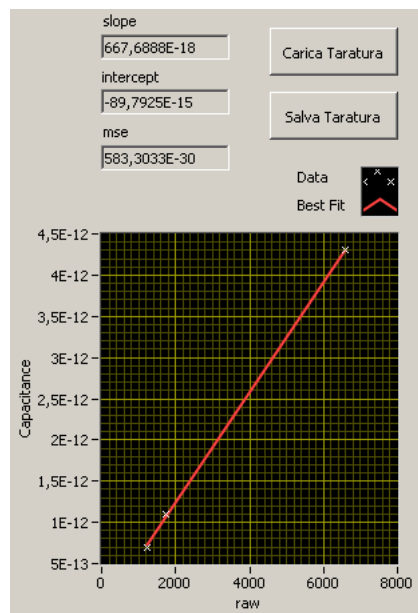


Figura 6.1: Grafico della caratteristica dello SPAD al Silicio

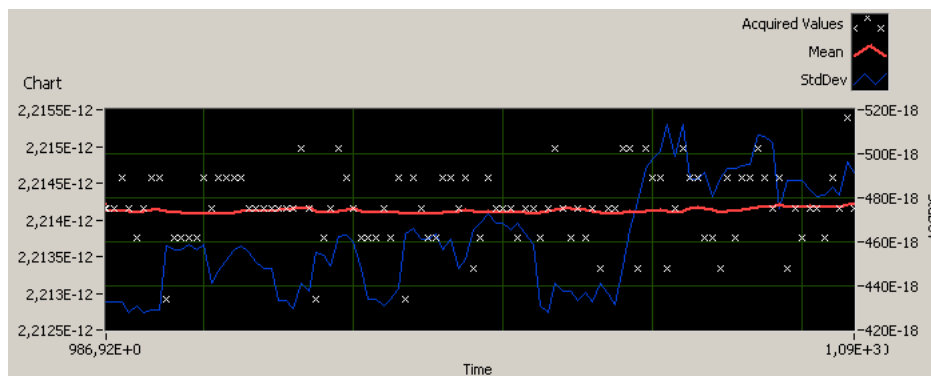
capacità parassite.

Per evitare di superare la tensione di breakdown dello SPAD, è stato impostato un set di valori di polarizzazione con distribuzione lineare, da 0 V a 15 V, suddivisa in 16 valori. La curva ottenuta è visualizzata in Fig. 6.1.

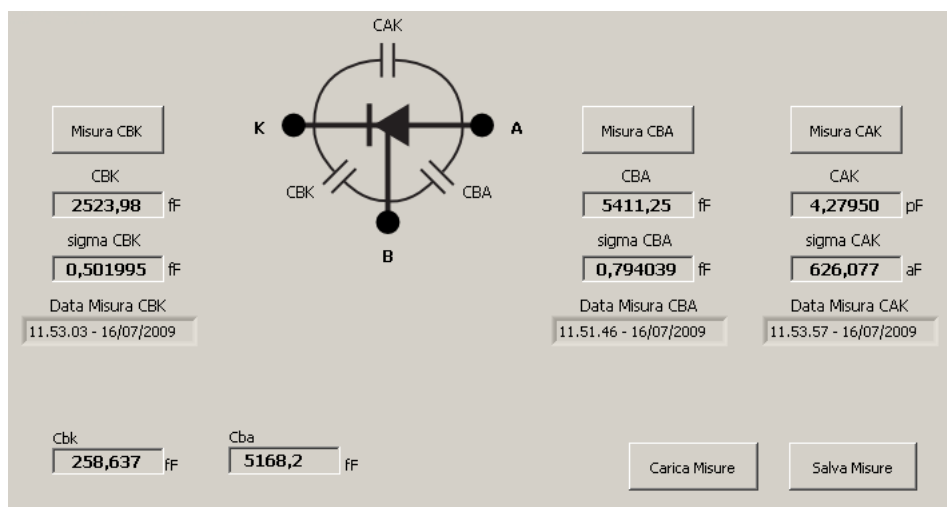
Durante questa ed altre misure effettuate, si è notato che tenendo lo SPAD al buio i risultati ottenuti sono stati più accurati. La curva caratteristica ottenuta mostra come la capacità anodo-catodo diminuisca all'aumentare della tensione di polarizzazione dello SPAD. Questo andamento è prevedibile poichè all'aumentare di V_a , aumenta l'ampiezza della regione svuotata e quindi la capacità di giunzione diminuisce.



(a) Schermata taratura



(b) Schermata misura singola



(c) Schermata capacità parassite

Figura 6.2: Schermate durante la misura

Capitolo 7

Conclusioni e prospettive future

7.1 Conclusioni

Nel lavoro svolto, è stato progettato e realizzato un sistema di precisione di facile utilizzo, in grado di misurare la capacità di giunzione di un fotodiodo SPAD.

Questa informazione è utile nel progetto del circuito di spegnimento passivo, in gated-mode con prelievo differenziale. In particolare utilizzando un segnale di abilitazione veloce (fronte di salita nell'ordine di 100-200 ps) la misura viene usata per il dimensionamento di C_{var} , ovvero la capacità inserita nel ramo speculare a quello dello SPAD in questa tipologia di circuiti. In questo modo vengono soppressi i disturbi all'ingresso del comparatore, introdotti dal segnale di gating. Ciò consente di minimizzare il jitter del segnale d'uscita.

Lo strumento è in grado di misurare capacità nel range di valori 0,1 pF - 20 pF, con una risoluzione di 2 fF. Esso inoltre consente di polarizzare il fotodiodo al fine di ricavare l'andamento della capacità al variare della tensione di bias. Nonostante sia espressamente progettato per la misura dei parassitismi dello SPAD, esso è anche in grado di misurare il valore di capacità normali, ovviamente restando all'interno del range di misura.

Grazie alla presenza di un microcontrollore sulla scheda sviluppata, la misura risulta essere completamente automatizzata, senza aver bisogno di un oscilloscopio. Inoltre, visto il collegamento con un PC, l'esecuzione della misura e la successiva elaborazione dei dati è resa molto più semplice. In particolare il collegamento USB garantisce un trasferimento veloce dei dati e un'elevata compatibilità, visto

che oggi giorno ogni PC è dotato di almeno una porta USB.

Mediante il software al calcolatore è possibile effettuare misure di caratterizzazione dello SPAD tenendo già conto delle capacità parassite, senza dover utilizzare ulteriori programmi di calcolo.

Infine il sistema è stato progettato per poter essere contenuto in un involucro dalle dimensioni ridotte, il che, combinato con l'uso del collegamento USB, lo rende un prodotto facilmente trasportabile. Questo ha influito anche positivamente sul costo del dispositivo che si aggira nel complesso a circa 50€.

7.2 Sviluppi futuri e prospettive

Per rendere completamente indipendente lo strumento bisogna sviluppare una adeguata sezione di alimentazione:

- $\pm 15\text{ V}$ per il funzionamento del “Capacimetro per diodi polarizzati”;
- con una tensione variabile con continuità da 0 V sino a circa 100 V per la polarizzazione dello SPAD.

Una possibile soluzione è presentata nel successivo paragrafo.

Altre prospettive di sviluppo per il futuro sono:

- la realizzazione di uno strumento unico formato dal “Capacimetro per diodi polarizzati”, dallo SPADCAPMETER e dall'alimentatore.
- determinare ogni quanto tempo si ha la necessità di ritardare lo strumento;
- lo sviluppo di driver di comunicazione USB per i sistemi operativi UNIX e MAC (il software è già compatibile con essi);

7.2.1 Alimentatore

Partendo dall'alimentazione singola fornita dall'USB, si può progettare un opportuno convertitore DC/DC che generi le tensioni di $\pm 15\text{ V}$ necessaria per il “Capacimetro per diodi polarizzati”. Prima di far ciò bisogna sempre controllare

che la corrente assorbita, oltre a quella usata dallo SPADCAPMETER , sia inferiore a 500 mA, ovvero la massima che l'USB può fornire. Stimato il consumo dello strumento, sicuramente non superiore ai 100 mA, restano ancora 400 mA. Questi devono essere ripartiti tra i +15 V e i -15 V e siccome si vuole partire da una tensione di 5 V la corrente massima erogabile è un terzo di quella assorbita (nel caso ipotetico di un convertitore ideale), cioè $200/3 \simeq 60$ mA. Questi sono abbondantemente sufficienti per alimentare l'unico componente attivo presente sul "Capacimetro per diodi polarizzati", l'LF356, il quale assorbe al massimo 10 mA.

Dopo un'attenta analisi dei dispositivi offerti dai principali produttori di convertitori switching, si propone l'integrato LT1945 della Linear Technology™ ; tale dispositivo presenta le seguenti caratteristiche:

- possibilità configurazione boost o SEPIC;
- due canali, uno per uscita positiva e uno per uscita negativa;
- switch interni ad alta tensione, può elevare tensioni fino a ± 34 V;
- corrente di switch internamente limitata a 350 mA;
- range della tensione di ingresso da 1.2 V a 15 V;
- bassissima corrente necessaria per il funzionamento (20 μ A senza carico, 0.5 μ A in standby);
- alta frequenza di switch per poter utilizzare piccoli induttori.

Un possibile utilizzo di tale integrato è mostrato in Fig. 7.1.

L'efficienza di conversione di questa configurazione è, da datasheet, di circa l'80% e questo viene confermato anche da una prova effettuata mediante il tool di simulazione LTspice, specifico per questo dispositivo.

Per quanto riguarda invece la generazione della tensione di polarizzazione bisogna prevedere un altro circuito boost che innalzi la tensione sino a circa 100 V. Questa tensione viene poi utilizzata per alimentare un amplificatore a transistori (o con amplificatori operazionali) che amplifica un segnale di tensione, per esempio 0-5 V proveniente da un DAC comandato dal microcontrollore; in uscita è così

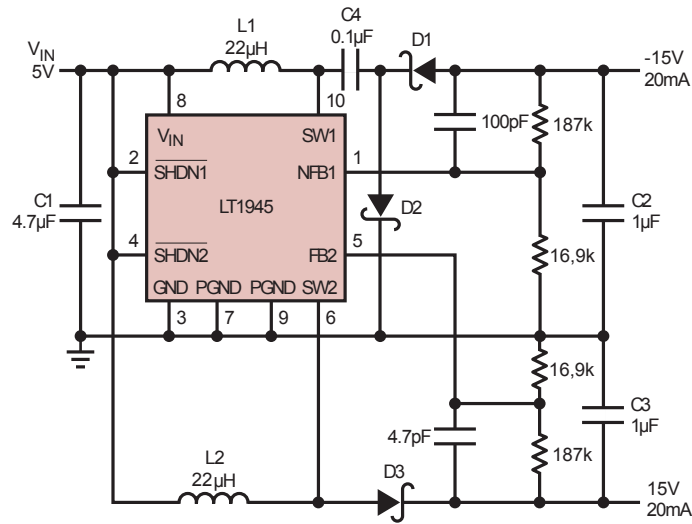


Figura 7.1: Circuito per convertitore Step-Up

possibile ottenere in una tensione variabile con continuità da 0 a 100 V. Bisogna ricordare che questa tensione serve per polarizzare lo SPAD, quindi l'assorbimento di corrente è veramente esiguo, visto che è dato da V_a/R_p dove R_p è nell'ordine delle centinaia di k Ω . L'assorbimento massimo è quindi di 1 mA quando la tensione è 100 V. Questi corrispondono a 20 mA assorbiti dall'USB se si considera un convertitore ideale.

In definitiva è quindi possibile costruire questo alimentatore, riuscendo ad assorbire una potenza inferiore alla massima che l'USB riesce a fornire.

Bibliografia

- [1] Sergio Cova, Massimo Ghioni, Andrea Lacaita, Carlo Samori, and Franco Zappa. Avalanche photodiodes and quenching circuits for single-photon detection. *Applied Optics*, vol.35(no.12):pp.1956–1976, 1996.
- [2] Alberto dalla Mora, Alberto Tosi, Simone Tisa, and Franco Zappa. Single-photon avalanche diode model for circuit simulations. *Photonics Technology Letters, IEEE*, vol.19(no.23):pp.1922–1924, 2007.
- [3] Perini Enzo. *Progetto di un Capacimetro per fotodiodi SPAD*. 2008.
- [4] S.O. Kasap. *Optoelectronics and Photonics: Principles and Practices*. Prentice-Hall Inc, 2001.
- [5] Microchip Technology Inc. Rawin Rojvanit. Migrating applications to usb from rs-232 uart with minimal impact on pc software. Technical report, 2004. <http://ww1.microchip.com/downloads/en/AppNotes/00956b.pdf>.
- [6] Cypress Semiconductor. *Technical Reference Manual*, May 2007. http://www.psocdeveloper.com/uploads/media/PSoC_TRM_00.pdf.
- [7] National Semiconductor. Easily tuned sine wave oscillators. Technical report, March 1971. <http://www.national.com>.
- [8] Microchip Technology. *PICmicro® 18C MCU Family Reference Manual*, 2000. <http://ww1.microchip.com/downloads/en/DeviceDoc/39500a.pdf>.
- [9] Microchip Technology. *MPLAB® C18 C Compiler User'S Guide*, 2005. http://ww1.microchip.com/downloads/en/DeviceDoc/C18_User_Guide_51288j.pdf.
- [10] USB.org. *Device Class Definition for Physical Interface Devices (PID)*, September 1999. http://www.usb.org/developers/devclass_docs/pid1_01.pdf.