

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA - CAMPUS DI CESENA

SCUOLA DI INGEGNERIA E ARCHITETTURA

Corso di Laurea Magistrale in INGEGNERIA ELETTRONICA E
TELECOMUNICAZIONI PER LO SVILUPPO SOSTENIBILE

Progettazione e collaudo di un ballast elettronico per lampade a scarica HID

Tesi in ELETTRONICA INDUSTRIALE

Candidato:
Francesco Bartoletti Stella

Relatore:
Prof. Rudi Paolo Paganelli

Correlatori:
Ing. Elvio Bigucci
Prof. Aldo Romani

Anno Accademico 2012/2013 - Sessione II

Indice

Introduzione	7
1 Informazioni Generali	9
1 Cenni Storici	9
2 Lampade a scarica	10
2.1 Lampade a scarica a bassa pressione	11
2.2 Lampade a scarica a elevata pressione	12
3 Ionizzazione dei gas	14
3.1 Regione di scarica oscura	15
3.2 Regione di scarica a bagliore	16
3.3 Regione di scarica ad arco	16
4 Innesco dell'arco elettrico	17
5 Cos'è un <i>ballast</i> ?	18
5.1 Ballast elettromagnetico	22
5.2 Ballast elettronico	23
6 Accensione della lampada	26
2 Progettazione del ballast elettronico	29
1 Stadio d'ingresso	30
2 Power Factor Correction	32
2.1 Circuito PFC-boost	35
3 Full-Bridge	36
3.1 Configurazione Full-Bridge	36
3.2 Rete di pilotaggio dei MOS	39
4 Filtro d'uscita	41
5 Circuito d'innescò	43

6	Sensing di corrente e tensione	46
6.1	Sensing di tensione	46
6.2	Sensing di corrente e reti di protezione	47
7	Flyback converter	51
8	Microcontrollore	52
9	Onde convogliate	53
3	Dimensionamento del ballast elettronico	55
1	Dimensionamento dello stadio d'ingresso	56
2	Dimensionamento dello stadio PFC	59
2.1	L'integrato per lo stadio PFC: L6564	64
2.2	Rete di polarizzazione di L6564	66
3	Dimensionamento Full-bridge	72
3.1	L'integrato di gate driver: FAN7382	73
4	Filtro d'uscita e circuito d'innesco	77
5	Dimensionamento reti di sensing	80
5.1	Reti di sensing di tensione	80
5.2	Reti di sensing di corrente	85
6	Dimensionamento flyback converter	88
6.1	L'integrato del flyback converter: TEA1721AT	89
4	Firmware	93
1	Il microcontrollore STM32F103CB	93
2	Il timer TIM1 per comandare i MOS	95
2.1	Inizializzazione di TIM1 nel file myBRIDGE.c	95
2.2	Funzioni di controllo del ponte	102
3	Gestione degli ingressi analogici	106
4	Ingressi e uscite general-purpose	108
5	Accensione della lampada	110
5.1	La funzione <code>voltage_check()</code>	111
5.2	La funzione <code>lamp_ignition()</code>	112
5.3	La funzione <code>lamp_warmup()</code>	116
5.4	Ricerca del duty ottimale	120
6	Prove di accensione della lampada	122

Indice	5
<hr/>	
Conclusioni	126
A Firmware: myBRIDGE.c	127
B Firmware: lamp.c	133
Bibliografia	141

Introduzione

Le lampade a scarica in gas ad alta pressione sono ad oggi la più consolidata e diffusa tecnologia utilizzata nell'illuminazione stradale, industriale e di spazi esterni in virtù del loro ottimo rapporto tra la qualità dell'emissione luminosa e il costo. Poiché la scelta della miscela di gas può privilegiare una particolare caratteristica (la potenza dell'emissione, la resa cromatica, l'efficienza energetica) sono lampade molto versatili e diffuse in numerose applicazioni.

Tradizionalmente queste lampade sono alimentate con dei reattori elettromagnetici che presentano però lo svantaggio di essere pesanti, voluminosi, al limite della vigente normativa sul Fattore di Potenza e di non consentire l'applicazione di strategie di risparmio energetico. Per queste ragioni l'utilizzo di un alimentatore elettronico costituisce un passo fondamentale per lo sviluppo delle moderne tecnologie di illuminazione, sia per le lampade a scarica che per la tecnologia a LED.

Nell'elaborato svolto viene presentato un ballast elettronico destinato ad applicazioni di illuminazione stradale. L'alimentatore deve sostituire il tradizionale reattore elettromagnetico attualmente montato sui lampioni, deve poter alimentare lampade fino a 150 W, deve soddisfare le più stringenti normative di futura approvazione, deve garantire ulteriori servizi aggiuntivi legati sia al migliore sfruttamento della lampada che a eventuali esigenze legate all'innovativo concetto di palo intelligente (alimentazioni ausiliarie e strategie di risparmio energetico).

La tesi è organizzata come segue:

Capitolo 1 - Vengono fornite informazioni generali sulle lampade a scarica ad elevata pressione, sulla ionizzazione del gas contenuto al loro interno e sull'importanza del circuito d'innescò dell'arco elettrico nel ballast.

Vengono inoltre illustrate le fondamentali funzioni svolte dal *ballast* (con particolare riferimento alla fase di accensione) mettendone in risalto i pregi e difetti delle due tecnologie esistenti sul mercato: il *ballast elettromagnetico* e il *ballast elettronico*.

Capitolo 2 - Sono analizzati tutti gli stadi che compongono il *ballast elettronico* progettato: esso è costituito da uno stadio d'ingresso per la correzione del fattore di potenza, da uno stadio di regolazione della corrente che svolge anche la funzione di inverter, da un circuito d'innescio, da un controllore elettronico che si occupa di armonizzare le funzioni dei vari stadi.

Capitolo 3 - Vengono riportati i calcoli svolti per il dimensionamento dei componenti utilizzati nei diversi stadi del sistema e le considerazioni svolte all'atto della scelta dei componenti di ogni singolo stadio e di tutti quei circuiti ausiliari necessari per la gestione di condizioni anomale (circuiti di protezione, filtri, diagnostica).

Capitolo 4 - All'interno di questo capitolo conclusivo sono raccolte le principali informazioni riguardanti la programmazione del microcontrollore, ovvero il cervello del reattore elettronico. La stesura del *firmware* (di cui una parte di esso viene riportato in appendice) ha determinato l'esecuzione di numerose prove e l'implementazione di diversi algoritmi per individuare quello che garantisce la gestione ottimale della lampada.

Capitolo 1

Informazioni Generali

*I will not say I failed 1000 times,
I will say I discovered there are 1000
ways that can cause failure*
- Thomas Edison (allegedly) -

1 Cenni Storici

Dal 1878, anno in cui Thomas Edison¹ inventò la lampada a incandescenza, ad oggi sono state condotte innumerevoli ricerche volte alla scoperta di nuove tipologie di lampade con l'obiettivo di aumentare il più possibile il rendimento luminoso e diminuirne i costi di realizzazione.

Ne seguì, nel 1903, la scoperta di William Coolidge² che utilizzò un filamento di tungsteno al posto del cotone carbonizzato precedentemente utilizzato da Edison. Una successiva variante fu messa a punto e portata sul mercato sotto il nome di *lampade alogene*, anch'esse appartenenti alla famiglia delle lampade ad incandescenza, ma che vedono il filamento immerso in un gas che ne aumenta l'efficienza luminosa (iodio, kripton e xeno) col quale il filamento stesso reagisce instaurando un continuo e ripetuto scambio chimico, chiamato appunto ciclo alogeno.

¹Thomas Alva Edison (11/02/1847 - 18/10/1931): inventore statunitense.

²William David Coolidge (23/10/1873 - 03/02/1975): fisico statunitense.

Di tutt'altra natura sono le *lampade a scarica di gas*. La loro scoperta, avvenuta a fine ottocento, è da attribuire a John Townsend³ e Daniel Moore⁴ che per primi condussero studi sulla conduzione elettrica dei gas e sulla ionizzazione di gas in tubo. La sostanziale e innovativa differenza sta nel fatto che in queste lampade l'emissione luminosa non avviene più per incandescenza da parte di un filamento conduttore, ma tramite la luminescenza di un gas ionizzato attraverso un'opportuna differenza di potenziale tale da generare un arco elettrico. Ecco perchè talvolta le lampade a scarica di gas vengono anche dette *lampade a luminescenza* oppure *lampade ad arco*. Un'ulteriore suddivisione di queste lampade sta nella pressione del gas luminescente: esistono infatti *lampade a scarica di gas a bassa pressione* dove viene utilizzato un gas nobile rarefatto (cioè a bassa pressione) con piccole quantità di mercurio e di materiali fluorescenti (un esempio sono le ben note lampade CFL, più comunemente chiamate lampade al neon), oppure *lampade a scarica di gas ad alta pressione* o *lampade HID* (*High Intensity Discharge*).

2 Lampade a scarica

In seguito alla scoperta della ionizzazione dei gas, fu Peter Cooper⁵ a proseguirne gli studi e a sviluppare nel 1901 la lampada a scarica a vapori di mercurio e successivamente, nel 1910, Georges Claude⁶ a scoprire la lampada al neon.

Una massa di gas rarefatto non è per sua natura un buon conduttore. Infatti è bene precisare che un gas allo stato naturale non si trova sotto forma atomica ma sotto forma di molecola (ad esempio H_2 : una molecola di idrogeno composta da due atomi) con bassissima attitudine conduttiva. Sotto l'azione del campo elettrico applicato agli elettrodi, la molecola in un primo tempo torna sotto forma atomica, spezzandosi nei due atomi di base, successivamente ogni atomo viene separato nelle sue componenti positive (che si sposteranno verso il catodo) e un flusso di elettroni (che si muoveranno verso l'anodo). In seguito a ciò il gas perde la sua natura dielettrica

³John Sealy Townsend (07/06/1868 – 16/02/1957): fisico e matematico irlandese.

⁴Daniel McFarlan Moore (27/02/1869 - 15/06/1936): inventore statunitense.

⁵Peter Cooper Hewitt (05/06/1861 – 25/08/1921): ingegnere elettronico statunitense.

⁶Georges Claude (24/09/1870 – 23/05/1960): ingegnere e inventore francese.

diventando elettricamente conduttivo ed assume lo stato di *plasma* (cioè di gas ionizzato) ad alta pressione e alta temperatura.

Questo movimento di particelle all'interno del tubo genera urti di vario genere:

- *urti a bassa velocità*: nell'urto tra le particelle viene ceduta energia cinetica con conseguente surriscaldamento del gas.
- *urti a alta velocità*: in seguito all'urto si ha l'eccitazione della particella, cioè uno o più elettroni dell'orbita esterna passano ad un livello energetico superiore restandoci per qualche nanosecondo, ricadendo poi nella configurazione stabile emettendo energia sotto forma di *fotoni*. La presenza nel gas di *vapori metallici* o *polveri metallici* che fa sì che la radiazione di fotoni avvenga nello spettro del visibile.
- *urti ad altissima velocità*: al momento dell'urto l'elettrone viene strappato dall'orbita appartenente generando un elettrone libero e uno ione positivo. L'elettrone urtando successivamente con atomi adiacenti mantiene attivo il processo.

Risulta chiaro come il rendimento e la caratteristica luminosa di una lampada siano contraddistinti principalmente da due parametri: il tipo di vapore metallico utilizzato e la pressione del gas ionizzante. Per agevolare questo procedimento viene aggiunto all'interno delle lampade un *gas d'innesco* (gas rari: xeno, neon, argon, elio, kripton) che presenta la caratteristica di facile ionizzazione.

2.1 Lampade a scarica a bassa pressione

Per questo tipo di lampade la pressione del gas è di qualche Pascal (pressione molto inferiore a quella atmosferica) e la temperatura dello stesso è relativamente bassa. La radiazione emessa è quasi interamente nel visibile e nell'ultravioletto, regalando a queste lampade un'elevata efficienza luminosa, ma il diagramma spettrale spiccatamente a righe presenta una piccola quantità di energia trasmessa al di fuori delle righe spettrali, con conseguente resa cromatica bassa. Inoltre il flusso luminoso emesso per unità di lunghezza è molto basso, per cui sono richiesti tubi di elevate dimensioni e ingombro.

Esempi di questa topologia di lampada è la *lampada a vapori di sodio a bassa pressione* oppure la più nota *lampada al neon*.

2.2 Lampade a scarica a elevata pressione

Per le lampade a scarica di gas ad elevata pressione o più semplicemente **lampade HID**, il principio di funzionamento è analogo a quello delle lampade a bassa pressione con l'unica differenza che in queste lampade il gas di ionizzazione è posto ad una pressione variabile tra i 10^4 e i 10^7 Pascal a seconda del gas utilizzato (1 atmosfera sono circa 10^5 Pascal) e che ne fa aumentare l'intensità e la frequenza con la quale avvengono gli urti tra atomi. In conseguenza a ciò, la temperatura raggiunta dal gas è più elevata. La scelta del gas utilizzato e la giusta miscela dei suoi costituenti (polveri di sodio o alogenuri metallici) determina lo *spettro di emissione* della lampada, della *temperatura colore* e della *resa cromatica*. Alcuni componenti, quali il mercurio, evaporano totalmente, mentre altri sono presenti in forma satura e rimangono in forma liquida, non evaporando. Si capisce dunque come la temperatura influenzi lo spettro di radiazione luminosa modulando la quota di ioni in grado di emettere in una precisa banda.

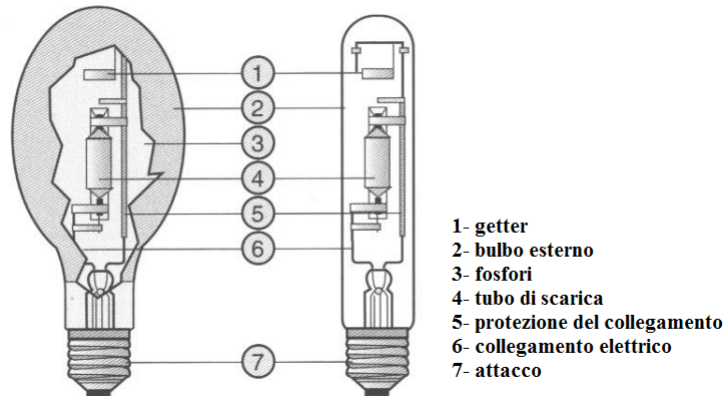


Figura 1.1: Struttura di una lampada a scarica di gas.[1]

Il gas viene racchiuso all'interno di un bulbo trasparente di quarzo o di alluminia policristallina (meno soggetta a contaminazione da parte della miscela interna), ermeticamente chiuso, in grado di resistere alle elevate temperature. All'estremità di esso sono presenti due elettrodi (i primi erano in

carbone, oggi sono in tungsteno) con il compito di generare l'arco elettrico che ionizza il gas. Un ulteriore tubo a vuoto esterno viene utilizzato nelle lampade HID per migliorarne l'isolamento termico e, attraverso l'impiego di particolari pellicole, per limitare i raggi UV prodotti.

Le lampade a scarica di gas ad alta pressione presentano maggiori irradiazioni nel campo dell'infrarosso (quindi al di fuori del visibile) che ne fa calare l'efficienza luminosa, ma allo stesso tempo il diagramma spettrale può essere quasi continuo ad alte pressioni garantendo una resa cromatica migliore. Il tallone di Achille di queste lampade risiede nella lunga e critica fase di accensione, suddivisa più correttamente nella fase di innesco dell'arco elettrico e nella stabilizzazione dello stesso.

	MERCURY VAPOR	HIGH PRESSUR SODIUM	CERAMIC METAL HALIDE
Lumens per watt	50-60	90-140	80-100
Temperatura Colore (CCT)	3300-5700	1900-2200	3000-4200
Colore lampada	Bluish-White	Gold	White
tempo di accensione (minuti)	5-7	3-4	2-5
tempo di vita (ore)	1200-2400+	1000-4000	750-2000

Tabella 1.1: Principali lampade HID a confronto.[2]

Le prime lampade HID ad andare sul mercato furono le lampade ai vapori di mercurio, seguite poi dalle lampade al sodio ad alta pressione e quelle agli alogenuri metallici che entrarono in commercio anche a causa della tossicità del mercurio. Resta comunque ancora viva la ricerca e lo studio di nuove tecnologie e di nuovi gas ionizzanti.

Principalmente queste tipologie di lampade sono utilizzate nelle zone dove vi è richiesta un'illuminazione a largo raggio e ad alta efficienza luminosa. Ne sono un esempio gli spazi pubblici come giardini, piazze, stadi, viali, ecc... Più recentemente, in seguito ad un continuo progresso nel rimpicciolimento delle dimensioni delle lampadine, vengono utilizzate anche in ambiente domestico (televisori e proiettori) e in ambiente botanico per riprodurre irradiazioni

solari nelle serre. Vengono inoltre utilizzate per i fari di automobili, motocicli e biciclette e, grazie alla loro elevata efficienza luminosa per unità di potenza, nella realizzazione di torce portatili o faretti per immersioni subacquee.

Come nota negativa va segnalato che la lampada non può essere immediatamente riaccesa a seguito di uno spegnimento, perché i sali metallici dai quali il gas è costituito devono avere il tempo di ri-cristalizzarsi. Un altro fenomeno negativo prende il nome di *cycling* e colpisce le lampade verso fine vita: sebbene la tensione di innesco dell'arco resta relativamente bassa anche per le lampade vecchie, una volta che il gas è ionizzato e la sua temperatura è elevata, anche la tensione di arco cresce e per le lampade vecchie può raggiungere livelli che il reattore non è più in grado di fornire provocando lo spegnimento dell'arco e di conseguenza della lampada. Una volta raffreddata la lampada può essere riaccesa, mettendo in essere un vero e proprio ciclo di accensioni e spegnimenti.



Figura 1.2: Esempi di utilizzo di lampade HID. A sinistra una lampada agli alogenuri metallici, a destra una lampada ai vapori di sodio.

3 Ionizzazione dei gas

Descrivere dettagliatamente cosa avviene all'interno del tubo di scarica durante la fase di ionizzazione del gas è cosa assai difficile per via di una enorme quantità di fattori e parametri in gioco quali: la natura del gas, il modo in cui la tensione viene applicata, dimensione e materiale utilizzato per gli elettrodi, natura delle superfici esposte, distanza fra gli elettrodi, forma e dimensione del tubo di scarica, ecc... Ne segue una notevole varietà di scariche nei gas che rendono difficile e mai definitivamente risolta la trattazione

elettrica della lampada. A grandi linee però è possibile distinguere tre fasi diverse di scarica al variare di tensione e corrente: *scarica oscura*, *bagliore* (o *glow*) e *arco*. Per pressioni vicine a quella atmosferica la fase di bagliore è estremamente veloce e di difficile identificazione e quindi la scarica assume subito la forma di arco.

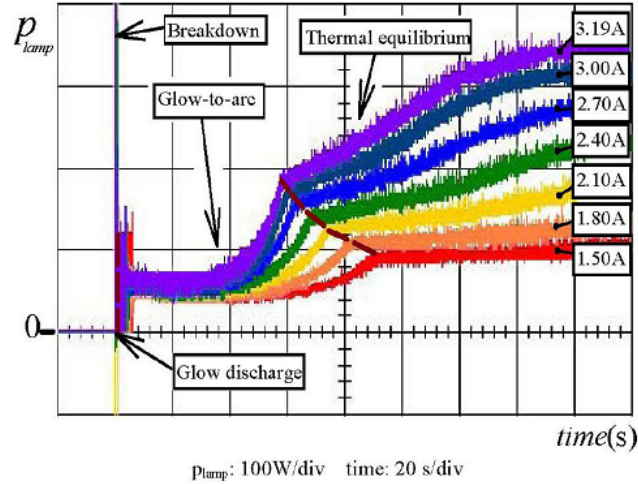


Figura 1.3: Andamento della potenza di lampada nel tempo subito dopo l'accensione e per diversi livelli di corrente.[3]

3.1 Regione di scarica oscura

In questa fase gli elettroni dispersi nel tubo (in parte elettroni liberi presenti nel gas e in parte elettroni emessi dal catodo) compresi tra i due elettrodi vengono accelerati ed acquisiscono energia tale per cui ad ogni collisione con gli atomi adiacenti viene emesso un ulteriore elettrone generando una *ionizzazione a valanga*. La corrente generata è esponenzialmente proporzionale a d , distanza tra gli elettrodi, e a α , che è un fattore inversamente proporzionale al cammino libero medio dell'elettrone tra due urti.

$$I \propto e^{\alpha d} \quad (1.1)$$

Tenendo conto che il cammino medio, esprimibile attraverso il rapporto tra energia dell'elettrone ϵ e campo elettrico applicato E , è inversamente proporzionale alla pressione del gas p se ne deduce che un parametro essenziale per caratterizzare la ionizzazione del gas è il rapporto tra campo

elettrico e pressione del gas E/p .

$$\alpha = Ape^{-\frac{B}{E/p}} \quad (1.2)$$

dove i parametri A e B dipendono dal gas utilizzato.

3.2 Regione di scarica a bagliore

Benché la tensione applicata agli elettrodi sia molto elevata e sufficiente a generare ionizzazione a valanga, il processo dipende fortemente dal flusso di elettroni derivanti dal catodo. Giunti in prossimità della *tensione di breakdown* si ha la vera rottura dielettrica del gas per la quale il flusso di elettroni non dipende più dal controllo esterno, ma viene determinato dall'emissione secondaria di elettroni nella vicinanza del catodo. La ionizzazione ora è autosostenuta ed è accompagnata dall'eccitazione di elettroni che causa un'emissione visibile con lunghezza d'onda che dipende dal gas utilizzato. Si identificano 3 zone di luce diffusa: al centro del tubo e in prossimità di anodo e catodo. Nella regione catodica l'elettrone viene estratto dal catodo, quindi accelerato, acquistando energia e dando origine alla ionizzazione a valanga. Nella regione in corrispondenza dell'anodo invece l'elettrone viene fortemente decelerato per poi essere riassorbito dall'anodo. Nella regione centrale il bagliore avviene poco prima che l'elettrone entri nella *colonna positiva* (zona di plasma dove numero di ioni e elettroni si eguaglia e il campo elettrico è nullo); la brusca perdita di energia genera un bagliore.

Alcune considerazioni per la regione di scarica a bagliore:

- * il valore tipico di corrente per una scarica a bagliore è delle decine di milliamperere;
- * la tensione di innesco (o tensione di breakdown) è funzione del prodotto tra pressione e distanza degli elettrodi;
- * il consumo di elettrodi cambia il valore della loro distanza, modificando quindi anche il valore della tensione d'innesco.

3.3 Regione di scarica ad arco

Nel passaggio da regione a bagliore a regione ad arco il contributo importante è dato dall'emissione termoionica del catodo, infatti se fino a quel

momento la ionizzazione avveniva per pura emissione catodica di elettroni, ora si aggiunge il forte riscaldamento del catodo a causa della collisione con gli ioni positivi del gas che ne fa aumentare l'emissione di elettroni per effetto termoionico. La tensione ai capi della lampada cala ulteriormente, mentre la corrente cresce molto velocemente, dalle decine alle migliaia di milliAmpe-re. Più precisamente è corretto dire che l'arco termico viene creato quando la temperatura elettronica uguaglia la temperatura ionica (viene perciò raggiunto l'equilibrio termico) che per una lampada a pressione atmosferica è di circa 5000K.

4 Innesco dell'arco elettrico

L'accensione delle lampade a scarica ad alta pressione è uno dei momenti più critici dal punto di vista elettrico e ne condiziona molto l'usura e l'invecchiamento.

I metodi per ottenere un gas ionizzato possono essere diversi, ma il metodo più semplice ed utilizzato è la già citata scarica a corrente continua nella quale le particelle sono accelerate mediante l'applicazione di un campo elettrico. La tensione di alimentazione delle lampade (230 V efficaci) risulta spesso volte insufficiente per generare l'arco e ionizzare il gas. Risulta perciò necessario trovare una soluzione a questo problema e favorire l'innesco dell'arco apportando modifiche alla struttura della lampada oppure utilizzando appositi circuiti esterni di innesco.

All'inizio della diffusione sul mercato delle lampade a scarica l'arco veniva generato tenendo inizialmente a contatto i due elettrodi e poi, attraverso un sistema meccanico, lentamente allontanati in modo tale che la corrente ionizzasse il gas e venisse generato e mantenuto l'arco. Altre soluzioni furono implementate successivamente: ne seguirono delle strutture nelle quali l'arco veniva prodotto utilizzando un elettrodo ausiliario. In alcuni casi l'elettrodo veniva messo sulla superficie esterna del tubo e mediante un impulso ad alta tensione ionizzava il gas, in altri casi invece l'elettrodo veniva posto nelle vicinanze di uno dei due elettrodi e, alimentato con la normale tensione di rete e data la breve distanza, innescava facilmente un piccolo arco; il riscaldamento degli ioni provocava l'innesco di tutto il gas restante.

La soluzione ad oggi utilizzata prevede l'utilizzo di appositi dispositivi, chiamati *starter* o *igniter*, che entrano in funzione solo al momento dell'accensione fornendo ad uno degli elettrodi una tensione di qualche KV. Lo starter è sostanzialmente un interruttore inizialmente aperto, racchiuso in una piccola ampolla contenente gas rarefatto che si ionizza una volta che è alimentato dalla rete. La ionizzazione del gas dello starter provoca il riscaldamento di una lamina bimetallica che si chiude e fa confluire alla lampada un flusso di elettroni elevato. Il gas dello starter non risulterà più ionizzato e il contatto si riaprirà provocando una variazione di corrente brusca che il reattore, per autoinduzione, commuterà in elevata tensione tale da superare la tensione d'innescò. Un altro dispositivo è l'igniter: è costituito da un *trasformatore ad impulsi* e un *dispositivo di scarica* (solitamente un DIAC), che per brevi istanti confluisce ad uno degli elettrodi una sovratensione tale da innescare l'arco. Per una trattazione dettagliata del circuito d'innescò si faccia riferimento al paragrafo 5 del capitolo 3.

5 Cos'è un *ballast*?

In tutti gli impianti di illuminazione che utilizzano lampade a scarica è previsto l'utilizzo di circuiti elettronici (o elettromagnetici) con il compito di regolare e gestire il corretto funzionamento della lampada. Questi componenti prendono il nome di *reattori* oppure *ballast* e sono sostanzialmente dei convertitori AC/AC operanti in range di tensioni precise con il compito di opporsi a rapide variazioni di correnti da parte del carico (la lampada in questo caso, oppure un driver per un motore che all'avvio richiede una corrente di spunto elevata), limitandone i picchi di corrente e attenuandone gli spike.

Il bipolo rappresentante la lampada in prima istanza può essere schematizzato come la serie di un generatore di tensione e una resistenza differenziale negativa⁷ poichè, una volta innescato l'arco, ad una corrente crescente reagisce con un abbassamento della tensione. Guardando infatti la dinamica

⁷La schematizzazione per mezzo di una resistenza è tanto più affine al comportamento reale quanto più alta è la frequenza di lavoro.

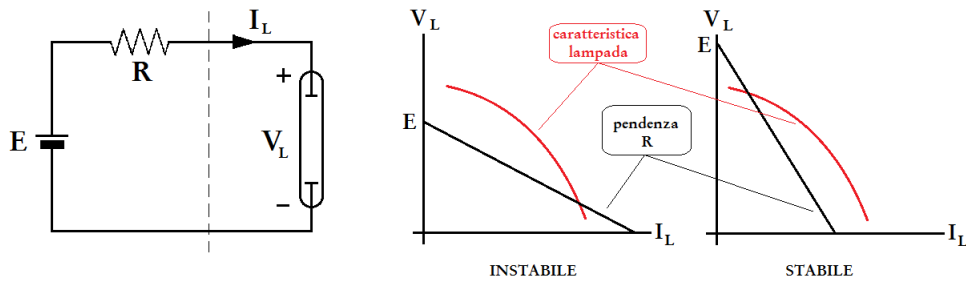


Figura 1.4: Bipolo a resistenza negativa rappresentante la lampada: comportamento stabile e instabile

della lampada (linea rossa di figura 1.4) si può notare come la risposta sia a resistenza differenziale negativa:

$$\frac{dV_{LAMP}}{dI_{LAMP}} < 0 \quad (1.3)$$

Ipotizzando di alimentare la lampada mediante tensione DC e facendo uso del bipolo equivalente di Thevenin, il risultato è quello mostrato in figura 1.4 dove si può vedere come il punto di intersezione tra la caratteristica della lampada e quella del bipolo possa essere *stabile* o *instabile*. Si prenda in considerazione ad esempio il caso instabile: un aumento di corrente richiesto dalla lampada porta la diminuzione della tensione di lampada al quale l'alimentatore risponde erogando una corrente maggiore di quella che ha perturbato l'equilibrio, portando così la lampada ad una deriva di correnti via via maggiori che ne causano la bruciatura. Analogo discorso si può dire in caso di diminuzione della corrente richiesta, dove in questo caso come risultato finale si ha lo spegnimento della lampada. Questo problema non si pone nel caso di comportamento *stabile* dove qualsiasi perturbazione di corrente richiesta dalla lampada viene controbilanciata dal bipolo alimentatore che risponde fornendo una quantità di corrente comunque inferiore a quella che ha causato la perturbazione dell'equilibrio, evitando l'instaurarsi di alcuna deriva. Affinché vi sia stabilità è necessario che R (che determina la pendenza della retta in figura 1.4) sia maggiore della *resistenza differenziale di lampada*, per ipotesi, negativa.

In prima analisi si può pensare di alimentare la lampada con un generatore di tensione DC direttamente collegato alla lampada. Questa soluzione risulta subito inadeguata per via del fatto che la resistenza equivalente di Thevenin è rappresentata dalla sola resistenza interna del generatore che è per sua natura molto piccola e quindi produce un comportamento instabile.

Vi sono due fondamentali ragioni per le quali l'alimentazione della lampada tramite una tensione DC e l'utilizzo di bipolo puramente resistivo con opportunamente elevato valore d'impedenza costituiscono un approccio sbagliato: nelle lampade ad arco esiste l'ulteriore necessità di invertire periodicamente la polarità per evitare un consumo asimmetrico degli elettrodi a causa dell'effetto indesiderato della *cataforesi*⁸. La seconda ragione riguarda l'efficienza del ballast: la resistenza è un componente che dissipa potenza perciò il bilancio energetico tra la potenza attinta dal generatore e la potenza effettivamente generata dalla lampada è poco equilibrato portando a valori di efficienza molto bassi; ad esempio mentre la lampada sta consumando 100 W di potenza luminosa, altrettanti Watt (se non di più) vengono dissipati dalla resistenza di polarizzazione per effetto Joule.

Il modo corretto di alimentare la lampada HID è dunque con un generatore equivalente di Norton, ovvero in corrente. Possibilmente, è poi bene invertire periodicamente il verso della corrente. Queste due considerazioni portano a dover rivedere l'approccio di alimentazione delle lampade a partire da sorgenti di alimentazione in tensione come quelle AC di linea. Si è passati perciò all'utilizzo di alimentazioni AC e di impedenze puramente reattive e non dissipative (minimamente). Utilizzando ad esempio un'induttanza di valore L la sua impedenza risulterà essere proporzionale alla frequenza di alimentazione ($\omega = 2\pi f$) e a L : $Z_L = j\omega L$. A meno di fenomeni resistivi parassiti (comunque presenti) che in prima istanza vengono trascurati, l'induttore fa crescere linearmente la corrente con pendenza proporzionale a L fino all'inversione o azzeramento della tensione ai suoi capi.

E' possibile dunque agire su ω e L per impostare il valore dell'impedenza presentata dall'induttanza. In origine, quando l'elettronica allo stato solido non era di vasto impiego, la soluzione comunemente utilizzata era il *ballast*

⁸La cataforesi consiste nella scomposizione delle molecole gassose in prossimità del catodo con relativa erosione dell'elettrodo.

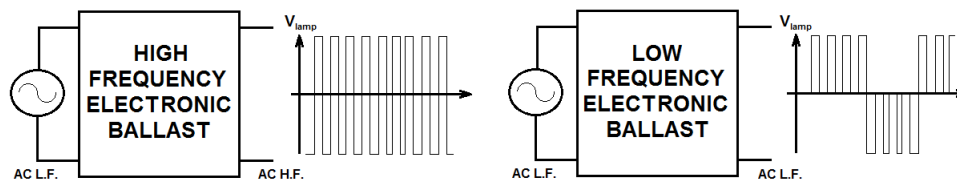


Figura 1.5: Ballast elettronici ad alta e bassa frequenza con relative tensioni di lampada

elettromagnetico: alimentato dalla tensione di rete (ad una frequenza perciò di 50-60 Hz), il ballast elettromagnetico è costituito da un induttore avvolto su un nucleo di materiale ferromagnetico (laminato per ridurre la dispersione di energia per riscaldamento da correnti parassite) posto in serie alla lampada. In alternativa possono essere utilizzati autotrasformatori o trasformatori a prese multiple.

Oggi, per aggirare il problema dell'ingombro portato da induttori di grandi dimensioni che determinano i richiesti elevati valori di L utilizzati nei ballast elettromagnetici, si preferisce passare all'utilizzo di *reattori elettronici* operanti a frequenze elevate consentendo l'uso di induttanze più piccole. Con riferimento alla frequenza di inversione dell'energizzazione della lampada i ballast elettronici si suddividono principalmente in due categorie: quelli operanti a bassa frequenza (tipici delle lampade a scarica ad alta pressione) dove la frequenza di lavoro va dai 50 ai 400 Hz e quelli operanti in alta frequenza (tipici delle lampade a bassa pressione e lampade fluorescenti) che lavorano con frequenze dell'ordine di decine di KHz.

Una domanda che ci si può porre è come modellare elettricamente la lampada. Se si esclude il comportamento della lampada negli istanti immediatamente successivi ai quali viene innescato l'arco, a regime la caratteristica della lampada è approssimabile tanto più a una resistenza quanto più alta è la frequenza di lavoro. Questo suggerirebbe di utilizzare una frequenza elevata⁹, se non fosse per l'insorgere del problema riguardante le perdite di commutazione oltre al problema della *risonanza acustica* (argomento trattato nel paragrafo 5.2).

⁹Più alta è la frequenza, minori saranno le dimensioni e gli ingombri dei componenti.

5.1 Ballast elettromagnetico

Nella sua forma più semplice il ballast elettromagnetico (*Electro-Magnetic Ballast, EMB*) è costituito da un semplice induttore posto in serie alla lampada. L'induttore introdotto limita la corrente che circola sulla lampada, ma allo stesso tempo introduce uno sfasamento tra tensione e corrente che porta ad avere un fattore di potenza (*Power Factor*) basso, con valori compresi tra 0.3 e 0.6. Questa defezione può essere efficacemente aggirabile mediante l'utilizzo di un condensatore di rifasamento che porta il fattore di potenza a un più appropriato 0.95 (oltre non è possibile andare per via della distorsione presente sulla corrente che, seppur di piccola entità, non è eliminabile del tutto).

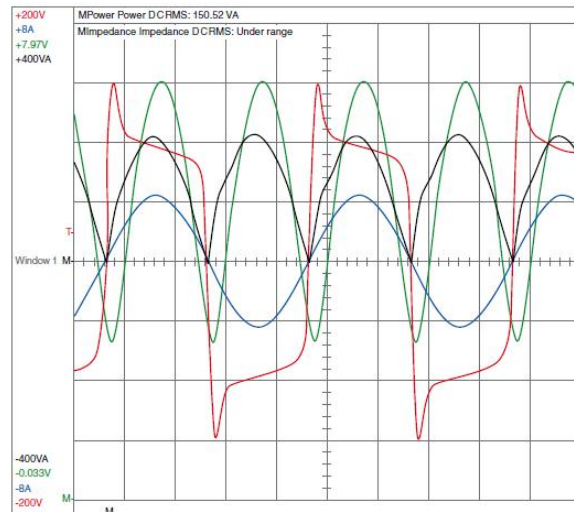


Figura 1.6: Andamenti di corrente (blu), tensione (rosso) potenza luminosa (verde) e potenza di lampada (nero) offerti da un ballast elettromagnetico.[4]

Alimentando la lampada con una corrente sinusoidale AC a 50 Hz si genera in prossimità di ogni attraversamento dello zero della corrente un picco di tensione: tale andamento è dovuto al fatto che la corrente diminuendo d'intensità porta un calo della potenza istantanea che tende così a raffreddare il plasma e gli elettrodi. Come conseguenza si ha l'aumento della resistività offerta dalla lampada che giustifica il picco di tensione di re-innesco (come visibile in figura 1.6). L'intensità del picco di re-innesco varia a seconda della temperatura alla quale la lampada lavora e dallo stato di usura degli elettrodi:

una temperatura di lavoro molto bassa rende difficoltoso l'instaurazione dell'arco e di conseguenza la tensione di picco sarà maggiore, in secondo luogo la cataforesi deteriora gli elettrodi perciò il cammino d'arco aumenta col tempo, aumentandone inesorabilmente la tensione d'innescò. Se il picco di re-innesco risulta maggiore della tensione di rete la lampada si spegne e non può essere riaccesa immediatamente¹⁰. Inoltre la potenza istantanea di lampada non ha un andamento costante, ma bensì segue un profilo di sinusoidale raddrizzata. Benché l'occhio umano non sia in grado di percepire questo comportamento a frequenza 100 Hz, tale funzionamento mette dei vincoli sul dimensionamento dei componenti che devono poter reggere potenze istantanee ben più grandi della potenza imposta da specifiche in fase di progettazione.

Se da una parte l'utilizzo di un ballast elettromagnetico è sinonimo di robustezza, semplicità di realizzazione ed economicità (il numero di componenti è limitato), dall'altra parte il suo utilizzo comporta ingombri elevati, stress della lampada notevole (il picco di re-innesco mette a dura prova la lampada) con conseguente durata di vita ridotta, efficienze non elevate e regolazione di potenza determinata a priori dal particolare circuito progettato e non gestibile a piacimento.

5.2 Ballast elettronico

Il funzionamento del ballast elettronico è radicalmente diverso da quello del ballast elettromagnetico. In questa tipologia di ballast è previsto l'utilizzo di dispositivi a MOS (o IGBT) che fungono da interruttori: la corrente che circola sulla lampada viene comunque calmierata mediante l'impiego di un induttore posto in serie ad essa, ma agendo sul tempo in cui il MOS risulta in conduzione è possibile regolare la corrente di lampada. Inoltre implementando una struttura *full-bridge* dei MOS è possibile risolvere abbastanza agevolmente il problema dell'inversione di energizzazione lenta presentato dai ballast elettromagnetici. Essendoci la possibilità di limitare la corrente agendo sul tempo di conduzione del MOS si può procedere utilizzando induttori assai

¹⁰Se all'interno del tubo il gas si trova ad una pressione maggiore della pressione a lampada fredda è richiesta una sovratensione maggiore per innescare l'arco. Ecco dunque che per riaccendere una lampada bisogna attendere qualche minuto durante il quale il gas si raffredda.

più piccoli di quelli utilizzati in precedenza e incrementando proporzionalmente la frequenza di commutazione del MOS, garantendo complessivamente impedenze¹¹ che non portano al *funzionamento instabile* del ballast. Sia la tensione presente ai capi della lampada sia la corrente che vi circola assumono andamenti ben differenti da quelli riportati in figura 1.6 per il ballast elettromagnetico. Dalla tensione di rete e utilizzando un convertitore *boost* è possibile ottenere la tensione DC con la quale alimentare il ponte (full-bridge o half-bridge). Successivamente un'opportuna configurazione di accensione dei MOS porta alla lampada la tensione DC per intervalli di tempo molto brevi nei quali la corrente che circola sull'induttore cresce linearmente, proporzionalmente a L e alla tensione ai suoi capi; terminata questa fase viene imposta la configurazione di ricircolo durante la quale la tensione impressa a L è nulla e la corrente sull'induttore assume pendenza negativa. Queste commutazioni avvengono a frequenze piuttosto elevate (decine di KHz) - HF - e permettono perciò di avere valori di induttanza ridotti. Ad una frequenza molto più bassa (centinaia di Hz) - LF - avviene l'inversione di polarizzazione della lampada sempre agendo sull'accensione di opportuni MOS; ciò è possibile solo utilizzando una configurazione *full-bridge* del ponte¹². L'inversione di polarizzazione è repentina e non genera i fastidiosi picchi di re-innesco che vincolavano i ballast EMB.

Numerosi sono i vantaggi riscontrabili dall'utilizzo di un ballast elettronico rispetto al precedente approccio elettromagnetico:

- utilizzando induttori più piccoli si ha una notevole riduzione del peso del dispositivo, oltre che ingombri ridotti
- induttanze più piccole rendono più veloce l'inversione della corrente al cambio polarità
- l'assenza del picco di re-innesco molto accentuato come nei ballast elettromagnetici porta meno stress alla lampada che perciò presenta una durata di vita più lunga (circa un 30% in più)

¹¹ $Z_L = \omega L$.

¹²La configurazione *half-bridge* senza capacità serie di disaccoppiamento della continua permette solo i livelli di tensione 0 e V_{DC} .

- intensità luminosa emessa dalla lampada è regolabile e stabile: la potenza di lampada, a meno degli istanti iniziali in cui la lampada deve entrare in equilibrio termico, è pressoché costante
- agendo direttamente sul tempo di conduzione dei MOS è possibile regolare la potenza fornita alla lampada e questa operazione risulta abbastanza semplice
- vi è la possibilità di avere un controllo attraverso microcontrollore sul tempo di conduzione dei MOS, ottenendo così informazioni sullo stato della lampada¹³
- il rendimento del ballast elettronico può essere molto elevato e il fattore di potenza è facilmente maggiore di 0.95, senza l'impiego di ingombranti condensatori di rifasamento e indipendentemente dalla fase di vita della lampada.

È bene precisare che sono comunque presenti gli spike di corrente all'inversione di polarità, ma sono di entità molto minore rispetto al ballast EMB, questo perché essendo l'inversione molto veloce non vi è il rischio che l'arco tenda a spegnersi (le particelle ionizzate del gas non fanno in tempo a raffreddarsi e ricombinarsi). Inoltre la corretta alimentazione della lampada avviene pilotando opportunamente tutti e 4 i MOS che non devono produrre configurazioni del ponte non ammissibili.

I primi ballast elettronici progettati regolavano l'inversione direttamente ad alta frequenza (dai 20 KHz in su) applicando perciò una corrente a valor medio nullo (primo schema di figura 1.5), ma ci si è accorti che tale configurazione mandava in *risonanza elettro-acustica* le lampade operanti ad alta pressione: alimentandola ad una frequenza così elevata si instaura all'interno del tubo un'onda stazionaria di pressione che porta alla vibrazione degli elettrodi e alla successiva loro deformazione. Inoltre lo stesso arco elettrico può essere perturbato e prendere direzioni diverse dal cammino minimo tra gli elettrodi, arrivando ad esempio a viaggiare sui bordi del tubo. La frequenza di risonanza meccanica è un parametro che non viene fornito dai costruttori di lampada in quanto difficile da determinare. Essa dipende dalla geometria del

¹³A parità di potenza che si vuole gestire, una lampada vecchia richiede tempi di polarizzazione più lunghi.

tubo, dalla composizione del gas, dalla temperatura di esercizio, dallo stato di invecchiamento della lampada, ecc...; vengono spesso fornite delle bande proibite alle quali non è consentito l'utilizzo del ballast¹⁴. Tutto ciò chiaramente comporta malfunzionamenti e invecchiamenti precoci della lampada che fanno cadere in parte i vantaggi presentati dai ballast elettronici.

La soluzione migliore che pone al riparo anche da questo problema di risonanza acustica prevede l'alimentazione ad alta frequenza della lampada e inversione di polarità a bassa frequenza (secondo schema di figura 1.5), come spiegato precedentemente.

6 Accensione della lampada

Analizzando il comportamento di tensione e corrente nella fase di accensione di lampade a scarica HID si capisce ancor meglio l'importanza dell'utilizzo dei ballast. Nell'esempio di accensione riportato in seguito si considera l'utilizzo di un ballast elettronico. A lampada fredda e con l'arco ancora non innescato il comportamento della lampada è approssimabile a quello di una resistenza di valore straordinariamente alto; non vi è infatti un percorso conduttivo che porta a circolare corrente e perciò la potenza prodotta è chiaramente nulla, sebbene la tensione ai suoi capi risulta quella ottenuta in uscita dal boost. Questa fase viene chiamata *innesco* della lampada.

Ad arco innescato inizia la fase di *warm-up* durante la quale corrente e tensione si portano più o meno lentamente ai valori di regime. Questa fase viene a sua volta suddivisa in ulteriori due fasi: *limitazione di corrente* e *regolazione di potenza*. Il passaggio dalla fase d'innescamento alla fase di limitazione di corrente è un momento assai critico per la lampada e per il gas al suo interno che si ritrova a passare dalla regione di scarica oscura, a quella di scarica a bagliore fino alla finale scarica ad arco come descritto nel paragrafo 3 del primo capitolo. La tensione ai suoi capi cala bruscamente a discapito di una corrente che invece crescerebbe illimitatamente se non si prendessero le opportune contromisure (limitazione della corrente). Durante questa fase

¹⁴Oltre alla difficoltà di conoscere a quale frequenza viene eccitato e attivato il modo meccanico di vibrazione bisogna porsi ad appropriata distanza anche dai modi multipli di quella fondamentale.

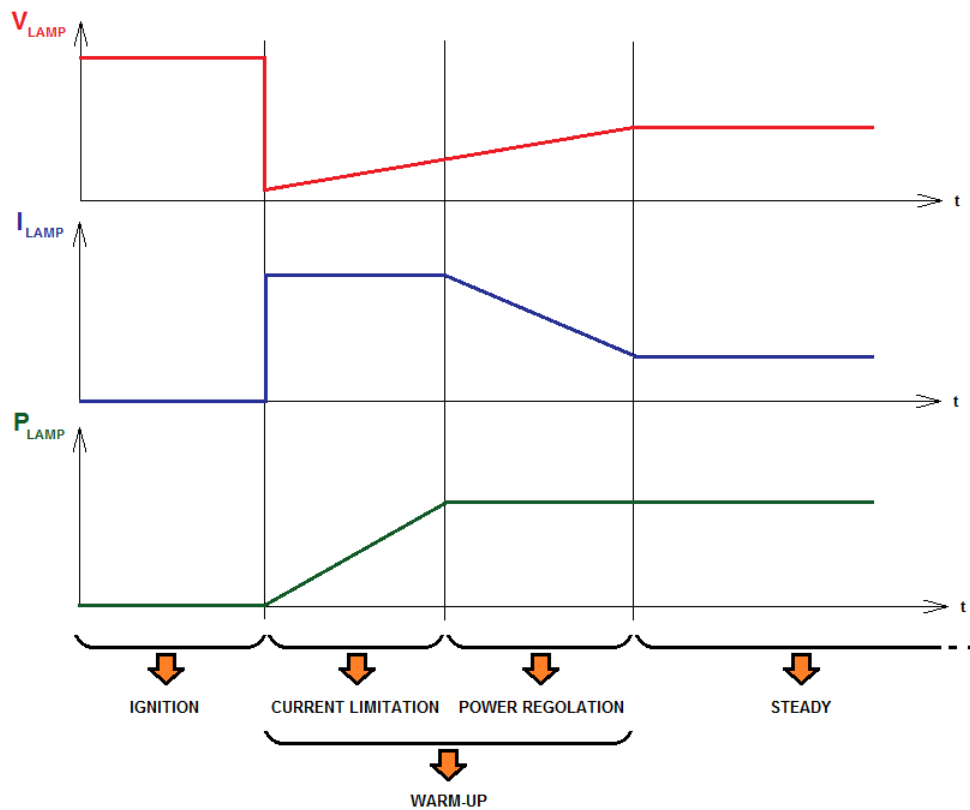


Figura 1.7: Schematizzazione degli andamenti di corrente (blu), tensione (rosso) e potenza (verde) di lampada all'accensione.

entra in funzione il ballast che si oppone a tale incremento e limita l'intensità della corrente. L'induttore fa crescere linearmente la corrente che però sale molto rapidamente essendo la tensione ai capi dell'induttore pressoché uguale alla tensione di alimentazione del ponte; sulla base di ciò è bene prevedere dei tempi di energizzazione molto brevi per non oltrepassare il limite di corrente che ci si è imposti. D'altro canto però avere tempi molto brevi può portare a spegnere l'arco appena innescato, perciò durante questa fase vi deve essere la *ricerca* del valore ottimale del duty-cycle di lampada¹⁵. Questa fase perdura per tutto il tempo in cui la lampada si scalda e sia la tensione sia la potenza crescono linearmente. Quando la potenza nominale di lampada raggiunge il

¹⁵L'alternativa proposta da *STMicroelectronics* prevede di agire sulla frequenza di commutazione HF di lampada lasciando inalterato il tempo di energizzazione della lampada, ma nel progetto realizzato si è preferito seguire la modifica *runtime* del duty-cycle e tenere la frequenza di commutazione fissa.

valore per la quale il ballast è stato progettato si passa alla fase di regolazione di potenza nella quale, come dice il nome, l'obiettivo è quello di mantenere costante la potenza di lampada. La tensione di lampada continua a salire per portarsi al valore nominale di lavoro perciò per mantenere costante la potenza si ha un abbassamento lineare della corrente, ottenuto anche in questo caso andando a regolare il valore del duty-cycle di lampada. La durata della fase di warm-up è determinata dallo stato di vita e dalle condizioni di lavoro della lampada. Tale valore è compreso tra i 30 secondi e un paio di minuti.

La fase di warm-up termina quando la tensione di lampada ha raggiunto il suo valore nominale di esercizio (tale valore dipende dallo stato d'usura e invecchiamento della lampada). La lampada rimane in questa fase, chiamata *steady*, durante tutto il tempo di esercizio fino a quando non sopraggiunge il comando di spegnimento della lampada (*volontario*) oppure fino a quando un malfunzionamento o un calo di tensione di rete non ne causa lo spegnimento (*involontario*).

Gli eventi che stabiliscono il transito da una fase all'altra sono:

- **transitorio da *innesco* a *limitazione di corrente***: in seguito all'innesco dell'arco la tensione di lampada scende bruscamente;
- **transitorio da *limitazione di corrente* a *regolazione di potenza***: viene raggiunto il valore di potenza nominale di progettazione;
- **transitorio da *regolazione di potenza* a *steady***: la tensione di lampada non cresce più e si assesta al valore di esercizio tale da produrre la potenza desiderata;

Capitolo 2

Progettazione del ballast elettronico

Nello studio e progettazione del ballast elettronico sono stati individuati ed esaminati separatamente tutti i blocchi che compongono il circuito per poterne fare un'analisi la più accurata possibile.

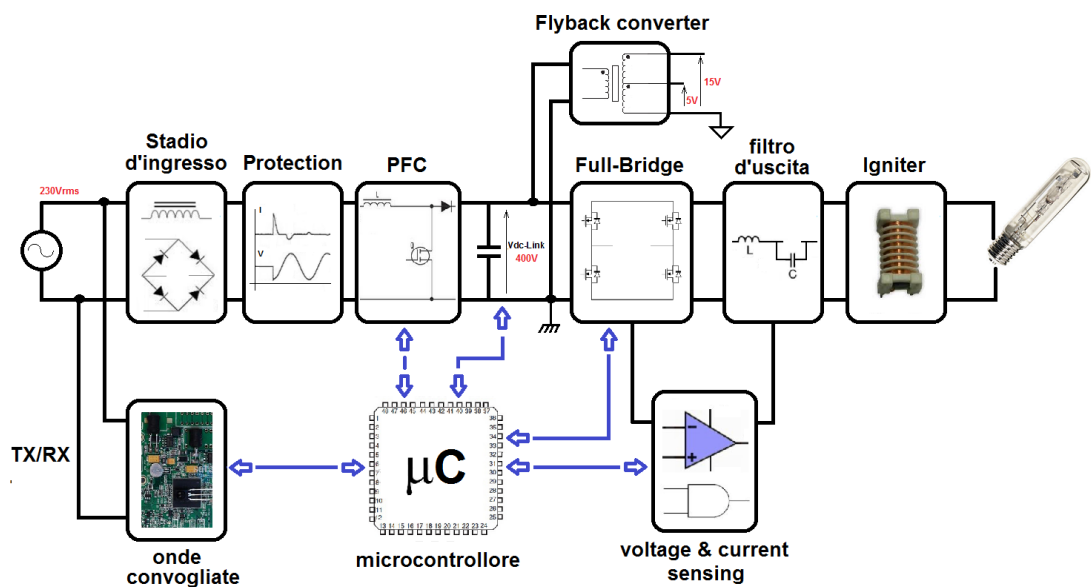


Figura 2.1: Schema generale del ballast elettronico progettato

1 Stadio d'ingresso

Lo stadio d'ingresso del ballast elettronico progettato presenta la serie del *filtro EMI* e del *ponte raddrizzatore a diodi*. A monte di queste due strutture è stato collocato un *fusibile* per la salvaguardia del dispositivo nel caso vi siano eccessi di corrente dovuti ad un errata configurazione operativa del ponte o in seguito a guasti di componenti di controllo e temporizzazione del ballast. La scelta di tale fusibile deve garantire l'interruzione del passaggio di corrente nel caso di guasto del dispositivo, ma al tempo stesso deve essere "insensibile" alla *in-rush current* presente al momento dell'accensione. Alla luce di ciò la scelta del fusibile è ricaduta su una topologia ad intervento molto lento detta *Slo-Blo*¹.

Successivamente al fusibile, nel rispetto delle normative *CEI* sulla compatibilità elettromagnetica dei dispositivi, vi è il filtro EMI con il compito di limitare il più possibile i disturbi prodotti dal dispositivo e quelli in ingresso allo stesso provenienti dalla rete di distribuzione. Tale filtro, nella più nota configurazione a π con un induttore e due capacità, deve risultare avere guadagno unitario nell'intorno della frequenza di rete (50-60 Hz), mentre deve poter tagliare le frequenze comprese tra i 150 KHz e 30 MHz, da normativa. Componente di rilievo del filtro EMI è il magnetico *common mode choke* costituito da un induttore a doppio avvolgimento di ugual numero di spire, collegati uno in serie alla fase e l'altro in serie al ritorno, il neutro. Questo magnetico è così progettato affinché una corrente di modo differenziale non genera flusso magnetico all'interno del nucleo (risultando una impedenza molto piccola, a meno delle resistenze parassite), mentre una corrente di modo comune genera flusso all'interno del nucleo, portando l'induttanza ad un valore di impedenza specificata per il componente e quindi attenuandone o meglio filtrandone il contributo. Per limitare la dispersione del flusso all'interno del nucleo ad alta permeabilità magnetica vengono scelti dei nuclei toroidali. Il materiale del nucleo è solitamente in polvere di ferro con l'aggiunta di qualche altro elemento (nichel, zinco, manganese) che ne determina la frequenza di utilizzo. I due condensatori posti a monte e a valle del common

¹Sono stati perciò scartati topologie di fusibile veloci - *fast-acting* - e molto veloci - *very-fast-acting*.

mode choke sono due condensatori di linea² e sono di classe X , mentre quelli di classe Y sono detti condensatori di neutro-terra. A loro è demandato il compito di filtrare le interferenze EMI di modo differenziale. Il principale requisito richiesto da questi condensatori è di poter resistere a sovratensioni. Vengono pertanto suddivisi in classe $X1$ (la più performante), $X2$ e $X3$ (la meno performante).

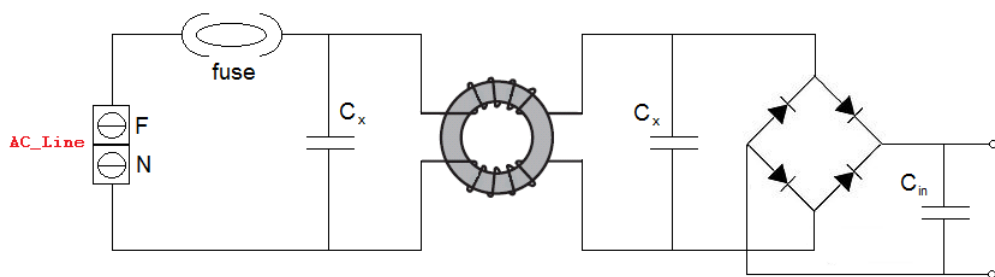


Figura 2.2: Stadio d'ingresso costituito da fusibile, EMI filter e ponte raddrizzatore

Infine vi è la necessità di raddrizzare la tensione di rete per poterla poi darla in gestione al power factor controller (paragrafo 2). A questo pensa il ponte raddrizzatore che deve esser scelto per poter operare con la sinusoide a 230 V efficaci provenienti dalla linea, ma deve poter resistere anche a accidentali e brevi sovratensioni che la linea porta con se. Anche la *nominal melting energy*³ che può sopportare il ponte è un parametro importante da tenere in considerazione, sebbene una corretta progettazione del ballast elettronico fa sì che la nominal melting energy del ponte sia superiore a quella del fusibile di modo che il primo a rompersi in caso di sovracorrenti è il fusibile. A valle del circuito raddrizzatore vi è la capacità che funge da capacità d'ingresso per lo stadio a di PFC boost.

²Detti anche *condensatori shunt*.

³Chiamato anche I^2t rappresenta l'energia che un dispositivo può sopportare prima che avvenga la sua rottura e dipende chiaramente dal quadrato della corrente in gioco e dal tempo.

2 Power Factor Correction

In seguito allo stadio d'ingresso e quello di protezione dalle correnti di in-rush è presente lo stadio di *Power Factor Correction*. Lo scopo del PFC è quello di assorbire dalla rete una corrente pressoché sinusoidale per avere così un *Power Factor* ⁴ che tende il più possibile all'unità. Per far ciò, il PFC ha lo scopo di rifasare il circuito di carico per farlo apparire quanto più possibile resistivo in modo tale che la potenza apparente sia uguale alla potenza reale. Introducendo dunque un circuito in grado di operare in questa maniera, si rende più efficiente la fornitura di energia da parte della rete e si soddisfa la normativa vigente per questa tipologia di apparati.

Come PFC possono essere utilizzate configurazioni in *buck converter*, *boost converter* oppure *buck-boost converter*.⁵ Sono essenzialmente due i motivi per il quale è stata scelta la topologia *boost* per lo stadio PFC: la prima è che questa configurazione prevede sulla linea d'ingresso la presenza di un'induttanza che limita la distorsione in corrente assorbita dal ponte raddrizzatore e riduce le interferenze EMI; la seconda è che nel mercato c'è larga diffusione di dispositivi integrati di controllo dell'interruttore presente in questa topologia. Tale interruttore altro non è che un MOS con il source connesso direttamente a massa, perciò risulta abbastanza facile e immediato il suo pilotaggio, oltre al fatto che la rete di condizionamento dell'integrato è esigua di componenti (con vantaggi dunque in termini economici e di ingombro). Come svantaggio si ha che in uscita dal boost, sul condensatore d'uscita DC-Link, vi è una tensione più alta della tensione di picco di rete e più alta della tensione strettamente necessaria per alimentare la lampada.

A seconda di come viene fatta scorrere la corrente sull'induttanza di boost, esistono essenzialmente due tipologie di controllo dello stadio PFC: *Fixed Frequency Continuous Conduction Mode* (FF-CCM) e *Boundary Conduction Mode* (BCM), altrimenti detto *Transition Mode* (TM). In letteratura sono

⁴Con il termine Power Factor si intende il rapporto tra la potenza reale P (misurata in Watt) e la potenza apparente S (misurata in Volt-Ampere).

⁵La topologia boost è principalmente utilizzata negli alimentatori switching dove in uscita dall'alimentatore abbiamo una tensione DC che deve essere mantenuta la più costante possibile, mentre la corrente assorbita dalla rete deve essere in fase e alla stessa frequenza della tensione di rete.

state poi proposte soluzioni che miscelevano la capacità di gestire grandi correnti della prima topologia con l'elevata semplicità e convenienza della seconda dando origine a stadi boost PFC *Peak-Current Mode with Fixed Off-Time* (FOT).

Nel modo di operare CCM la corrente che scorre sull'induttore viene controllata in un intorno della forma di corrente desiderata (sinusoide raddrizzata) e il ripple massimo di corrente è ben più piccolo della corrente desiderata. Per far ciò vengono utilizzati dei circuiti integrati (quali ad esempio L4981 A/B della *STMicroelectronics*) che implementano un comparatore con isteresi sul valore di corrente desiderato generando un ripple a frequenza variabile, oppure comparatori a frequenza fissa dove si va a regolare il tempo di accensione (quello di spegnimento è determinato di conseguenza) sulla base della comparazione tra il segnale di corrente retroazionato e un riferimento massimo di corrente. Come è possibile vedere in figura 2.3 non si avrà mai l'annullamento della corrente sull'induttore (a parte brevi istanti dove il riferimento di corrente tende a zero) ed inoltre il ripple di corrente sull'induttore è limitato, ottenendo così picchi di corrente contenuti. Riconosciute queste caratteristiche è immediato pensare che tali topologie di boost sono maggiormente utilizzate nei circuiti di potenza dove le correnti assorbite sono elevate. Come contro vi è un controllo in retroazione non banale che richiede un numero consistente di componenti.

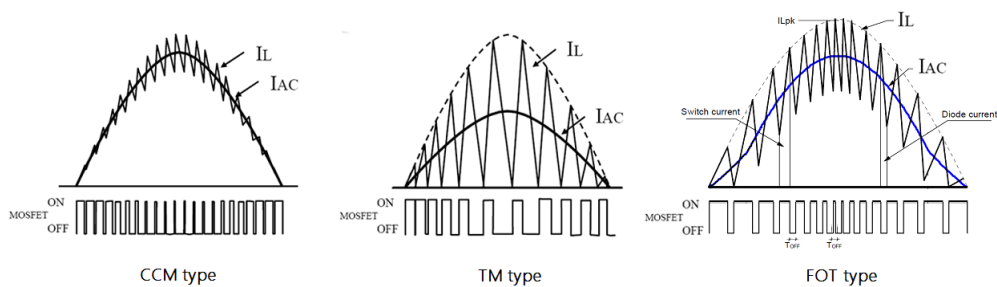


Figura 2.3: Topologie di Power Factor Correction. [5]

Nella topologia TM invece la corrente che circola sull'induttore è al limite della discontinuità: viene infatti fatta crescere in un intervallo di valore costante T_{on} e il picco di corrente raggiunto è proporzionale alla tensione di

rete in T_{ON} , mentre la durata della fase T_{OFF} è stabilita dall'intervallo di tempo che intercorre dall'esaurimento della fase T_{ON} all'istante in cui la corrente è scesa fino ad arrivare ad annullarsi, facendo successivamente ripartire l'intervallo T_{ON} . Risulta chiaro (e l'immagine in figura 2.3 lo rende visibile) che la frequenza di commutazione dell'interruttore MOS è variabile, raggiungendo massimi quando la corrente è prossima allo zero e minimi quando la corrente è in prossimità di massimi. Come vantaggio si ha un utilizzo minimale di dispositivi data la rete di condizionamento abbastanza semplice, mentre come svantaggio si ha un ripple di valore doppio rispetto alla corrente media desiderata. Ciò comporta un dimensionamento dei dispositivi (induttore e diodo sopra tutti) che deve essere il doppio di quelli che si sarebbero utilizzati nel caso CCM. Maggior attenzione va rivolta all'induttore che deve poter supportare tali correnti senza mandare in saturazione il magnetico del nucleo. Vi è dunque una limitazione nelle potenze sopportate da questo tipo di PFC.

Sul mercato sono diversi i circuiti integrati in grado di operare in modalità transition quali ad esempio L6561, L6562, L6562A, L6564 (che è stato utilizzato in questo progetto di ballast elettronico), ecc...

Come detto precedentemente vi è una terza configurazione, detta FOT, che cerca di prendere i benefici di ciascuna delle due topologie sopra citate: il principio di funzionamento è discontinuo per bassi valori di corrente, per poi diventare continuo per alte correnti in ingresso. Alla base di tutto vi è la prerogativa di fissare un tempo T_{OFF} e mantenerlo di durata costante ottenendo come risultato finale un giusto compromesso tra la semplicità di realizzazione della rete di controllo e la non perfetta idealità della forma d'onda di corrente assorbita dalla rete (che ne degrada il Power Factor).

2.1 Circuito PFC-boost

In uscita dal ponte di diodi vi è una tensione sinusoidale raddrizzata, utilizzata come ingresso del convertitore boost che, attraverso un opportuno comando di pilotaggio del MOS, genera in uscita una tensione regolata DC.

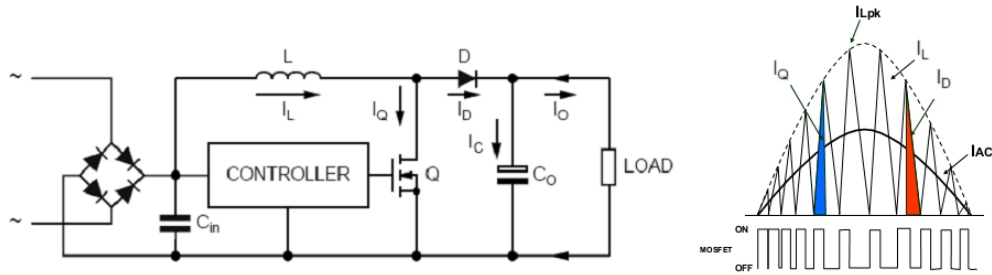


Figura 2.4: Circuito di PFC boost e andamento della corrente.[6]

Volendo regolare l'uscita ad un valore prefissato se ne retroazionerà una frazione attraverso una rete resistiva: confrontando tale valore con un riferimento, si produrrà un segnale *errore* la cui dinamica è da considerarsi assai più lenta della stessa frequenza di rete. Il segnale di errore prodotto (quasi DC) viene moltiplicato per una frazione della tensione raddrizzata presa in ingresso: come risultato si è ottenuta una tensione che rappresenta il limite di corrente massima che assume l'andamento di una sinusoidale raddrizzata, sincrona con la rete e la cui ampiezza è lentamente variabile e proporzionale all'errore della tensione in uscita sul condensatore DC-link rispetto al valore atteso.

A MOS acceso, la corrente che circola sull'induttore cresce linearmente fino a quando, grazie ad un sensing retroazionato di corrente, non raggiunge il valore in uscita dal moltiplicatore che causa lo spegnimento del MOS (circuito aperto). Ora la corrente è costretta a circolare sul diodo D che entrando in conduzione porta sul morsetto dell'induttanza la tensione V_{OUT} (a meno della soglia del diodo). Tale tensione (negativa per il riferimento scelto della corrente I_L) produce una rampa negativa della corrente, fino a causare l'annullamento e lo spegnimento del diodo D .⁶ Ne segue quindi una nuova fase

⁶La tensione sul drain del MOS diventa flottante.

di energizzazione dell'induttanza accendendo il MOS e via via, ciclicamente, si ripete l'accensione e lo spegnimento dell'interruttore.

3 Full-Bridge

Per la funzione integrata di regolazione di corrente ad alta frequenza viene utilizzato una configurazione a MOS ad *half-bridge* oppure *full-bridge* opportunamente comandati. La realizzazione del ponte in configurazione full-bridge (detto anche *ponte ad H*) consente una maggiore flessibilità e regolazione del duty-cycle oltre che un raddoppio della frequenza di regolazione della corrente di lampada, a tutto vantaggio delle prestazioni.

Una *resistenza di sensing* posta tra il source dei MOS inferiori e massa permette di misurare il valore della corrente che circola sul ponte durante la fase energizzazione della lampada, quando cioè vi è un percorso conduttivo che permette di prelevare corrente dal DC-link, farla circolare sul carico (filtro d'uscita, igniter e lampada) per richiudersi infine a massa, attraversando la resistenza di sensing; durante la fase di ricircolo invece la resistenza non è percorsa da corrente perciò una misura di corrente effettuata durante questa fase restituisce un valore nullo. Inoltre il valore di corrente misurabile risulta sempre positiva indipendentemente dal tipo di energizzazione (positiva o negativa).

3.1 Configurazione Full-Bridge

A seconda dei MOS contemporaneamente attivi si identificano tre possibili stati del ponte: *energizzazione positiva*, *energizzazione negativa*, *ricircolo*. Lo stato di energizzazione lo si ottiene attivando il MOS superiore di un leg e quello inferiore dell'altro leg; la discriminazione sul fatto che possa essere positiva o negativa è stabilita dal verso di riferimento che viene assunto per la tensione d'uscita del ponte V_{OUT} . Facendo riferimento alla figura 2.5 attivando M1 e M4 (e tenendo spenti M2 e M3) si ottiene energizzazione positiva, attivando M2 e M3 (e tenendo spenti M1 e M4) si ottiene energizzazione negativa. Lo stato di ricircolo, identificabile da una V_{OUT} nulla, si può ottenere in due diverse configurazioni, attivando cioè M1 e M3 (tenendo spenti

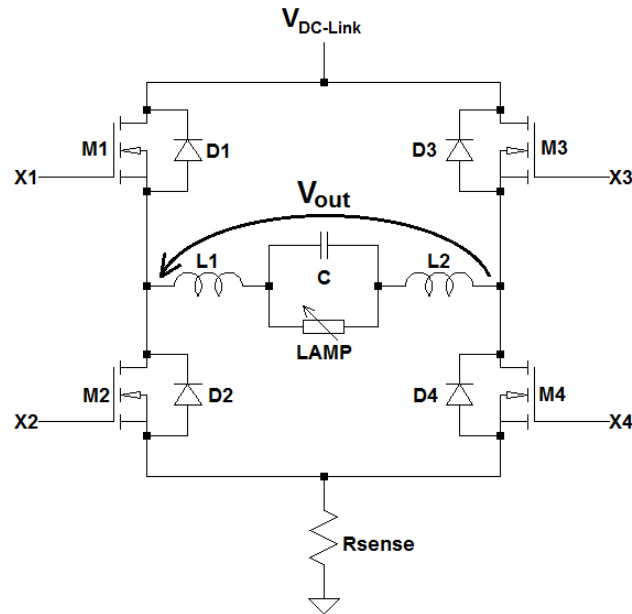


Figura 2.5: Realizzazione dello stadio full-bridge tramite i MOS $M1$, $M2$, $M3$ e $M4$ comandati rispettivamente con i segnali $X1$, $X2$, $X3$ e $X4$

$M2$ e $M4$) e viceversa. Lo stato in cui $M1$ e $M2$ (oppure $M3$ e $M4$) sono contemporaneamente attivi è una configurazione che è assolutamente da evitare in quanto si creerebbe un cortocircuito tra alimentazione e massa che porterebbe una sovracorrente in grado di produrre danni irreparabili. E' per questa ragione che il MOS $M2$ viene comandato con il segnale complementare di $X1$ e, analogamente, $M4$ viene comandato dal segnale complementare di $X3$. Inoltre è importante prevedere (attraverso firmware oppure utilizzando appositi integrati) che vi siano *blanking-time*⁷ quando vi è la commutazione dei segnali sul leg.

Osservando la figura 2.6 si può notare come il segnale $X4$ risulti uguale a $X1$ ma sfasato di $T^{X1}/2$ e ugualmente per i segnali $X2$ e $X3$.

Occorre non fare confusione tra due diversi duty-cycle che d'ora in avanti verranno presi in esame: con la lettera D verrà indicato il duty-cycle che regola l'accensione e lo spegnimento dei MOS sui leg del ponte, mentre con

⁷Detto anche *dead-time*, viene indicato una frazione di tempo posto ad ogni commutazione nel quale i MOS dello stesso leg vengono tenuti entrambi spenti per evitare cortocircuiti indesiderati.

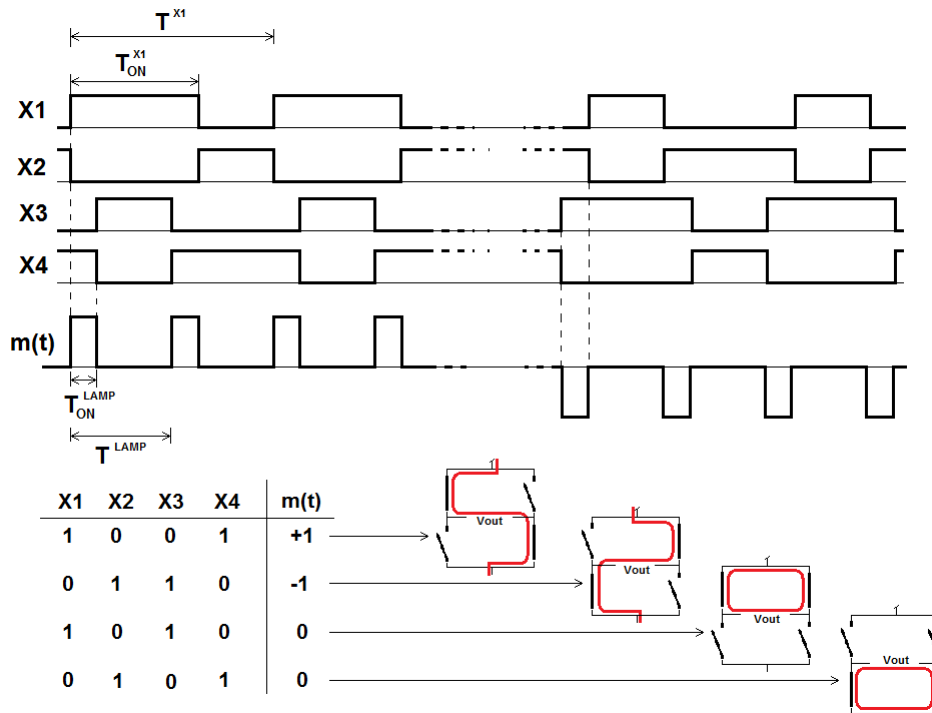


Figura 2.6: Esempi di segnali di pilotaggio dei MOS per ottenere energizzazione positiva [$m(t) = +1$] e negativa [$m(t) = -1$] oppure ricircolo [$m(t) = 0$].

la lettera δ verrà indicato il duty-cycle effettivo sulla lampada (risulta chiaro che i due duty-cycle sono strettamente correlati). Inoltre viene indicata con $m(t)$ la legge di commutazione applicata alla lampada, identificata dai 3 stati precedentemente descritti.

Per realizzare il ricircolo della corrente non vi è la stretta necessità di attivare entrambi i MOS superiori o inferiori, questo perché il *diode di free-wheeling* presente all'interno di ogni MOS di potenza si accende automaticamente in funzione del verso della corrente sul carico. La ricerca di un consumo il più possibile uniforme e bilanciato su tutti e 4 i MOS impone però che il ricircolo della corrente avvenga attraverso il canale del MOS e non tramite il proprio diodo di free-wheeling⁸, per questo motivo la fase di ricircolo avviene attivando entrambi i MOS, superiori o inferiori. Inoltre il MOS dovrebbe garantire una conduzione migliore rispetto al diodo.

Il duty-cycle δ è il vero parametro da controllare in quanto stabilisce per quanto tempo operare energizzazione (positiva o negativa) sulla lampada,

⁸Chiaramente durante i dead-time l'attivazione dei diodi di ricircolo risulta obbligatoria.

mentre il duty-cycle D è il parametro sul quale è possibile operare direttamente attraverso un microcontrollore. In equazione 2.1 viene mostrata la relazione che esiste tra i due parametri.

$$T_{ON}^{X1} = D \cdot T^{X1} = \frac{T^{X1}}{2} + T_{ON}^{LAMP} = \frac{T^{X1}}{2} + \delta \cdot \frac{T^{X1}}{2} = (1 + \delta) \cdot \frac{T^{X1}}{2} \quad (2.1)$$

ottenendo quindi le seguenti relazioni tra i due duty-cycle:

$$D = \frac{1}{2}(1 + \delta) \implies \delta = 2D - 1 \quad (2.2)$$

Per generare correnti positive sulla lampada ed ottenere energizzazione positiva occorre avere un duty-cycle D dei segnali $X1$ e $X4$ superiore al 50%. Viceversa, con duty-cycle D inferiore al 50% la tensione d'uscita del ponte sarà negativa. Si nota che anche per tensioni di lampada tendenti a 0 (la lampada all'accensione tende ad un cortocircuito) il duty-cycle δ tende al 0%, di difficile realizzazione, mentre il duty-cycle D tende al 50%, più 'comodo' da realizzare.

3.2 Rete di pilotaggio dei MOS

I MOS che costituiscono il ponte sono MOS di potenza, in grado di resistere a tensioni V_{DS} superiori a 500 V e correnti I_D di decine di Ampere. Non è perciò pensabile comandare l'accensione e lo spegnimento dei MOS attraverso i segnali provenienti dal microcontrollore che presentano al più una tensione di 3.3 V. Occorre dunque introdurre un circuito (il più delle volte è tutto compreso in unico integrato) in grado di generare opportuni segnali di pilotaggio. Il MOS superiore del ponte ha il source fuori massa, perciò la tensione tale da accendere il dispositivo da applicare al gate deve essere superiore della tensione presente al source, cioè la tensione di uscita del ponte (figura 2.5). Per ottenere tale tensione si utilizza la tecnica denominata di *bootstrap*: tramite il *diodo di bootstrap* viene accumulata sul *condensatore di bootstrap* tutta la tensione necessaria per pilotare il MOS superiore. Questo circuito di carica del condensatore entra in funzione ogni volta che viene acceso il MOS inferiore come mostrato in figura 2.7.

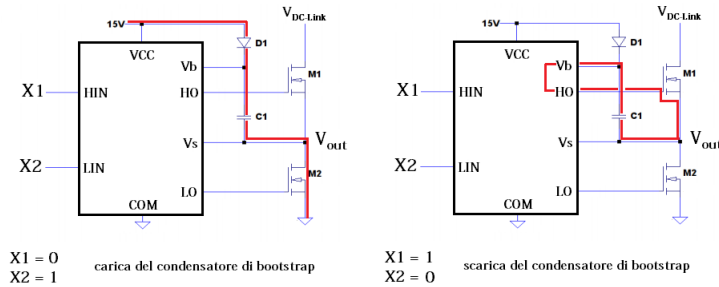


Figura 2.7: Carica e scarica del condensatore di bootstrap ($C1$) mediante il diodo di bootstrap ($D1$) per la generazione della tensione di gate del MOS superiore.

Quando il MOS inferiore è acceso avviene la carica del condensatore di bootstrap attraverso il diodo di bootstrap che si carica della tensione di alimentazione dell'integrato (15 V). All'atto dell'accensione del MOS $M1$ la tensione dal piedino Vb , che risulta essere la somma della tensione V_{OUT} e la tensione sulle armature del condensatore, consente il pilotaggio del gate del MOS $M1$ con il corretto livello di tensione e sfruttando l'energia presente sul condensatore di bootstrap. Il procedimento verrà ripetuto ciclicamente durante il normale esercizio. Il MOS $M2$, avendo il source riferito a massa, non necessita di tensioni aggiuntive a quelle di alimentazione dell'integrato.

L'altra funzione che questi dispositivi sviluppano (per la verità non tutti) è quella di introdurre dead-time al fronte della commutazione dei segnali di pilotaggio per evitare che si manifestino situazioni con entrambi i MOS attivi. Per far ciò i *gate drivers* hanno al loro interno una logica di controllo che monitora costantemente i segnali di pilotaggio portati ai MOS. Esempi di integrati utilizzati come gate driver sono LTC4440, LTC4441, FAN7382, LT1160, ...

Nel progetto implementato è stato utilizzato l'integrato FAN7382 che non prevede quest'ultima funzione descritta: perciò i dead-time saranno da gestire e implementare tramite microcontrollore.

4 Filtro d'uscita

Il filtro scelto è del secondo ordine del tipo LC ed è parte fondamentale del progetto in quanto all'induttore viene dato il compito di calmierare la corrente durante le fasi di energizzazione mentre il condensatore viene utilizzato come bacino di energia per la lampada durante le fasi di ricircolo.

La forma d'onda della corrente che alimenta la lampada è un'onda quadra a bassa frequenza (LF), mentre la frequenza con la quale viene ciclicamente energizzata la lampada è ad un valore elevato (HF). Un valore elevato del condensatore è sconsigliabile in quanto può provocare instabilità legate alla dinamica d'arco, mentre un valore elevato di induttanza (con i giusti compromessi legati all'ingombro via via crescente) è ben desiderabile. L'impedenza presentata dall'induttanza è proporzionale a ω e a L perciò, a parità d'impedenza, un valore elevato di L può portare a lavorare con frequenze HF più basse ottenendo vantaggi in termini di perdite di commutazione dei dispositivi di potenza, ma perdendo in termini di efficienza: un valore d'induttanza basso stressa meno la lampada al momento dell'inversione istantanea a bassa frequenza LF. Inoltre valori elevati di induttanza presentano un magnetico che va in saturazione più facilmente. Dunque la scelta del valore dell'induttanza risulta il frutto di un compromesso. In prima istanza ci si mette con un valore elevato e successivamente, andando ad esaminare le dissipazioni e le forme d'onda si possono trarre le dovute conclusioni.

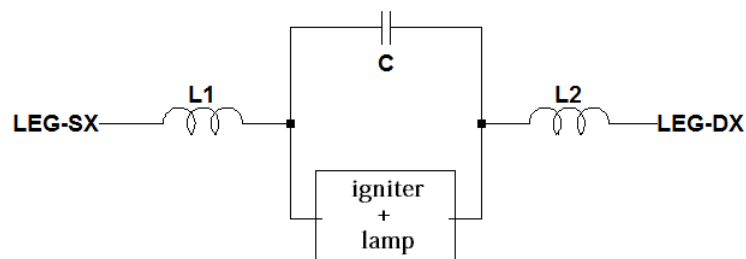


Figura 2.8: Filtro d'uscita LC implementato nel progetto.

Con l'intento di avere un filtro il più simmetrico possibile si è deciso di mettere due induttanze in serie al carico. In questo modo è possibile uti-

lizzare due induttanze di dimensioni più piccole piuttosto che una unica di dimensioni considerevoli.⁹

Per quanto riguarda il fattore di qualità Q del filtro il suo valore è direttamente proporzionale al valore della resistenza offerta dalla lampada che risulta variabile e di difficile caratterizzazione. Ad arco non ancora innescato la resistenza è molto elevata (lampada approssimabile ad un circuito aperto) e di conseguenza anche il fattore di qualità, mentre appena dopo l'innescio dell'arco la resistenza ha valore molto basso, di qualche centinaio di Ohm (si considera la lampada come un cortocircuito).¹⁰

$$Q = R_{LAMP} \cdot \sqrt{\frac{C}{L}} \quad (2.3)$$

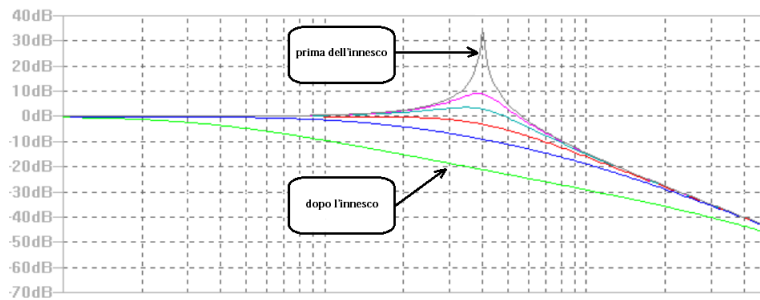


Figura 2.9: Risposta del filtro d'uscita nel dominio delle frequenze.

⁹Due induttanze poste in serie si sommano: $L_{TOT} = L_1 + L_2$.

¹⁰Giunti alla frequenza di risonanza l'impedenza è puramente resistiva, ma risulta difficile determinarne il valore per via di numerosi contributi dati dalle resistenze parassite dei componenti e dall'usura degli elettrodi elettrodi.

5 Circuito d'innesco

L'innesco dell'arco elettrico è un aspetto importante nella vita della lampada. Per innescare l'arco si possono seguire diverse strade: sulla base di quanto esposto nel paragrafo precedente alimentare la lampada ad una frequenza prossima a quella di risonanza del filtro (con la lampada fredda) può risultare sufficiente ad innescare l'arco elettrico. Per far ciò però vi è poco controllo della lampada e c'è la necessità di avere un'alimentazione a frequenza variabile che permette di avvicinarsi alla frequenza di risonanza per poi allontanarsi. Inoltre la frequenza di risonanza non è possibile determinarla con precisione per via di contributi reattivi apportati dalla lampada che non può essere schematizzata come sola resistiva, benché l'approssimazione sia abbastanza coerente.

La soluzione più utilizzata per innescare l'arco (prevista anche in questa progettazione del ballast) prevede l'impiego di un *trasformatore d'impulsi* e un *dispositivo di scarica*. Questi due componenti sono in grado di elevare la tensione ai capi della lampada fino a migliaia di Volt, innescare l'arco per poi disattivarsi e restare inoperativi fino alla successiva accensione.

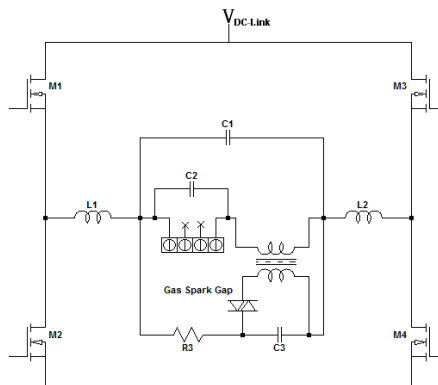


Figura 2.10: Circuito di iniezione costituito dal *trasformatore d'impulsi* e il *Gas Spark Gap*. Al connettore a 4 porte va collegata la lampada a scarica HID.

Si faccia riferimento alla figura 2.10. A lampada fredda e non ancora accesa, la sua impedenza è approssimabile a quella di un circuito aperto, per questo motivo la tensione di alimentazione del ponte ($V_{DC-Link} = 400V$) viene totalmente applicata ai capi del condensatore del filtro C1, in parallelo

al circuito d'innesco.¹¹ Se il ponte viene comandato per un funzionamento *discontinuo* e cioè si alimenta il carico facendo funzionare solo due MOS del ponte posti nella diagonale senza mai accendere i restanti MOS posti sull'altra diagonale (si evitano perciò le fasi di ricircolo), il condensatore si carica fino a raggiungere la tensione sulle sue armature di circa 400 V. Contemporaneamente, con una costante di tempo RC data dal prodotto tra R3 e C3, si carica il condensatore C3. Quando la tensione sulle armature di questo condensatore supera i 350 V entra in conduzione lo Spark Gap (in alternativa si sarebbe potuto utilizzare un DIAC) che porta la tensione sul primario del trasformatore ad impulsi. Sul secondario sarà presente la tensione amplificata dal rapporto spire del trasformatore (solitamente un rapporto largamente superiore a 1:10 che produce una tensione di qualche KV) che può innescare l'arco elettrico della lampada. Qualora l'arco non fosse innescato bisogna attendere una nuova carica del condensatore C3 e riprovare ulteriori tentativi.

In parallelo alla lampada vi è il condensatore C2 al quale è demandato il compito di entrare in risonanza con il secondario del trasformatore d'impulsi. Qualora l'impulso prodotto non generi l'arco, durante la fase di ricarica del condensatore C3, vi sarà uno scambio di energia tra il condensatore e il trasformatore tale per cui la lampada è sottoposta a tensione (seppur piccola) anche durante questa fase. Un valore elevato del condensatore C2 fa sì che l'oscillazione sia consistente a discapito però di un impulso sempre meno piccato. In figura 2.11 è stato simulato il circuito d'innesco mediante *LTspice* e ne sono riportate le grandezze d'interesse con particolare riferimento a due diverse simulazioni: la prima con $C2=2200\text{ fF}$ e la seconda con $C2=220\text{ pF}$.

¹¹Non essendoci corrente si può trascurare le cadute di tensione sui MOS e sulle induttanze di filtro L1 e L2.

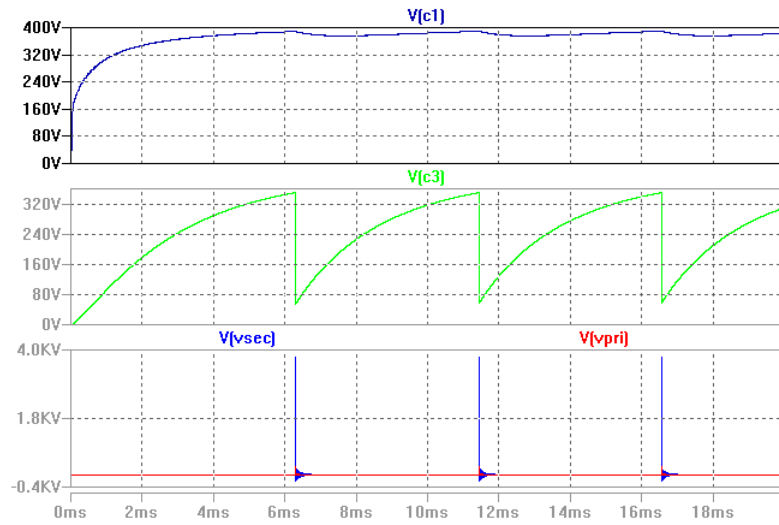


Figura 2.11: Simulazione attraverso *LTspice* del circuito di iniezione: la prima onda rappresenta la carica del condensatore C1, mentre la seconda quella del condensatore C3. Nell' ultimo pannello vengono mostrate le tensioni al primario e al secondario del trasformatore d'impulsi nel caso di $C2=2200$ fF.

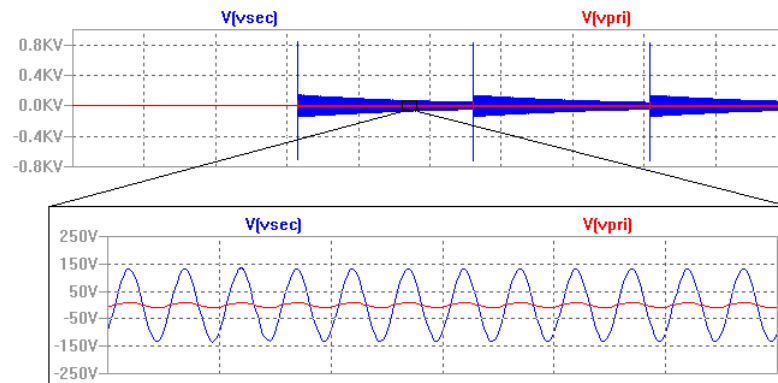


Figura 2.12: Tensioni al primario e al secondario del trasformatore d'impulsi nel caso di $C2=200$ pF. Nello zoom è possibile visualizzare i residui di energia intrappolati in un non ben definito circuito risonatore che genera queste ondulazioni.

6 Sensing di corrente e tensione

Il controllo della lampada viene gestito attraverso il microcontrollore che osservandone parametri d'interesse quali ad esempio la corrente di lampada e la tensione di lampada (quindi in pratica un controllo diretto sulla potenza di lampada) decide di operare opportunamente. Lo stesso microcontrollore dovrà gestire le fasi di innesco e warmup generando adeguati segnali di pilotaggio dei MOS.

6.1 Sensing di tensione

Il sensing della *tensione di lampada* viene eseguito attraverso un amplificatore differenziale ad elevata reiezione di modo comune (CMRR) che per aumentarne l'immunità dai disturbi di modo comune utilizza un amplificatore nella tipica architettura degli operazionali da strumentazione. La tensione di lampada viene prelevata direttamente dal condensatore del filtro e non presenta perciò elevate differenze di modo comune, mentre invece presentano elevate escursioni dinamiche le singole tensioni se riferite alla massa. Attraverso una prima rete resistiva le tensioni ai capi della lampada vengono riportate all'interno di un range di valori tali da poter esser processati dall'amplificatore da strumentazione. Per questa rete resistiva è bene che la massa sia quella di potenza e non quella di segnale in quanto se così non fosse porterei il rumore di modo comune sulla massa dei circuiti di segnale. Successivamente vi sono i due amplificatori non invertenti e l'amplificatore differenziale a costituzione dell'amplificatore da strumentazione. Il guadagno dell'intera architettura è stabilito dalla regolazione del trimmer posto nello stadio iniziale (il guadagno dell'amplificatore differenziale invece è ben bilanciato per ridurre al minimo la risposta di modo comune). Il valore che deve assumere il trimmer è ben esplicitato nel paragrafo 5.1 del capitolo 4. Infine un'ulteriore rete resistiva porta il range di tensioni in uscita dall'amplificatore (0-15 V) ad un range consentito per il microcontrollore (0-3.3 V).

La tensione di lampada periodicamente (con frequenza LF) cambia la sua polarità: perciò di amplificatori differenziali ne sono stati utilizzati due, con ingressi invertiti e messi successivamente in *OR* con due diodi. In questo

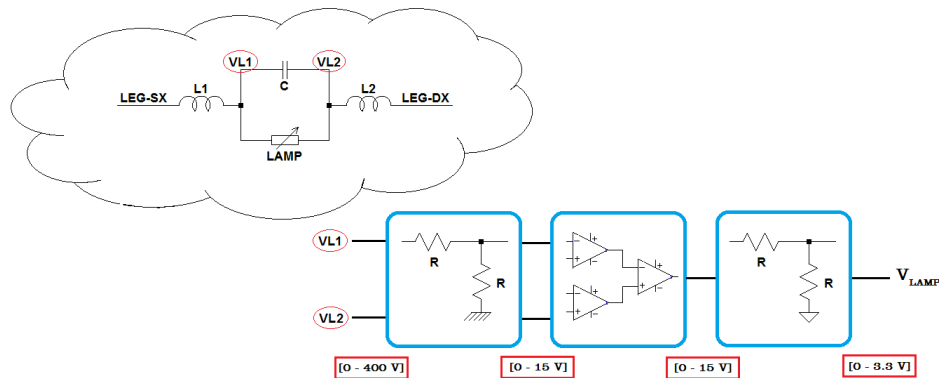


Figura 2.13: Schematizzazione della rete di sensing della tensione di lampada. Il segnale V_{LAMP} viene portato come ingresso al microcontrollore.

modo la tensione V_{LAMP} che viene monitorata e letta dal microcontrollore assumerà valori solo positivi, da 0 a 3.3 V.

Infine, sono state predisposte due reti resistive (dei semplici partitori) che permettono di scalare adeguatamente il valore della tensione di rete e della tensione sul DC-Link. In questo modo il microcontrollore potrà avere informazione anche su queste due tensioni (V_{RETE} e V_{BUS}) e operare scelte al fine di migliorarne le performance (ad esempio attivare il PFC solo se la tensione di rete supera una certa soglia e in caso contrario comunicare, tramite onde convogliate, che la tensione di rete risulta fuori specifica).

6.2 Sensing di corrente e reti di protezione

Il sensing della corrente di lampada risulta più complicato di quanto si possa immaginare: un sensore posto in serie alla lampada che ne rilevi la corrente che circola tra gli elettrodi comprometterebbe di non poco il normale funzionamento della lampada. La misura della corrente che circola sulle bobine del filtro sarebbe già un buon risultato, ma anche in questo caso rilevarla non è per niente agevole. Si è optato infine per un resistore di sensing posto tra il source dei MOS inferiori e la massa (come riportato in figura 2.5). Tale resistenza è in grado di fornire informazioni solo negli istanti di energizzazione (positiva o negativa), permettendo di risalire alla corrente di lampada attraverso una misura di tensione.¹² Requisiti fondamentali per il resistore di

¹²Scegliendo un resistore da 1 Ω la conversione è 1:1.

sensing sono che presenti una bassa induttanza e che sia in grado di dissipare agevolmente potenza. La tensione così generata (che nel progetto verrà denominata I_{LAMP}) viene filtrata da un filtro RC per rimuovere possibili spike di tensione che produrrebbero errate letture di corrente.

Viene implementata una rete analogica di controllo e protezione di tutto il circuito da sovracorrenti che possono circolare sul ponte in caso di malfunzionamento o rottura della lampada. La scelta di implementare un'apposita rete analogica per il controllo del ponte in caso di sovracorrente è dettata dal fatto che gestire la possibile failure attraverso il microcontrollore potrebbe non essere adeguatamente immediato ed efficace. La soluzione adottata prevede l'utilizzo di un comparatore a soglia nel quale un ingresso è la I_{LAMP} e il secondo ingresso è un soglia che viene generata dal microcontrollore.¹³ Nel progetto vien data la possibilità di generare la soglia in due modi diversi: la prima attraverso il convertitore DAC presente all'interno del microcontrollore mentre la seconda, qualora il microcontrollore fosse sprovvisto del DAC, tramite l'utilizzo di un segnale PWM opportunamente filtrato da una rete RC che ne estrae il valor medio. Il secondo caso è quello che è stato implementato nel progetto in quanto il microcontrollore non possiede il convertitore DAC. Il segnale in uscita dal comparatore (I_{OK}) verrà portato in ingresso sia al microcontrollore che così potrà tenere traccia delle volte in cui è scattata la protezione¹⁴, sia ad un'ulteriore rete di condizionamento che va a disabilitare i comandi di pilotaggio dei MOS.

Con lo stesso principio di funzionamento viene implementata un'altra rete analogica di controllo che entra in funzione nel caso in cui la precedente rete subisca danni e non riesca ad esercitare la sua funzione di controllo delle sovracorrenti. Tale comparatore ha una soglia più alta (7,5 V) realizzati con un partitore resistivo sulla tensione di alimentazione di 15 V. La soglia è volutamente posta più alta in modo che questa rete entri in funzione solo in casi eccezionali e si dia la priorità alla rete che genera I_{OK} . Il segnale generato (I_{MAX}) viene portato al microcontrollore che rileva l'eventuale anomalia e di

¹³La scelta del valore della suddetta soglia viene fatta esaminando quale sia il vincolo più stringente tra la saturazione del magnetico delle induttanze oppure la massima corrente sopportata dai MOS del ponte.

¹⁴Questo segnale risulterà fondamentale per la gestione dell'algoritmo di warmup in limitazione di corrente.

tutta risposta spegne tutti e 4 i MOS.

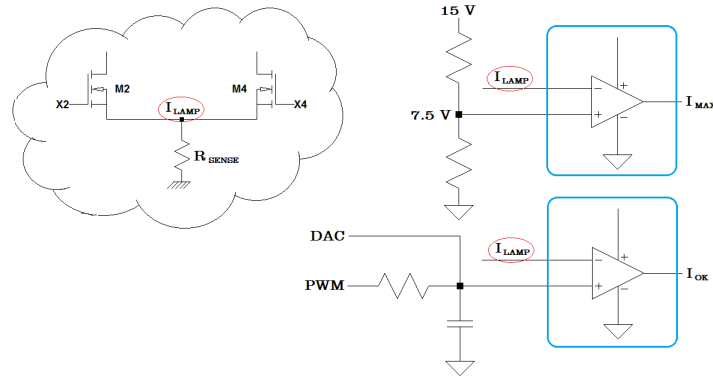


Figura 2.14: Schematizzazione della rete di sensing della corrente di lampada. I segnale I_{LAMP} , I_{OK} e I_{MAX} vengono portati come ingressi al microcontrollore, mentre i segnali DAC e PWM sono generati dal microcontrollore.

Ci si sofferma ora sulla rete di condizionamento per spegnere i MOS a fronte di avvenuta superamento della soglia che ne identifica la sovracorrente. Per farlo si passerà ad una digitalizzazione dei segnali che verranno considerati come 1 (tensione di 3.3 V) o 0 (tensione di 0 V). Valori intermedi di tensione non sono contemplati. Il segnale I_{OK} è uguale a 1 fintanto che la corrente di lampada non supera la soglia (in tal caso I_{OK} vale 0). La prima soluzione potrebbe esser quella di mettere tutti i segnali di comando dei MOS in *AND* con il segnale I_{OK} di modo che tutti i MOS vengano spenti in caso di sovracorrente. Tale soluzione risulta però poco performante: immaginando di metterci proprio in questa condizione di rilevata sovracorrente la rete di condizionamento spegne tutti i MOS e la corrente di lampada si ritrova inevitabilmente a scendere. Appena scesa al di sotto del valore di soglia del comparatore I_{OK} ritorna ad assumere un valore alto e quindi i MOS ritornano attivi e se ciò che causa sovracorrente non è stato opportunamente ridimensionato (ad esempio un duty-cycle di lampada troppo grande) si manifesterà ancora la condizione di sovracorrente. Il risultato è un segnale I_{OK} che oscilla ripetutamente tra 0 e 1. Occorre perciò pensare ad un miglioramento della rete di condizionamento ad esempio introducendo un *flip-flop* \overline{SR} nel quale i segnali di comando dei MOS sono collegati al Set mentre I_{OK} è collegato al Reset (entrambi in configurazione negata).

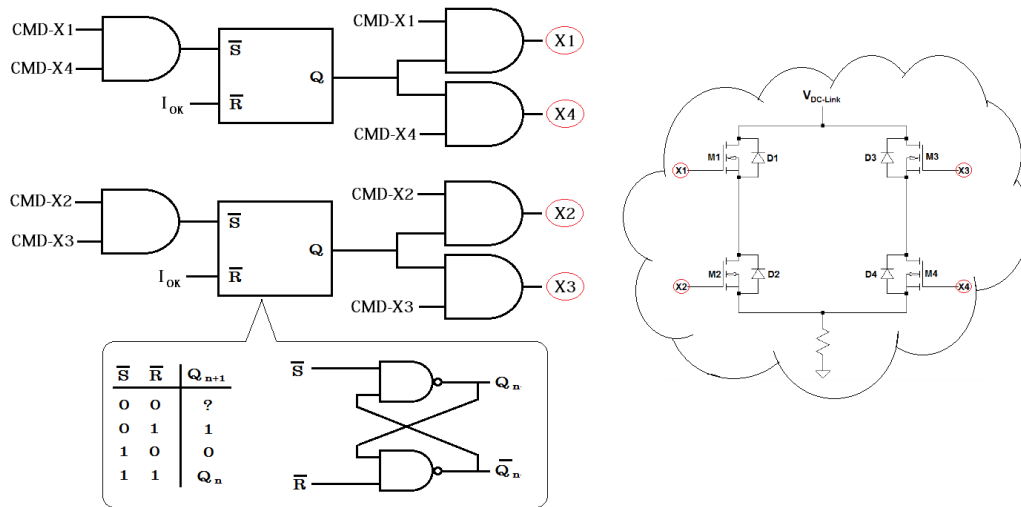


Figura 2.15: Rappresentazione della rete di condizionamento dei segnali di pilotaggio provenienti dal microcontrollore *CMD-X1*, *CMD-X2*, *CMD-X3* e *CMD-X4* mediante flip-flop \overline{SR} realizzato a porte *NAND*.

Come mostrato in figura 2.15 la rete di condizionamento è divisa in due blocchi identici: il primo gestisce i comandi che generano energizzazione positiva (X1 e X4), mentre il secondo quelli di energizzazione negativa (X2 e X3). Quando gli ingressi del flip-flop sono entrambi a 1 l'uscita viene mantenuta, mentre la configurazione con entrambi gli ingressi posti a 0 non è contemplata; se invece l'ingresso I_{OK} viene portato a 0 (vi è cioè sovracorrente) l'uscita del flip-flop viene portata a 0¹⁵, mentre se è il segnale di pilotaggio dei MOS a passare al valore basso, l'uscita viene posta ad 1.¹⁶ Se ne deduce che l'uscita del flip-flop è sempre al valore 1 al di fuori di quando si verifica sovracorrente. Con ulteriori due AND si separano nuovamente i due segnali di pilotaggio dei MOS che risultano così mascherati dal segnale di controllo generato dal flip-flop.

Si prenda come esempio la rilevazione di sovracorrente durante la fase di energizzazione positiva (X1=1, X2=0, X3=0, X4=1). I_{OK} passa dal valore alto al valore basso, portando l'uscita del flip-flop ad un valore basso; in questo modo tutti MOS vengono spenti e la sovracorrente positiva sulla lampada

¹⁵Consiste nell'operazione di *Reset* del flip-flop, ma in versione negata.

¹⁶Consiste nell'operazione di *Set* del flip-flop, ma in versione negata.

inizia a circolare sui diodi di ricircolo D2 e D3 che generano una configurazione di energizzazione negativa. La corrente subisce una forte inversione di pendenza che quindi inizia a scendere e questa situazione perdura fino all'esaurimento di quella che sarebbe stata la fase di energizzazione positiva. Seguirà poi la fase di ricircolo e terminata anche essa verrà eseguito il Set del flip-flop che ne identifica il ripristino al normale esercizio.

7 Flyback converter

Nel progetto è previsto l'utilizzo di un *convertitore flyback*¹⁷ a bassa potenza in grado di fornire il giusto voltaggio per l'alimentazione dei dispositivi integrati (PFC, gate drivers, integrati per il sensing delle tensioni e correnti e non ultimo il microcontrollore). Le tensioni ottenute sono di 5 V e 15 V. Questo stadio è composto principalmente da un trasformatore con avvolgimento sul secondario in controfase rispetto al primario¹⁸ e da un controller integrato *Switched Mode Power Supply* (SMPS).

Il principio di funzionamento prevede l'accumulo e il rilascio di energia da parte del trasformatore nel suo circuito magnetico. Chi regola gli istanti di accumulo e di rilascio è lo switching controller che mediante un semplice MOS al suo interno apre e chiude il circuito collegato al primario del trasformatore. Il flusso magnetico che si viene a creare nel nucleo del trasformatore fa sì che ai due filamenti presenti sul secondario, mediante un opportuno dimensionamento del numero di spire, si possano prelevare le due tensioni. Durante la fase in cui il MOS chiude il circuito sul primario si ha il trasferimento di energia dal DC-Link al nucleo del trasformatore. Il diodo sul secondario risulta interdetto (la tensione d'uscita dal trasformatore è negativa), ma l'energia in uscita dal flyback viene fornita dal condensatore. Durante la fase in cui il MOS è aperto l'energia al carico è fornita dal trasformatore.

L'integrato di switching regola la tensione d'uscita dal trasformatore attraverso un sensing diretto della tensione del primario ed elimina quindi la necessità di una rete che preleva la tensione del secondario, riducendone la complessità. Per alti livelli di potenza viene utilizzato un controllo in fre-

¹⁷Particolare implementazione di un *convertitore buck-boost*.

¹⁸Gli avvolgimenti sul secondario saranno due per prelevare le due tensioni necessarie.

quenza e in corrente, mentre per bassi livelli di potenza e in assenza di carico il controllo entra in modalità *burst*, limitando così consumi e rumori EMI.

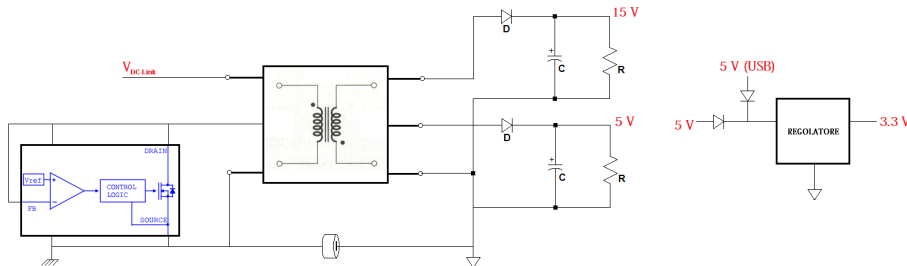


Figura 2.16: Convertitore flyback e regolatore di tensione.

Siccome la tensione di alimentazione del microcontrollore e del circuito di condizionamento della corrente di lampada (paragrafo 6.2) è di 3.3 V vi è la necessità di aggiungere un *regolatore di tensione* che fornisca in uscita 3.3 V stabili a fronte di 5 V in ingresso. Questa soluzione risulta necessaria in quanto anche l'alimentazione proveniente dal collegamento *micro USB* per il debug del microcontrollore è di 5 V. Un semplice *OR* realizzato mediante due diodi pone le due sorgenti a 5 V in ingresso al regolatore di tensione.

Infine in questo stadio del circuito viene realizzata la separazione della *massa di segnale* dalla *massa di potenza* mediante una ferrite opportunamente scelta.

8 Microcontrollore

Il microcontrollore è il cervello dell'apparato. Implementando opportuni algoritmi di sensing e controllo delle grandezze fondamentali del ballast elettronico è possibile andare a impostare e modificare *run-time* parametri quali le frequenze LF e HF del ponte, il duty-cycle dei segnali PWM, ecc... Sempre tramite microcontrollore si eseguono letture ripetute della tensione di rete e della tensione sul DC-Link operando opportune scelte. Inoltre al microcontrollore è demandato il compito di gestire le onde convogliate che permettono di spedire e ricevere informazioni sulla linea ad alta tensione. Una trattazione più approfondita di questo importante e fondamentale componente sarà svolta nel capitolo 4.

9 Onde convogliate

Nel ballast elettronico progettato vi è anche la possibilità di utilizzare la tecnologia ad onde convogliate (o *Power Line Communication*): un apposito modulo prende in gestione tale funzione al fine di comunicare con il microcontrollore utilizzando la linea d'alimentazione ad alta tensione. Una circuiteria di disaccoppiamento ha il compito di disaccoppiare la parte di elaborazione logica del ballast dalla linea di potenza, consentendo comunque la ricezione e trasmissione di dati. Nell'ambito della progettazione svolta e dei collaudi effettuati questo stadio non è stato preso in considerazione e quindi non sarà trattato nel seguito dell'elaborato.

Capitolo 3

Dimensionamento del ballast elettronico

Verranno ora analizzati i calcoli svolti per l'individuazione e la scelta di tutti i dispositivi utilizzati nella realizzazione del ballast elettronico. Per far ciò è bene aver chiare le *specifiche di progetto*, qui di seguito riportate.

- ★ **Tensione d'ingresso:** $V_{ACrms} = 220 \text{ V} \pm 20\% \implies [176 - 264]\text{V}$
- ★ **Frequenza minima d'ingresso:** $f_{INmin} = 47 \text{ Hz}$
- ★ **Potenza massima erogata alla lampada:** $P_{OUT} = 200 \text{ W}$ (si è tenuto un margine di 50 W per l'alimentazione (PFC) del DC-Link, la potenza erogata alla lampada sarà nell'intorno dei 150 W).
- ★ **Tensione sul DC-Link** $= V_{DC-Link} = 400 \text{ V}$ (la scelta è dovuta al fatto che il boost opera in salita e la massima tensione di picco in ingresso è 374 V).
- ★ **Sovratensione massima** $= V_{OVP} = 430 \text{ V}$
- ★ **Massima tensione di ripple sul DC-Link** $= \Delta V_{DC-Link} = 5\%$ di $V_{DC-Link} \implies \Delta V_{DC-Link} = 20 \text{ V}$
- ★ **Frequenza di commutazione sui leg** $= f_{HF} = 24 \text{ KHz}$ (la frequenza è stata scelta in modo tale da stare al di fuori della banda udibile).
- ★ **Frequenza di inversione della corrente** $= f_{LF} = 80 \text{ Hz}$ (la scelta di un multiplo della frequenza di rete può causare problemi).
- ★ **Massima temperatura ambiente** $= T_{AMBmax} = 75^\circ\text{C}$

★ **Power Factor** = $PF = 0.9$ (per il soddisfacimento della normativa IEC61000-3-2).

Nei paragrafi a seguire verranno discussi e dimensionati i singoli stadi costituenti il ballast progettato.

1 Dimensionamento dello stadio d'ingresso

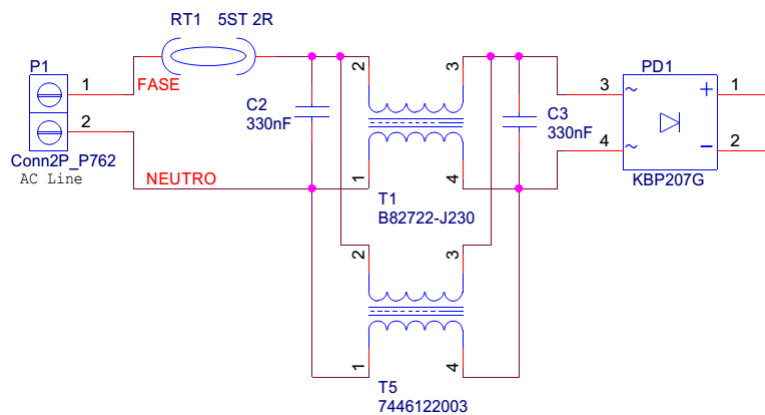


Figura 3.1: Schematico dello stadio d'ingresso.

FUSIBILE

Si è scelto un fusibile MT785225 della categoria microfusibile della *Omega* che può operare per tensioni efficaci fino a 250 V presentando una resistenza nominale di 29 mΩ (72.5 mV la caduta di tensione per una corrente nominale di 2.5 A). Si tratta di un fusibile a caratteristica ritardata così da non entrare in funzione per le correnti di in-rush (la massima corrente riportata nel datasheet è di 35 A). Infine la *nominal melting energy* è $I^2t = 45A^2s$. Occorre prevedere un opportuno derating del 75% della corrente nominale affinché il fusibile non si bruci con le correnti massime di esercizio, dunque quando il fusibile lavora ad una temperatura di 25°C è in grado di far passare correnti di circa 1,8 A. La temperatura incide molto sulla corrente massima che può circolare abbassandola del 10% ogni 100°C.

FILTRO EMI

Per la scelta del *common mode choke* ci si è affidati a quanto visto sulle ana-

loghe evaluation board di *ST*. Un prodotto molto utilizzato è il B82722-j230 della *EPCOS*: composto con un nucleo toroidale in ferrite del tipo “current compensated ring core choke”, presenta una rated voltage di 250 V efficaci (ma è in grado di resistere anche a tensioni di migliaia di Volt per qualche secondo) ed è inoltre consigliato in applicazioni quali ballast o applicazioni switch-mode. Con riferimento al singolo avvolgimento, presenta un valore d'induttanza di 1.2 mH quando è percorso da una corrente DC di 3 A. Viene infine esplicitata la resistenza serie (56 mΩ - 2.5 Ω) e l'induttanza dispersa (17 μH - 800μH). Dai grafici riportati nel datasheet è possibile osservare come tale componente operi il massimo dell'effetto filtrante intorno ai 20 MHz e questo può risultare non sufficiente ad eliminare tutti i disturbi EMI, per questa ragione nello schematico (e di conseguenza anche nella board) vi è stata disposta la possibilità di utilizzare un altro *common mode choke*: la scelta è ricaduta sul prodotto 7446122003 della *WURTH ELEKTRONIK* che presenta, oltre che una piedinatura diversa, un'induttanza di 3.3 mH quando percorso da una corrente di 1.5 A.

I due condensatori di shunt sono di tipo B32932A3334M: di classe X2, sono da 330 nF e sono di tipo *Metallized Polyester Film Capacitors* (MKT).

PONTE RADDRIZZATORE

La scelta del ponte raddrizzatore è ricaduta sul KBP207G. Questo componente è in grado di bloccare fino a 1000 V in inversa e in questa maniera si pone al riparo da eventuali errori di connessione in fase di installazione (applicando una seconda fase al posto del neutro si avrebbero 380 V efficaci) e da eventuali accidentali disconnessione del neutro (il quale, sbilanciandosi, non sarebbe più vincolato al potenziale di centro stella). La corrente media massima sopportata dal ponte è di 2 A. Questo valore risulta sufficiente e ciò viene esplicitato nei calcoli seguenti. La corrente nominale in uscita dal DC-Link è così ottenuta:

$$I_{OUT} = \frac{P_{OUT}}{V_{OUT}} = \frac{200}{400} = 0.5A \quad (3.1)$$

mentre la potenza attinta dalla rete è determinata dal rendimento presunto di PFC e ponte:

$$P_{IN} = \frac{P_{OUT}}{PF} = \frac{200}{0.9} = 222W \quad (3.2)$$

Il valore efficace massimo della corrente d'ingresso è stimabile dalla definizione di PF mettendosi nel caso peggiore di tensione d'ingresso minima (175 V efficaci):

$$PF = \frac{P_{IN}}{V_{INrms} I_{INrms}} \Rightarrow I_{INrms} = \frac{P_{IN}}{PF \cdot V_{INrms}^{min}} = \frac{222}{0.9 \cdot 175} = 1.41A \quad (3.3)$$

da cui è facile ottenere la corrente di picco:

$$I_{INpk} = \sqrt{2} I_{INrms} = \sqrt{2} \cdot 1.41 = 2A \quad (3.4)$$

La corrente efficace sui diodi del ponte rettificatore, conducendo per un semiperiodo a coppie di due, è:

$$I_{Drms} = \frac{1}{\sqrt{2}} I_{INrms} = \frac{1.41}{\sqrt{2}} = 1A \quad (3.5)$$

mentre la corrente media sulla singola semionda è:

$$I_{D0} = \frac{I_{INpk}}{\pi} = \frac{2}{\pi} = 0.64A \quad (3.6)$$

Infine si calcola la potenza dissipata dal ponte rettificatore calcolandone prima la potenza istantanea dissipata dal singolo diodo p_{Ddiss} (una parte è dovuta alla soglia di accensione del diodo, una parte dalla resistenza serie di conduzione), trovando successivamente la potenza media del singolo diodo P_{Ddiss} ed infine la potenza media dissipata dal ponte P_{PONTE} (in conduzione ci sono 2 diodi per semionda).

$$p_{Ddiss} = v_D i_D = (V_{th} + R_d i_D) i_D = V_{th} \cdot i_D + R_d \cdot i_D^2 \quad (3.7)$$

$$P_{Ddiss} = \int_0^T p_{Ddiss} dt = V_{th} \cdot \int_0^T i_D dt + R_d \cdot \int_0^T i_D^2 dt = V_{th} \cdot I_{D0} + R_d \cdot I_{Drms}^2 \quad (3.8)$$

$$P_{PONTE} = 2 \cdot (V_{th} \cdot I_{D0} + R_d \cdot I_{Drms}^2) = 2 \cdot (V_{th} \cdot 0.64 + R_d \cdot 1) \quad (3.9)$$

La nota negativa riscontrata nella scelta di questo componente risiede nella *nominal melting energy* pari a $15 A^2s$. Questo valore risulta più basso di quello del fusibile ($45 A^2s$), ma questo componente era già disponibile in UMPI e quindi, in una prima fase progettuale, si è deciso di soprassedere questo aspetto.

2 Dimensionamento dello stadio PFC

Verranno ora analizzati il dispositivo integrato (L6564) e i dispositivi di potenza (la *capacità d'ingresso*, il *MOS*, il *boost inductor*, il *diodo*, il *diodo di in-rush* ed infine la *capacità d'uscita*) costituenti lo stadio PFC.

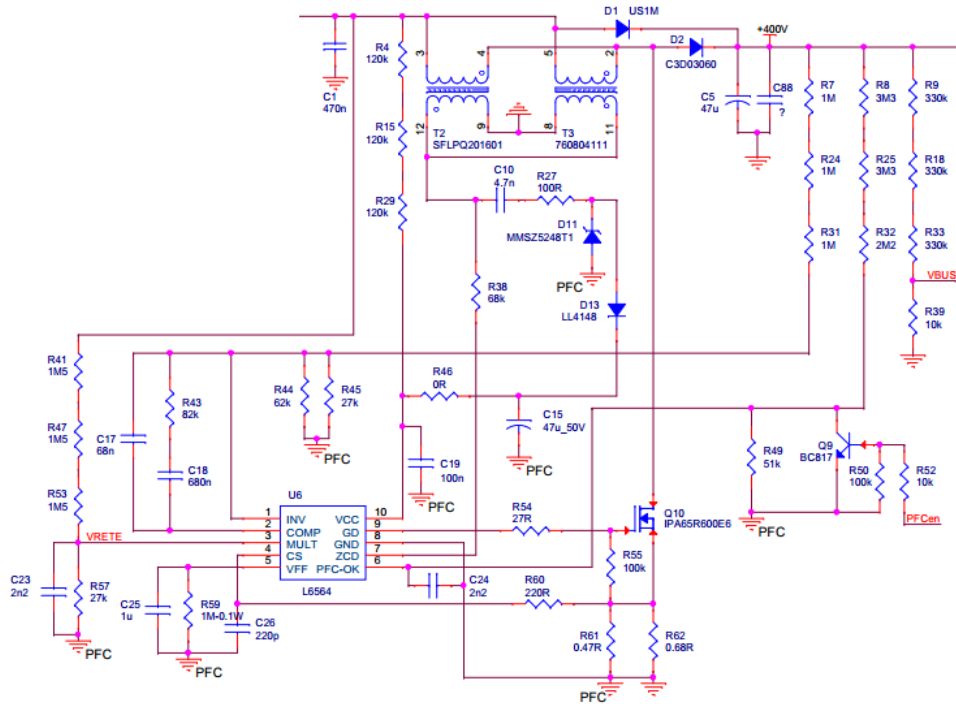


Figura 3.2: Schematico dello stadio PFC.

CAPACITÀ D'INGRESSO

La capacità d'ingresso viene utilizzata per attenuare le oscillazioni in uscita dal ponte dovute alle commutazioni del circuito boost a valle, in modo da garantire una tensione pressoché costante sull'induttore se valutata in un periodo di commutazione HF, ma da consentire la dinamica LF relativa alla tensione sinusoidale raddrizzata in uscita dal ponte. Di solito si sceglie un ripple massimo ammesso pari al 5-50% del picco della minima tensione efficace di ingresso (247 V), ovvero 12-50 V. Si è scelto perciò un condensatore in poliestere del valore di $C1=470$ nF. Per tale componente è opportuna una bassa ESR (resistenza serie equivalente).

MOS DEL BOOST

Il MOS Q10 utilizzato nello stadio boost PFC è lo stesso impiegato nel full-bridge: IPA65R600E6 CoolMOS della *INFINEON*. Presenta una tensione di breakdown di 700 V e la corrente continua che può gestire supera i 4.6 A (riuscendo a reggere correnti impulsive fino al massimo di 18 A). Il MOS, a package completamente isolato, presenta una resistenza termica verso il case di $R_{thJC} = 4.5^{\circ}C/W$, mentre la resistenza termica verso l'ambiente è $R_{thJA} = 80^{\circ}C/W$. Analizzando i fattori dissipativi la R_{DSon} è di 0.6Ω a $25^{\circ}C$, ma sale fino a 1.4Ω a $150^{\circ}C$, mentre invece le perdite di commutazione dipendono dalla frequenza di commutazione che, come detto nel capitolo 2 paragrafo 2, è variabile ed è in funzione dell'ampiezza della sinusoide.

DIODO DI FREEWHEELING

Il diodo D2 è quello costituente il circuito boost del PFC. E' stato scelto il *CREE Schottky Diode* C3D03060 in grado di resistere ad un massimo di 600 V e 3 A in continua (alla temperatura di $100^{\circ}C$). La sua carica di recovery è di 6.7 nC.

DIODO DI IN-RUSH

E' predisposto l'utilizzo anche del diodo D1, che entra in conduzione per evitare la saturazione del magnetico dell'induttore di boost allo startup. Deve perciò essere in grado di resistere alle correnti di in-rush per poi non entrare più in funzione durante il normale esercizio. É stato scelto l' Ultrafast Diode US1M della VISHAY, in grado di reggere a correnti di picco di 30 A.

BOOST INDUCTOR

Con riferimento all'immagine 2.3 del capitolo 3, appurato che il PFC lavorerà in transition mode, si calcolano ora le correnti che scorrono sull'induttore una volta note la I_{INrms} di equazione 4.3 e la I_{INpk} di equazione 4.4.

$$I_{Lpk} = 2 \cdot I_{INpk} = 2 \cdot 2.0 = 4A \quad (3.10)$$

infatti il valore di picco della corrente che circola sull'induttore raggiunge il doppio del picco della corrente assorbita dalla rete. Questo valore torna utile per dimensionare il nucleo dell'induttore ed evitare che finisca in saturazione. Il valore efficace della corrente sull'induttore è calcolato assumendo

che la forma di corrente è un'onda triangolare¹. Il valore efficace calcolato in prossimità del valore di picco è:

$$I_{Lrms}^{pk} = \frac{I_{Lpk}}{\sqrt{3}} \quad (3.11)$$

ma nella realtà l'onda triangolare è modulata in seno, quindi si ottiene un valore efficace della corrente di induttore pari a:

$$I_{Lrms} = \frac{1}{\sqrt{2}} \frac{1}{\sqrt{3}} \cdot I_{Lpk} = \frac{4}{\sqrt{6}} = 1.63A \quad (3.12)$$

Il valore dell'induttore di boost viene determinato considerando anche la minima frequenza al quale il convertitore deve operare. Tale frequenza, presente al valore di cresta della sinusoide d'ingresso, deve essere superiore alla banda dell'udibile (20 KHz). Applicando l'equazione costitutiva dell'induttore nell'intervallo T_{ON} e passando ad una trattazione in radianti si ottiene:

$$v_L = L \cdot \frac{di}{dt} \Rightarrow V_{ACpk} \sin\theta = L \cdot \frac{I_{Lpk} \sin\theta}{T_{ON}} \quad (3.13)$$

dove è stata assunta una forma d'onda di corrente ottimale per avere un fattore di potenza unitario. Il valore di picco della tensione di rete a 220 V efficaci vale 311 V (non considerando però le possibili fluttuazioni del $\pm 20\%$).

$$T_{ON} = L \cdot \frac{I_{Lpk}}{V_{ACpk}} \quad (3.14)$$

constatando che il periodo T_{ON} all'interno della sinusoide dovrebbe rimanere di valore costante ed è inversamente proporzionale al valore di picco della tensione di rete. Essendo I_{Lpk} il doppio di I_{INpk} ed utilizzando la definizione di PF visto in equazione 4.3 si può riscrivere l'espressione di T_{ON} :

$$T_{ON} = L \cdot \frac{2P_{IN}}{V_{ACrms}^2} \quad (3.15)$$

avendo assunto per la corrente sull'induttore un andamento che, a meno del ripple, porterebbe ad un valore di $PF = 1$. Discorso diverso invece per il tempo T_{OFF} che invece risulta essere dipendente da θ :

$$V_{ACpk} \sin\theta - V_{OUT} = L \cdot \frac{-I_{Lpk} \sin\theta}{T_{OFF}} \Rightarrow T_{OFF} = L \cdot \frac{I_{Lpk} \sin\theta}{V_{OUT} - V_{ACpk} \sin\theta} \quad (3.16)$$

¹per le onde triangolari il valore efficace è $V_{rms} = \frac{V_{pk}}{\sqrt{3}}$

$$T_{OFF} = L \cdot \frac{2\sqrt{2}P_{IN}\sin\theta}{(V_{OUT} - \sqrt{2}V_{ACrms}\sin\theta) \cdot V_{ACrms}} \quad (3.17)$$

È ora possibile determinare la frequenza di commutazione dello stadio boost:

$$f_{sw} = \frac{1}{T_{ON} + T_{OFF}} = \frac{1}{2LP_{IN}} \cdot \frac{V_{ACrms}^2 \cdot (V_{OUT} - \sqrt{2}V_{ACrms}\sin\theta)}{V_{OUT}} \quad (3.18)$$

dove si evince che la massima frequenza si ottiene per $\sin\theta = 0$ in corrispondenza dello zero crossing dove i T_{OFF} sono molto brevi, mentre la frequenza minima si ha per $\sin\theta = 1$ in corrispondenza della cresta della tensione di rete dove i T_{OFF} sono lunghi. Dalla 3.18 si nota che se L cresce la frequenza cala (T_{ON} e T_{OFF} si allungano). Scegliendo ora come minima frequenza di commutazione dello stadio PFC il valore $f_{sw}^{min} = 40$ KHz si può determinare il valore di L in presenza della tensione nominale $V_{ACrms} = 220$ V:

$$L = \frac{V_{ACrms}^2 \cdot (V_{OUT} - \sqrt{2}V_{ACrms})}{2 \cdot f_{sw}^{min} \cdot P_{IN} \cdot V_{OUT}} = \frac{220^2 \cdot (400 - 311)}{2 \cdot 40000 \cdot 222 \cdot 400} = 606\mu H \quad (3.19)$$

Se invece la tensione d'ingresso raggiunge il suo valore di minimo ammesso da specifiche (176 V) l'induttanza risultante vale $658 \mu H$, mentre se raggiunge il valore di massimo ammesso (264 V) l'induttanza vale $265 \mu H$. Nel progetto è stato scelto il trasformatore SFLPQ201601 della *ITACOIL* con le seguenti caratteristiche:

95 - 170W					
Codice	Induttanza ±10% , @10KHz-100mV	DCR Tip @20°C Avvolgimento Principale	DCR Tip @20°C Avvolgimento Ausiliario	Rapporto Spire Principale/Aux	Rigidita' Dielettrica Principale/Aux
SFLPQ201601	330μH	335mΩ	295mΩ	10:1	1.0KV



Figura 3.3: Caratteristiche dell'induttore SFLPQ201601 scelto nel progetto.

Questo componente, come riportato nel datasheet, ha una perdita di 0.28 W (legata alle dimensioni del filo e alle caratteristiche del magnetico) ed ha una frequenza minima nominale di commutazione di 73 KHz.

Come per il filtro EMI viene disposta la possibilità di utilizzare un trasformatore alternativo a quello scelto: T3 è il trasformatore 760804111 della *WURTH ELEKTRONIK*. In questo caso la piedinatura è più piccola di quella del trasformatore ITACOIL e il valore dell'induttanza è di 250 μH con una corrente massima di 1.1 A.

CAPACITÀ D' USCITA : DC-Link

Il calcolo della capacità d'uscita viene fatto valutando la scarica della capacità durante il periodo T_{ON} in cui è in conduzione il MOS e il carico continua a richiedere la corrente nominale di 0.5 A. Come da equazione 3.15 il valore di T_{ON} è costante all'interno della semionda della tensione raddrizzata. Raggiunge il massimo valore in corrispondenza della minima tensione efficace di rete (176 V):

$$T_{ON}^{max} = 330 \cdot 10^{-6} \cdot \frac{2 \cdot 222}{176^2} = 4.73 \mu\text{s} \quad (3.20)$$

Ipotizzando di voler contenere il ripple della tensione ad alta frequenza² sul DC-Link ad un valore del 1% allora $\Delta V_{pp} = 4$ V. Mettendo a bilancio la carica sul condensatore $\Delta Q = C_{OUT} \cdot \Delta V_{pp}$ e inoltre essendo $\Delta V_{pp} = T_{ON}^{max} \cdot I_{OUT}$ si ha:

$$C_{OUT} = \frac{T_{ON}^{max} \cdot I_{OUT}}{\Delta V_{pp}} = \frac{4.73 \cdot 10^{-6} \cdot 0.5}{4} = 591 \text{ nF} \quad (3.21)$$

Tuttavia il DC-Link è soprattutto soggetto al ripple a bassa frequenza per il fatto che la corrente di ricarica deve inseguire l'andamento sinusoidale della tensione di rete, al fine di ottenere un buon PF.

Esaminando la figura 3.4 sono rappresentati gli andamenti della corrente di carica ($I_{pk} \sin \theta$) e scarica (I_{OUT}) del DC-Link. Il rapporto che lega i due contributi è quello del valore medio per una onda sinusoidale raddrizzata.

$$I_{pk} = I_{OUT} \cdot \frac{\pi}{2} \quad (3.22)$$

L'angolo θ_0 è l'angolo per il quale la funzione seno assume il valore I_{OUT} . Il suo valore è $\arccos \sin \frac{2}{\pi} \simeq 40^\circ = 0.22 \text{ rad}$. L'area rossa rappresenta la carica ΔQ_+

²con *alta frequenza* viene indicata la frequenza di commutazione del PFC all'interno della semionda di rete

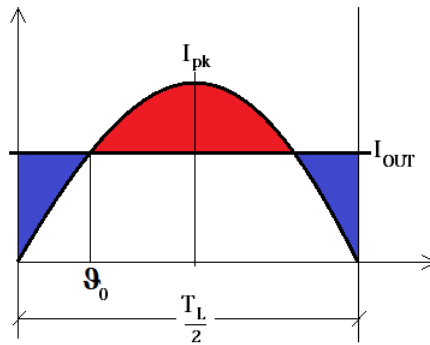


Figura 3.4: Corrente di carica del DC-Link (di natura sinusoidale) e scarica del DC-Link (corrente nominale I_{OUT}).

mentre l'area blu rappresenta ΔQ_- . In via approssimativa si può considerare il contributo ΔQ_- come il doppio dell'area del triangolo che ha per base I_{OUT} e per altezza θ_0 .

$$\Delta Q = I_{OUT} \cdot \frac{\theta_0 \cdot T_L}{2\pi} = \frac{20 \cdot 10^{-3} \cdot 0.22 \cdot 0.5}{2\pi} = 0.35mC \quad (3.23)$$

A questo punto, tramite la nota equazione $\Delta Q = C_{OUT} \cdot \Delta V_{pp}$ e detto che ΔV_{pp} è uguale a 4 V, si determina la capacità del DC-Link:

$$C_{OUT} = \frac{\Delta Q}{\Delta V_{pp}} = \frac{0.00035}{4} = 87.5\mu F \quad (3.24)$$

Stando a questi calcoli la scelta del condensatore del DC-Link è ricaduta sul parallelo di due condensatori: C5 è un elettrolitico da 47 μF di tipo Aluminium Electrolytic Capacitor e-vz della *NICHION*, mentre C88 è un condensatore ceramico della *EPCOS* Metallized Polypropylene Film Capacitor MKP_B32653 da 470 nF con un valore di ESR trascurabile.

La scelta di un condensatore più piccolo di quanto appena stimato genererà più ripple di quanto ipotizzato: se il ripple dovesse risultare un problema ci si riserva di aggiungere un ulteriore condensatore in parallelo da 47 μF .

2.1 L'integrato per lo stadio PFC: L6564

Nel ballast progettato si è deciso di utilizzare uno stadio boost operante in *transition mode* e l'integrato utilizzato a questo scopo è l' **L6564** della *STMicroelectronics*.

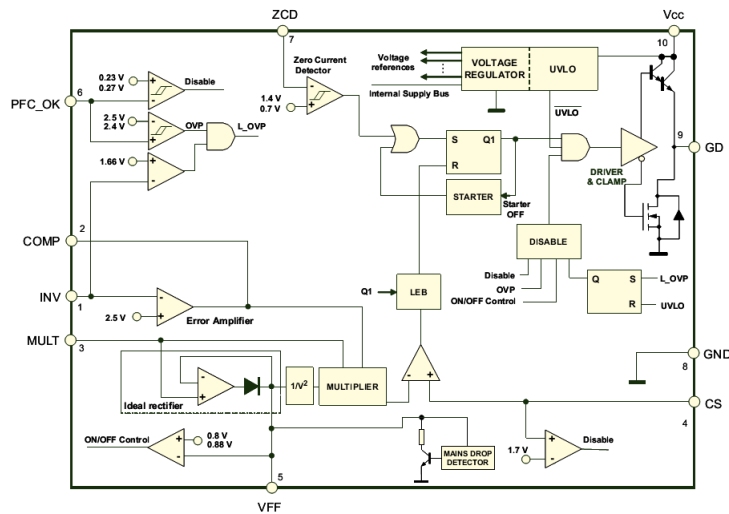


Figura 3.5: Diagramma a blocchi interno dell'integrato L6564 della STMicroelectronics preso dal suo datasheet.[7]

Il moltiplicatore (*MULTIPLIER*) al suo interno viene utilizzato per generare il set point della corrente in funzione della tensione di rete (proveniente da *MULT*) e dell'errore sulla tensione quasi continua sul DC-link (proveniente da *INV*). Dunque all'uscita del moltiplicatore vi è il riferimento per lo spegnimento del MOS. Il dispositivo L6564 è provvisto di un circuito per la correzione della distorsione di crossover al passaggio dallo zero della tensione di rete ³. La stabilità dell'anello di retroazione per la lettura della tensione viene migliorata utilizzando una rete di *feed-forward*⁴ che migliora il termine moltiplicativo del moltiplicatore leggendo il valore di cresta della tensione di rete. In questa maniera si lavorerà sempre a potenza costante, qualsiasi sia la tensione applicata in ingresso. Inoltre, qualora la tensione d'ingresso presentasse variazioni brusche, il circuito adottato per l'individuazione del fattore moltiplicativo è dotato di una notevole dinamica grazie all'intervento di un circuito non lineare (*MAINS DROP DETECTOR*) e di un rilevatore di cresta.

³tali accorgimenti migliorano il fattore di potenza PF e la distorsione armonica totale della corrente d'ingresso THD

⁴col termine *feed-forward* viene indicato un controllo effettuato in modo predittivo, senza generare anelli di retroazione, a tutto vantaggio della stabilità, esaminando l'entità di un disturbo e intraprendendo l'azione corretta per compensarlo sulla base della previsione dell'effetto che produrrebbe sull'uscita.

Questo dispositivo di correzione del Power Factor presenta anche una protezione da sovratensioni (*OVP*) attraverso il quale viene sospesa la ricarica di energia sul condensatore DC-link nel caso in cui la tensione in uscita eccedesse un livello di guardia. Il circuito di protezione da tensioni troppo basse (*UVLP* altrimenti detto *main-brownout*) viene utilizzato per spegnere il dispositivo qualora la tensione di accensione del MOS risultasse troppo debole per garantirne un'accensione sicura.

Esiste infine un circuito di triggering con riarmo per l'identificatore della condizione di *zero corrente* nell'induttanza di boost, osservando se il nodo di drain del MOS è a potenziale V_{OUT} (a meno della caduta di tensione sul diodo) oppure è flottante.

2.2 Rete di polarizzazione di L6564

Con riferimento alle figure 3.2 e 3.5 viene ora descritto il dimensionamento della rete di polarizzazione per il dispositivo L6564, analizzando ogni singolo piedino dell'integrato.

Prima però è bene precisare che il guadagno del moltiplicatore non può essere un valore fissato a priori, ma deve poter adattarsi al valore di tensione di alimentazione. Infatti, se così non fosse, il valore della corrente attinta dalla rete varierebbe proporzionalmente con la tensione di rete e questo renderebbe il guadagno di potenza del PFC proporzionale al quadrato della tensione efficace di rete. Risulta perciò conveniente esplicitare il guadagno del moltiplicatore (che genera il segnale V_{CS} ⁵, cioè la soglia di spegnimento del MOS) come funzione del valore di picco della tensione in ingresso. L'espressione di V_{CS} risulta la seguente:

$$V_{CS} = V_{CS_OFFSET} + k_m \cdot \frac{(V_{INV} - 2.5) \cdot V_{MULT}}{V_{FF}^2} \quad (3.25)$$

Il prodotto tra V_{MULT} e $V_{INV} - 2.5$ rappresenta l'uscita del moltiplicatore (sarebbe più opportuno considerare l'uscita compensata dalla rete di compensazione - V_{COMP} - piuttosto che V_{INV}), moltiplicato per un fattore di

⁵la tensione V_{CS} deve essere una sinusoide raddrizzata, in fase con la tensione di rete, la cui ampiezza è proporzionale all'errore sulla tensione d'uscita

scala k_m che riporta l'uscita del moltiplicatore nel range tra 0 e 1 V. La divisione per V_{FF}^2 (tensione di picco della sinusoide d'ingresso) viene attuata per tener in considerazione che il picco può essere variabile, mentre l'addizione del termine V_{CS_OFFSET} viene utilizzata per gestire la distorsione di crossover presente al passaggio per lo zero della tensione d'ingresso (e quindi vi è l'annullamento della potenza attinta). Aggiungendo il termine V_{CS_OFFSET} si vuole forzare il trasferimento di più energia di quanto strettamente necessaria così da minimizzare l'intervallo di tempo in cui non viene attinta potenza attiva dalla rete.

PIN 1: INV

A questo pin viene collegato l'anello di retroazione della tensione in uscita per poterla quindi regolare: tale tensione risulta essere quindi uno dei due ingressi dell' *Error Amplifier* (EA) e inoltre viene controllata da un comparatore per la protezione da sovratensioni sul DC-link. Il riferimento interno dell'Error Amplifier è di 2.5 V, perciò i 400 V sul DC-link (tensione imposta da prerogative di progetto) devono essere scalati attraverso un partitore resistivo. Al fine di limitare la corrente in ingresso al piedino (al massimo 1 μ A) la corrente che circola sull'intero partitore deve essere di due ordini di grandezza superiore: scelto un valore di 3 M Ω (realizzato attraverso la serie di 3 resistenze da 1 M Ω) per il lato alto del partitore, il lato basso deve valere 18.8 K Ω , ottenuto approssimativamente tramite il parallelo di due resistenze, una da 62 K Ω e una da 27 K Ω .

PIN 2: COMP

A questo pin viene collegata una rete compensatrice di tipo bassa-basso in grado di attenuare adeguatamente il ripple presente sulla tensione d'uscita a 100Hz. Sul piedino *COMP* infatti troviamo l'uscita dell'Error Amplifier che deve essere data al moltiplicatore insieme alla copia scalata del segnale d'ingresso. La rete compensatrice deve essere posta tra l'uscita dell'Error Amplifier e il pin *INV*. Potrebbe risultare sufficiente una rete a singolo polo con frequenza di taglio a 20-30Hz, ma per garantire maggiormente un sufficiente guadagno DC si è preferito una rete composta da due poli e uno zero, come suggerito da *ST*. La rete viene realizzata tramite una resistenza

($R_{43}=82\text{ K}\Omega$) e due condensatori ($C_{17}=68\text{ nF}$ e $C_{18}=680\text{ nF}$) che generano una frequenza di cross-over attorno ai 12 Hz.

PIN 3: MULT

Costituisce il secondo ingresso del moltiplicatore (il primo è l'uscita dell'Error Amplifier): rappresenta la tensione, scalata tramite partitore, della sinusoide AC raddrizzata d'ingresso. Come dimensionare tale partitore? Ciò che si vuole ottenere sono 3 V in presenza del picco della tensione di rete, ma per sfruttare al meglio la linearità del moltiplicatore è conveniente impostare una tensione inferiore ai 3 V in presenza del picco d'ingresso. E' perciò stato scelto un valore di $4.5\text{ M}\Omega$ per la parte alta (costituita dalla serie di 3 resistenze da $1.5\text{ M}\Omega$) mentre per la parte bassa è stata scelta una resistenza $R_{57}=27\text{ K}\Omega$ (con in parallelo un condensatore da 2.2 nF per eliminare possibili disturbi EMI). In questa maniera alimentando il ballast con una sinusoide a 230V efficaci, si ha un valore di picco pari a 325V e questo produce una tensione di 1.94 V su questo piedino.

Vi è inoltre implementata la protezione da "AC-main brownout": qualora la tensione in ingresso scendesse sotto i 0.8 V (corrispondenti a 134 V con i valori di resistenza scelti), il MOS non viene attivato fino a quando la tensione non supera i 0.88V (corrispondenti a 148 V).

PIN 4: CS

Attraverso questo pin il chip riceve informazioni sul livello di corrente istantanea che scorre sull'induttore di boost grazie all'analisi della tensione sul resistore di sensing (opportunamente filtrata da una rete RC). Appena tale tensione supera la soglia V_{CS} viene resettato il flip-flop la cui uscita comanda lo spegnimento del MOS. Questo stato viene mantenuto fin tanto che non arriva il comando di *set* del flip-flop, generato mediante la rete di *Zero Current Detection*. Per evitare indesiderati spegnimenti, è implementato anche un blanking time per non considerare i picchi di corrente di recovery del diodo che si spegne all'accensione del transistor. Il resistore di sensing viene dimensionato considerando il valore massimo che può assumere la V_{CS} che è pari a 1 V^6 e considerando che la corrente massima che ci si aspetta di moni-

⁶quella minima garantita da datasheet

torare è di 4 A. Inoltre è presente un ulteriore comparatore con soglia a 1.7 V per proteggersi da eventuali saturazioni del magnetico dell'induttore che porterebbero ad avere valori di corrente pericolosamente alte (la saturazione del magnetico non è un evento così raro come potrebbe attendersi e ci sono situazioni, come allo startup o con carichi estremamente variabili dove si va a lavorare in prossimità della saturazione).

In base a ciò si è scelto un resistore di sensing di 0.27Ω , ottenuto dal parallelo di due resistenze: $R61=0.47 \Omega$ e $R62=0.68 \Omega$.

PIN 5: VFF

Il controller L6564 è pensato per poter svolgere il suo lavoro di correttore di Power Factor per tensioni che variano dai 110 a 220 V. In figura 3.6 viene mostrata la risposta del moltiplicatore a differenti valori di tensioni in ingresso: per bassi valori ($V_{FF}=1$ V) per attingere la stessa potenza sono necessari livelli di corrente più elevati e quindi il moltiplicatore risponde con riferimenti di corrente più elevati (vedi anche caso contrario con $V_{FF}=3$ V). Più lineare è la risposta del moltiplicatore, minore sarà la distorsione della corrente assorbita dalla rete.

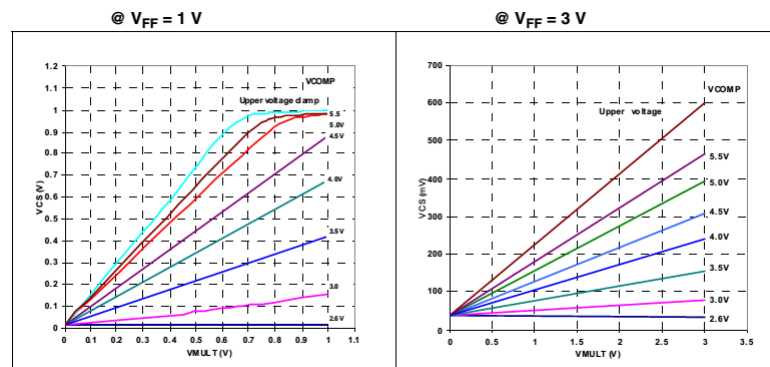


Figura 3.6: Caratteristica del moltiplicatore per due differenti valori di V_{FF} (da datasheet). [7]

Risulta perciò necessario avere informazioni sulla tensione d'ingresso. A ciò pensa questo pin che fa una rivelazione di picco della tensione di rete: internamente è presente un diodo rettificatore ideale (a bassa resistenza d'uscita e senza soglia) il cui anodo è connesso a $MULT$ e il catodo a V_{FF} . Esternamente una rete composta dal parallelo tra una resistenza e un con-

densatore aggiusta la costante di tempo per la rivelazione di picco: più elevata è la costante di tempo e migliore è il comportamento del PFC e viceversa esiste una costante di tempo minima per permetterne il regolare funzionamento. ST suggerisce di impostare i seguenti valori: $R_{59}=1\text{ M}\Omega$ e $C_{25}=1\text{ }\mu\text{F}$.

PIN 6: PFC_OK

Realizza un sensing aggiuntivo sulla tensione di uscita attraverso un partitore differente da quello collegato al pin *INV*. La funzione principale di tale piedino è quella di proteggere il circuito da sovratensioni in uscita (presenti principalmente allo starup). Il superamento della soglia interna di 2.5 V attiva il segnale *OVP* che sospende l'attivazione del MOS fino a quando l'uscita non scende al di sotto di 2.4 V. Per far ciò si è scelto il valore di 430 V come valore massimo di tensione sull'uscita che implica la scelta di una serie di 2 resistenze da 3.3 M Ω (R8 e R25) e una da 2.2 M Ω (R32) per la parte superiore del partitore, mentre per la parte inferiore si è scelta $R_{49}=51\text{ K}\Omega$. Qualora la sovratensione sia riconosciuta da questo partitore, senza che sia riconosciuta dal piedino *INV* (che presenta una dinamica più lenta), si genera il segnale *L_OVP* avvertendo che uno dei due sensing dell'uscita ha un problema.

Un ulteriore comparatore con isteresi (0.23 V - 0.27 V) permette di utilizzare questo piedino come piedino di *Enable* del chip, implementato tramite rete di pull-down ad opera di un BJT (attivando il MOS, sul piedino giunge una tensione nulla che disabilita il chip).

PIN 7: ZCD

A questo pin viene dato il compito di rilevare l'istante in cui si deve accendere il MOS, cioè non appena la corrente sul magnetico si annulla, quindi va a fare un sensing della tensione per verificare l'istante in cui non è più clampata dal diodo in conduzione. Per far ciò è implementato l'ennesimo comparatore ad isteresi con soglia inferiore a 0.7 V (scesi al di sotto della quale, viene attivato il *set* del flip-flop e il MOS viene acceso). Prima che un ulteriore trigger possa avvenire, è necessario che la tensione superi la soglia di 1.4 V. Si collega perciò tale pin al secondario dell'induttanza di boost con resistore da 68 K Ω (R38) per limitarne la corrente in ingresso al pin che da datasheet

è 2.5 mA. Quando il MOS è acceso e in ingresso vi è il picco di tensione (325 V) al secondario dell'induttanza di boost vi sono 32.5 V⁷ che divisi per la resistenza danno la corrente di 0.48 mA (<2.5 mA). Il caso peggiore a MOS spento si ha quando sul DC-link ci sono 430 V, che riflessi del rapporto spire diventano 43 V e quindi portano una corrente di 0.63 mA (<2.5 mA).

PIN 8: GND

Riferimento di massa, sia per i pin di segnale, sia per quello di comando del MOS di potenza.

PIN 9: GD

Pin di pilotaggio del MOS: all'accensione è in grado di erogare 600 mA, mentre allo spegnimento di assorbirne fino a 800 mA. La tensione massima di accensione è di 12 V. L'eventuale sottotensione di rete, mantiene tale pin ad un potenziale di 1.1 V attraverso una rete di pull-down che mantiene il MOS spento. Un resistore da 27 Ω (R54) serve per evitare ringing sul gate.

PIN 10: VCC

La rete per portare la giusta tensione di alimentazione del chip (che deve eccedere i 12 V per poterlo accendere) è suggerita in numerosi esempi nelle *Application Note*. La serie di 3 resistori da 120 KΩ fa sì che in presenza del picco della tensione di rete (325 V), entra in funzione il clamper da 25 V dell'integrato e inizia a circolare una corrente da meno di 1 mA sul piedino, restando ampiamente dentro i limiti imposti di 20 mA. A chip attivato, l'alimentazione viene fornita dalla seconda rete di alimentazione che prende la tensione dal secondario del trasformatore di boost e prevede l'utilizzo di una capacità C10=4.7 nF, un resistore R27=100 Ω, il diodo zener D11 da 18 V e 45 mA massimi ed infine dal diodo rettificatore D13 e il condensatore C15=47 μF. Un ulteriore condensatore C19=100 nF collegato il più possibile vicino all'integrato serve da by-pass per gli spike di tensione.

⁷in quanto si è scelto un trasformatore come induttanza di boost che ha rapporto spire 10:1.

3 Dimensionamento Full-bridge

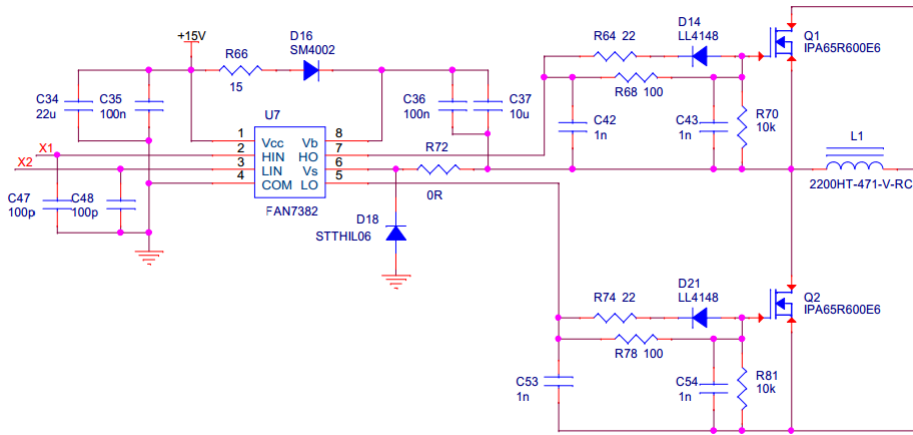


Figura 3.7: Schematico del leg sinistro del full-bridge con l'integrato FAN7382 di pilotaggio dei MOS. [Per far più chiarezza è stato riportato solamente il leg sinistro; quello destro risulta del tutto analogo]

I DISPOSITIVI ATTIVI DEL PONTE

I MOS utilizzati nella realizzazione dello stadio full-bridge sono gli stessi utilizzati nella realizzazione del boost PFC: Q1, Q2, Q3, Q4 sono IPA65R600E6 CoolMOS della *INFINEON*.

La lampada deve erogare 150 W, ma come da specifiche è bene prendersi un discreto margine e ipotizzarne al più 200 W. Se la lampada opera a 100 V, questo vuol dire che sul ponte circola una corrente di 1.5 A (2 A al massimo della potenza erogabile), ma i valori riportati sono da intendersi come i valori medi dell'onda triangolare di corrente che circola sugli induttori. Nell'ipotesi di funzionamento *continuo*⁸ il ripple picco picco HF è impostato per essere al massimo di 3 A, perciò il valor massimo di corrente che circola sull'induttore è 2 (valor medio) + 1.5 (valore di picco del ripple) = 3.5 A. Il caso però non è ancora il peggiore stimabile: appena vi è l'innesco dell'arco la tensione è molto bassa, ad esempio 30 V; per ottenere 150 W la corrente deve valere circa 4.5 A e sommando i 1.5 A di ripple si ottengono 6 A.

⁸il funzionamento discontinuo - dove la corrente assume valori nulli - vero e proprio non può mai verificarsi per via del fatto che l'implementazione full-bridge non lo permette; in questo caso per "funzionamento continuo" si intende la situazione in cui la corrente all'interno di un periodo HF resta sempre positiva o sempre negativa

I dispositivi MOS scelti hanno una portata di corrente continua di 4.6 A mentre il diodo di free-wheeling è in grado di resistere a valori fino a 6.3 A.

Per questi componenti è bene fare un'analisi anche dei contributi dissipativi di potenza che portano un incremento della temperatura. I MOS, che si presentano nella versione TO220 Full-Pack hanno una resistenza termica senza dissipatore di 80 °C/W, un valore decisamente alto. Utilizzando un dissipatore tale valore può scendere fino a 20 °C/W. Trascurando per ora il contributo portato dalla dissipazione per commutazione e considerando solo la dissipazione portata dalla resistenza di conduzione di 1.4 Ω attraversata dalla corrente efficace di 2.5 A (come detto in precedenza il valore può essere anche superiore) si ottiene un contributo di potenza da dissipare di 10 W. Con il dissipatore si ha perciò un innalzamento di temperatura di 200 °C e questo suggerisce l'utilizzo di una ventola di raffreddamento per diminuire ancor di più la resistenza termica del dissipatore. Benché i dati riportati siano abbastanza allarmanti la questione termica è un aspetto da prendere in considerazione successivamente. Una volta stimate bene le perdite occorrerà studiare bene il dissipatore (si prevede di ridurre la resistenza termica usando un altro package). Un ulteriore miglioramento potrà venire se si riesce a trovare la frequenza di commutazione ottimale che minimizzi le perdite complessive (conduzione e commutazione).

3.1 L'integrato di gate driver: FAN7382

Come integrato di pilotaggio dei MOS è stato utilizzato il FAN7382 della *FAIRCHILD*: è un integrato in grado di comandare un half-bridge, per questo motivo nel circuito ne sono implementati due, uno per il leg sinistro (in figura 3.8) e uno per il leg destro. Per questo integrato la corrente erogata è di 350 mA mentre quella assorbita è di 650 mA, mentre al pin di bootstrap (V_b) possono arrivare fino a 600 V.

Al suo interno l'integrato dispone di una tecnica in grado di cancellare il rumore di modo comune sugli ingressi, oltre che un circuito di *UVLO* implementato su entrambi gli ingressi in grado di prevenire malfunzionamenti in caso V_{CC} o V_{BS} scenda al di sotto di valori insufficienti (8.2 V) al corretto pilotaggio. Le protezioni garantiscono un'accensione dei MOS in modo efficace e senza tentennamenti o ringing che potrebbero portare anomalie.

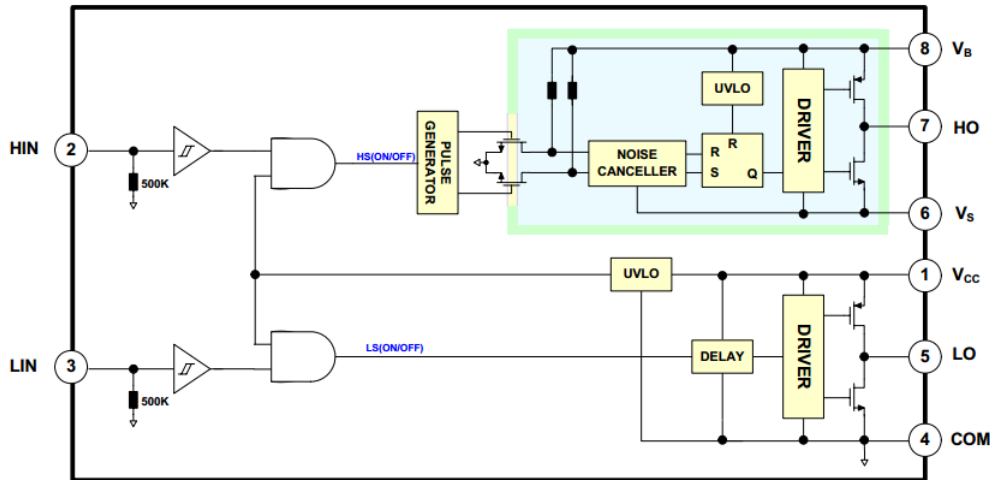


Figura 3.8: Diagramma a blocchi interno dell'integrato di gate driver FAN7382.[8]

PIN 1: VCC

L'alimentazione che l'integrato supporta è di 15-18 V. Nel progetto è stata utilizzata la tensione di 15 V in uscita dal flyback converter. Nel datasheet del dispositivo consigliano di utilizzare due condensatori tra VCC e COM, rispettivamente di valore $C34=22 \mu\text{F}$ e $C35=100 \mu\text{F}$ (a basso ESR) come riserva di energia vicino all'integrato per eliminare eventuali disturbi sull'alimentazione. Inoltre, sempre al piedino di alimentazione è collegata la rete di carica del condensatore di bootstrap.

PIN 2 e PIN 3: HIN e LIN

A questi due piedini sono collegati gli ingressi di pilotaggio dei MOS in uscita dalla rete di condizionamento dei segnali del microcontrollore. Esternamente ogni ingresso ha un condensatore di bypass da 100 pF per la riduzione del rumore. Internamente invece presentano un trigger a soglia per determinare il passaggio alto-basso (o viceversa) degli ingressi. Successivamente vengono messi a "AND" col segnale in uscita dal *Under Voltage Lock Out* che forza i segnali ad essere bassi qualora si presentasse un calo eccessivo di tensione d'alimentazione.

PIN 4: COM

É il piedino al quale collegare la massa dell'integrato, utile al pilotaggio del MOS inferiore. L'accensione del MOS inferiore passa per il valore presente a questo piedino

PIN 5: LO

Piedino di accensione del MOS inferiore, viene collegato al gate del MOS attraverso una rete di taratura. I MOS utilizzati hanno una soglia di accensione di 3 V, ma più alta è la tensione che viene portata al gate minore sarà la resistenza del canale e di conseguenza minori saranno le perdite per conduzione. Il FAN7382 per accendere il MOS inferiore utilizza una rete di push-pull che fa arrivare la tensione VCC al piedino.

La rete di taratura è stata presa dalle demo-board di *STMicroelectronics*. Fra gate e source viene posto un resistore R81 da 10 K Ω , anche se nell'Application Note 6076 è espressamente detto che si può fare a meno. Un condensatore da C53=1 nF viene utilizzato per stabilizzare come al solito l'uscita. Vengono utilizzate due resistenze di gate, una per l'accensione (R78) e una per lo spegnimento del MOS (R74). Il valore viene calcolato noti i seguenti parametri del MOS utilizzato: $Q_{gs} = 2.75$ nC, $Q_{gd} = 12$ nC, $V_{GSon} = 3$ V. La corrente erogata dal driver è di 350 mA ed il tempo di accensione assunto è di 200 ns.

$$I_g = \frac{Q_{gs} + Q_{gd}}{t_{sw}} = \frac{2.75 + 12}{200} = 73.75 \text{ mA} \quad (3.26)$$

$$R_{TOTAL} = \frac{V_{CC} - V_{GSon}}{I_g} = \frac{15 - 3}{73.75 \cdot 10^{-3}} = 163 \Omega \quad (3.27)$$

$$R_{DV_{Son}} = \frac{15}{350 \cdot 10^{-3}} = 43 \Omega \quad (3.28)$$

$$R_g = R_{TOTAL} - R_{DV_{Son}} = 163 - 43 = 120 \Omega \quad (3.29)$$

Viene scelta una resistenza da 100 Ω , anche se il datasheet del MOS consiglia un valore di 6.8 Ω con l'obiettivo di avere tempi di commutazione di 10

ns. Per quanto riguarda invece la resistenza di spegnimento del MOS (posta in serie al diodo D21 per condurre le correnti uscenti dal gate del MOS) si è scelto il valore di 22Ω . Entrambi i valori delle resistenze incidono sulle perdite di commutazione che come detto in precedenza devono eguagliare quelle di conduzione.

PIN 6: V_s

Su questo piedino vi è la tensione d'uscita del ponte V_{OUT} , nonché tensione di riferimento per l'accensione del MOS superiore. L'introduzione del diodo di clamping D18 è dovuta alla volontà di impedire che il nodo V_s possa scendere troppo al di sotto della tensione di massa di potenza a salvaguardia dell'integrato che potrebbe esser danneggiato. La resistenza R72 viene consigliata nell'Application Note 6067 e il valore suggerito è di qualche Ω . Questa resistenza viene utilizzata per la carica del condensatore di bootstrap. Nel progetto è stata utilizzata una resistenza a 0Ω .

PIN 7: HO

La rete esterna di pilotaggio del gate del MOS superiore è analoga a quella utilizzata per il MOS inferiore. Il gate driver dispone di una rete di cancellazione di rumore e un flip-flop SR per pulire il segnale di pilotaggio essendo quest'ultimo generato attraverso la rete di bootstrap.

PIN 8: V_b

A questo piedino è collegata la rete di bootstrap analizzata nel paragrafo 3.2 del capitolo 2.

Il diodo di bootstrap D16 inizialmente presente nel progetto (mostrato in figura 3.8) è un SM4002: questo componente però è risultato sottodimensionato in quanto facendo funzionare il ponte in discontinuo, sul carico vi sono all'incirca 400 V e tale tensione risulta troppo elevata per questo diodo che presenta una *tensione di breakdown* di 100 V. La scelta perciò è ricaduta su diodi SM4007 che presenta una tensione di breakdown di 1000 V.

La capacità di bootstrap è realizzata mediante il parallelo di due condensatori uguali a quelli utilizzati tra VCC e COM. L'unica specifica che devono rispettare questi condensatori è che devono essere almeno di 2 ordini di gran-

dezza più grandi della capacità d'ingresso del MOS ($C_{ISS} = 440 \text{ pF}$). Infine il resistore R66 da 15Ω viene suggerito per limitare le correnti di picco.

4 Filtro d'uscita e circuito d'innescò

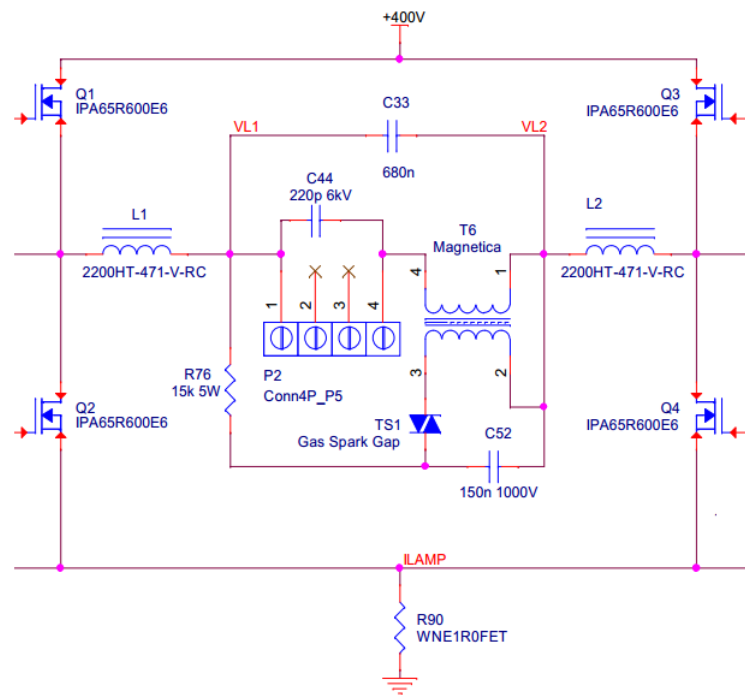


Figura 3.9: Schematico del filtro d'uscita del ponte e del circuito d'innescò. Al connettore a 4 porte è collegata la lampada a scarica HID.

Per il dimensionamento dell'induttore del filtro ci si è affidati ad una semplice analisi delle tensioni e correnti su questo componente durante il normale esercizio. Ciò che si vuol ottenere è un funzionamento continuo del ponte durante il periodo HF di energizzazione della lampada. Nell'ipotesi di avere una lampada nuova che a regime presenta una tensione di lavoro di 100 V , questo fa sì che per ottenere i 150 W di potenza sulla lampada circolino 1.5 A medi. Il caso limite di funzionamento continuo impone che la corrente parta da 0 e arrivi al valore di 3 A durante il periodo di energizzazione, ottenendo così 1.5 A medi. La tensione sull'induttore è il risultato della differenza tra la tensione del DC-Link (400 V) e la tensione di lampada (100 V). Infine

il tempo T_{ON} è facilmente deducibile come la porzione di tempo necessaria per portare 100 V sulla lampada dei 400 V disponibili, cioè $\frac{1}{4}$ del periodo HF T_{LAMP} . Come da specifiche la frequenza di commutazione sui leg è di 24 KHz che porta ad una frequenza di commutazione su lampada di 48 KHz. Il periodo T_{LAMP} perciò ottenuto è di 20.83 μs mentre T_{ON} è 5.2 μs .

$$v_L = L \cdot \frac{di_L}{dt} \implies 300 = L \cdot \frac{3 - 0}{5.2 \cdot 10^{-6}} \implies L = 520 \mu H \quad (3.30)$$

Questa situazione non è la peggiore preventivabile. Infatti in caso di utilizzo di una lampada nuova o appena accesa, ci si aspetta di lavorare con una tensione di lampada inferiore a quella precedente di 100 V e questo porta ad una corrente media maggiore. Infine un parametro molto importante che influenza considerevolmente i conti svolti è la saturazione del magnetico dell'induttore: in seguito a correnti medie troppo elevate l'induttore non presenta il valore di induttanza che ci si aspetta, ma bensì ridimensionato (può addirittura arrivare ad essere la metà) e la situazione di saturazione del magnetico è più probabile di quanto si possa immaginare. Perciò si è optato per un valore di induttanza di 940 μH .

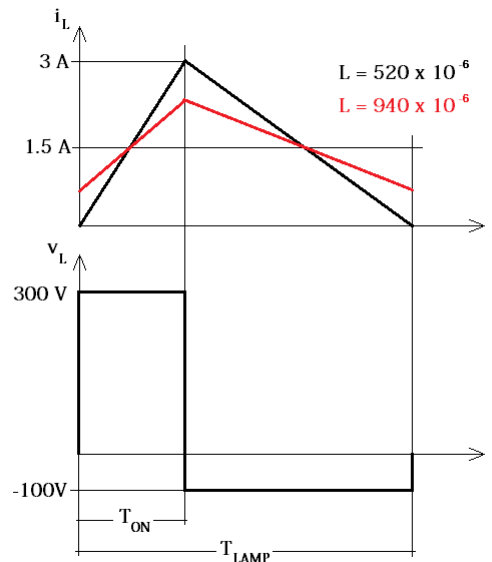


Figura 3.10: Corrente i_L e tensione v_L sull'induttore del filtro. Nello specifico viene raffigurata la corrente nel caso di induttore da 540 μH (nero) e nel caso di 940 μH (rosso).

Un'induttanza da 940 μH potrebbe allarmare in termini di ingombro (una prerogativa fondamentale del ballast elettronico è quello di avere un induttore

per limitare le correnti più piccolo rispetto a quello del ballast elettromagnetico). Perciò è stato scelto di utilizzare due induttori di valore da $470 \mu\text{H}$ posti in serie al carico e rispettivamente uno su ciascuno dei leg per ottenere dunque la perfetta simmetria fra i leg. In questo modo entrambi i morsetti della lampada sono ad alta impedenza per le alte frequenze e questo porta benefici dal punto di vista EMI in quanto il corpo della lampada può essere facilmente un veicolo di interferenza verso l'esterno.

Sulla base di quanto esposto fra tutti i magnetici disponibili (vi è una vasta e variegata offerta sul mercato) si è scelto per due bobine 2200HT-471-V-RC prodotte dalla *BOURNS*: hanno il vantaggio di poter lavorare in un range esteso di temperature (fino a $200 \text{ }^\circ\text{C}$) e possono condurre una corrente DC fino a 2.8 A (portando un incremento di temperatura di $50 \text{ }^\circ\text{C}$). Qualunque sia la famiglia di induttanze individuate esiste il seguente compromesso: più è elevato è il valore della corrente di saturazione (DC) e più piccolo è il valore di induttanza.

Il condensatore di carico in parallelo alla lampada è stato scelto di classe X2 e con un rating di $305 V_{ac}$: il B32924C3684K00 della *EPCOS* del valore di 680 nF

Il circuito d'innescò è costituito dal trasformatore d'impulsi per il quale si è scelto un Ignition Transformer 150 W della *MAGNETICA* con un rapporto di trasformazione di 16.6 a 10 KHz ed un valore di $55.6 \mu\text{H}$ al secondario e di 260 nF al primario, mentre il valore di resistenza serie presente al secondario (quella al primario quasi trascurabile e di poco interesse nel funzionamento del circuito d'innescò) è di un centinaio di $\text{m}\Omega$.

Il dispositivo di scarica è lo Spark Gap Surge Arrester EC350X della *EPCOS* che ha una tensione di scarica di 350 V .

Resta da determinare il valore di R76, C52 e C44. Caratteristica fondamentale di questi 3 componenti è la possibilità di resistere ad alte tensioni, in particolare alla capacità C44 posta in parallelo alla lampada e quindi anch'essa sottoposta a migliaia di Volt durante l'innescò dell'arco. Per questo è stata scelta la capacità da 220 pF in grado di resistere a 6 KV . Il circuito RC composto da R76 e C52 caricandosi manda in conduzione lo Spark Gap che come detto ha soglia di conduzione a 350 V . Scegliendo $R76=15 \text{ K}\Omega$ e $C52=150 \text{ nF}$ si ottiene una costante di tempo $\tau=2.25 \text{ ms}$. In via del tutto

approssimativa si può determinare il tempo che intercorre tra due scariche d'arco, corrispondente al tempo di carica del condensatore C38 fino al valore di 350:

$$V = E \cdot (1 - e^{-\frac{t}{\tau}}) \Rightarrow 350 = 400 \cdot (1 - e^{-\frac{t}{2.25 \cdot 10^{-3}}}) \Rightarrow t = 4.57ms \quad (3.31)$$

Questo risultato è visibile nella simulazione riprodotta con LTspice riportata in figura 2.14 del capitolo 3: la prima carica del condensatore fino al valore di 350 V è maggiore di 4.57 ms, ma questo è dovuto al fatto che durante la prima carica la tensione ai capi del condensatore C33 non è possibile considerarla costante a 400 V come invece è stato fatto nell'equazione 3.31.

5 Dimensionamento reti di sensing

5.1 Reti di sensing di tensione

Nel circuito sono presenti 3 reti di sensing per le tensioni d'interesse: la prima e la seconda sono due semplici partitori resistivi posti rispettivamente prima e dopo il circuito PFC per leggere il valore della tensione di rete (V_{RETE}) e quella sul DC-Link (V_{BUS}). La terza rete di sensing invece è quella in grado di restituire il valore della tensione di lampada (V_{LAMP}) e per questa non si può implementare un semplice partitore resistivo. Le 3 tensioni così prodotte si devono interfacciare al convertitore A/D presente all'interno del microcontrollore che, convertendole, le legge e le analizza. Le tensioni in ingresso al microcontrollore possono assumere valori compresi nella gamma tra 0 e 3.3 V (tensione d'alimentazione del microcontrollore), perciò le reti vanno dimensionate di conseguenza.

TENSIONI V_{RETE} e V_{BUS}

Con riferimento alla figura 3.2 sono stati calcolati i seguenti valori di resistenze⁹: come da specifiche, la tensione di rete può assumere un valore efficace massimo di 264 V che corrisponde ad un valore di picco di 373 V (per precauzione si assume il valore di 400 V), perciò il dimensionamento della prima

⁹A differenza di quanto presente in precedenza dove l'utilizzo del PIC al posto del microcontrollore dava la disponibilità di avere tensioni comprese nella gamma tra 0 e 5 V

rete resistiva può essere fatto associando a questa tensione il valore di 3.3 V. Nella realtà però tale tensione viene utilizzata anche dall'integrato L6564 nel pin *MULT* come spiegato nel paragrafo 2.2, perciò il guadagno della prima rete resistiva è vincolato essere uguale a $G_{rete} = 5.96 \cdot 10^{-3}$, ottenuto mediante il partitore tra $R_{high} = 4.5 \text{ M}\Omega$ e $R_{low} = 27 \text{ K}\Omega$.

Il guadagno del partitore per determinare V_{BUS} è definito in maniera analoga, con la differenza che si è impostato il valore massimo raggiungibile sul DC-Link di 500 V (prendendo così un buon margine, come nel caso precedente, per non far lavorare in convertitore in prossimità del fondo scala).

$$V_{BUS} = V_{DC-Link.max} \cdot G_{bus} \Rightarrow G_{bus} = 6.6 \cdot 10^{-3} \quad (3.32)$$

Scelto il valore di $R_{high} = 990 \text{ K}\Omega$ (serie di 3 resistenze da 330 K Ω) si ottiene $R_{low} = 6577\Omega$. La resistenza appartenente allo standard E12 scelta è da 5.6 K Ω (a differenza della precedente resistenza da 10 K Ω prevista per l'interfacciamento del partitore con un PIC alimentato a 5 V) che porta ad avere un guadagno del partitore pari a $G_{bus} = 5.62 \cdot 10^{-3}$.

TENSIONE V_{LAMP}

In questo caso la rete di sensing non può essere un semplice partitore: vi è la necessità di leggere tensioni differenziali (VL1 - VL2) che possono assumere valori sia positivi (energizzazione positiva) sia negativi (energizzazione negativa), restituendo sempre valori di tensioni compresi tra 0 e 3.3 V che così posso essere processati dal microcontrollore. La rete implementata è mostrata in figura 3.11 dove sono stati divisi i vari sotto-blocchi ed è stato assegnato il nome alle tensioni in ingresso e in uscita da ogni stadio. Il primo stadio deve riportare l'escursione di VL1 e VL2 (0-400 V) a valori compresi tra 0 e 15 V (livelli di tensione ammessi in ingresso all'integrato TS274/SO).

$$V_x = G_1 \cdot VL1 \Rightarrow G_1 = \frac{V_x}{VL1} = \frac{15}{400} = 0.0375 \quad (3.33)$$

Scelta la R_{high} come la serie di due resistenze (R86 e R87 nel caso di VL1) da 510 K Ω si ottiene $R_{low} = 36867\Omega$. Viene dunque scelto un resistore da 39 K Ω (R88) in parallelo ad un condensatore (C58) da 1 nF per stabilizzare il segnale, ottenendo $G_1 = 0.0368$.

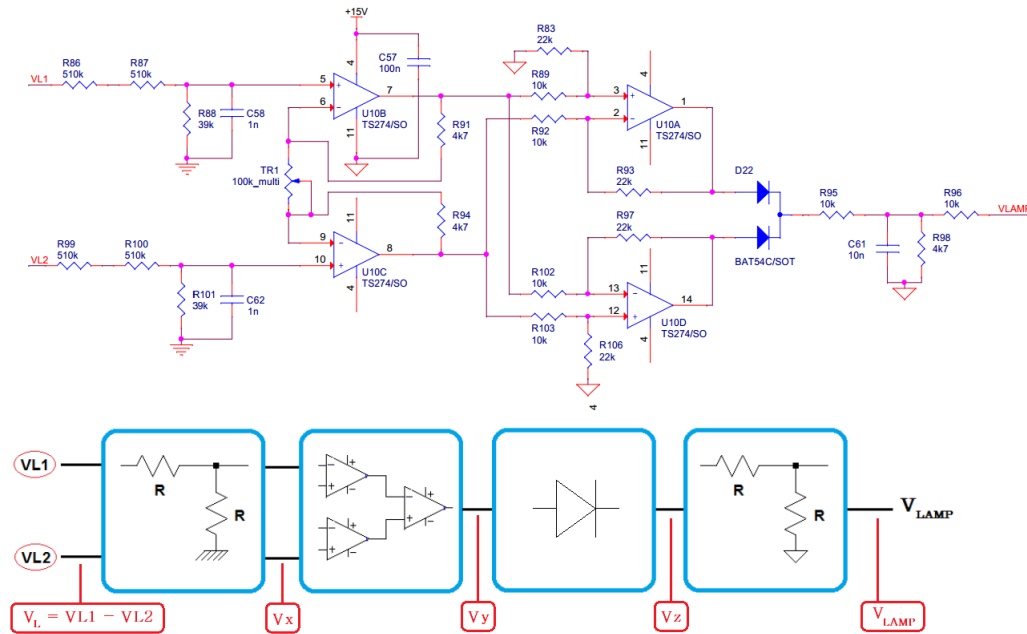


Figura 3.11: Schematico del circuito di sensing della tensione di lampada. In basso uno schema con riportate le tensioni d'interesse ai vari nodi.

Segue l'amplificatore differenziale per strumentazione. Il guadagno di questo stadio viene impostato facendo la seguente considerazione: a lampada accesa la tensione ai suoi capi non supererà mai valori di 150 V.¹⁰ Risulterà perciò inutile processare valori superiori. Il guadagno di questo stadio viene perciò dimensionato in modo tale che in presenza di $V_L=158$ V¹¹ (cioè $V_x=5.82$ V) ci sia $V_y=15$ V (massimo tensione d'uscita dall'operazionale). In questa maniera verrà aumentata la precisione nel range tra 0 e 150 V (gamma di tensioni d'interesse). Dunque il guadagno deve valere $G_{diff}=2.577$. Il guadagno dell'amplificatore differenziale è:

$$G_{diff} = \left(1 + \frac{2 \cdot R1}{R_G}\right) \cdot \left(\frac{R3}{R2}\right) \quad (3.34)$$

Avendo scelto il valore di $R1=4.7$ K Ω , $R3=22$ K Ω e $R2=10$ K Ω si ottiene il

¹⁰La tensione $V_{LAMP} = VL1 - VL2$ avrà valori superiori solo a lampada spenta quando può essere considerata un circuito aperto.

¹¹vengono presi 8 V di margine per non lavorare con il convertitore a fondo scala

valore che deve assumere il trimmer:

$$R_G = \frac{2 \cdot R1}{G_{diff} \left(\frac{R2}{R3} \right) - 1} = \frac{2 \cdot 4700}{2.04 \left(\frac{10000}{12000} \right) - 1} = 54854\Omega \quad (3.35)$$

In questo caso non vi è la necessità di arrotondare il risultato ottenuto. Trattandosi di un trimmer il valore corretto viene impostato andando ad effettuare la lettura tramite multimetro di V_y imponendo una tensione V_x nota a priori. Così è stato svolto: ad una tensione di lampada di 100 V corrisponde $V_x=3.68$ V; si è regolato il valore del trimmer affinché in uscita fosse presente una tensione di 9.48 V.

Questo stadio è stato duplicato ma con ingressi invertiti in modo tale che la rete di sensing restituisca valori positivi anche in caso di tensioni di lampada negative. Le due uscite sono messe ad *OR* mediante l'utilizzo di 2 diodi; la tensione V_z a valle dei diodi risulterà sempre maggiore o uguale a 0. Si risolve così il problema di voler leggere V_{LAMP} sempre positiva anche in presenza di energizzazione negativa. I diodi utilizzati sono presenti in un unico componente: il BAT54C ha al suo interno due diodi Schottky che presentano una caduta di 320 mV quando percorsi da una corrente di 1 mA.

L'ultimo stadio, un partitore resistivo composto da R95 e R98 (con in più il solito condensatore di livellamento) porta il range di tensioni di V_z - che arriva al più a 14.68 essendoci la caduta di tensione sul diodo - al range utile al microcontrollore (0-3.3 V):

$$V_{LAMP} = V_z \cdot G_2 \implies G_2 = \frac{3.3}{14.68} = 0.225 \quad (3.36)$$

Utilizzando una R_{high} di 12 K Ω si ottiene un valore di R_{low} di 3483 Ω . Scelta R98 da 3.3 K Ω , lo stadio finale ha guadagno di $G_2 = 0.215$. Analogamente a quanto detto prima, anche queste ultime due resistenze sono state modificate rispetto a quanto mostrato in figura 3.11. In ultimo, la resistenza R96 è stata cortocircuitata in quanto sulla base di numerose prove, ci si è resi conto che tale resistenza in qualche modo infastidiva il convertitore analogico/digitale che al suo interno ha un circuito di *sample and hold*.

Viene indicato con G_{TOT} il *guadagno totale* della rete di sensing cioè, ad esempio, il guadagno che la rete presenta per ottenere una V_{LAMP} di 1.970 V a fronte di una tensione ai capi della lampada di 100 V (come da tabella 3.1). Due fattori rendono non lineare l'andamento di questo parametro: la presenza

dei diodi fa sì che per tensioni di lampada troppo basse la caduta di tensione su di essi sia più fastidiosa; in secondo luogo c'è l'effetto di "saturazione" che si ha per tensioni di lampada superiori ai 158 V, ma ciò non deve allarmare perché questo effetto è stato volutamente imposto.

V_L	G_1	V_x	G_{diff}	V_y	V_D	V_z	G_2	V_{LAMP}	G_{TOT}
10	0.0368	0.368	2.577	0.948	0.32	0.628	0.215	0.135	0.0135
50	0.0368	1.84	2.577	4.741	0.32	4.421	0.215	0.950	0.0190
100	0.0368	3.68	2.577	9.483	0.32	9.163	0.215	1.970	0.0197
150	0.0368	5.52	2.577	14.225	0.32	13.905	0.215	2.989	0.0199
200	0.0368	7.36	2.577	15	0.32	14.68	0.215	3.156	0.0157
250	0.0368	9.2	2.577	15	0.32	14.68	0.215	3.156	0.0126
300	0.0368	11.04	2.577	15	0.32	14.68	0.215	3.156	0.0105

Tabella 3.1: Con riferimento all'immagine 3.11 sono elencati i valori intermedi di tensione e i guadagni in tensione offerti dai singoli stadi.

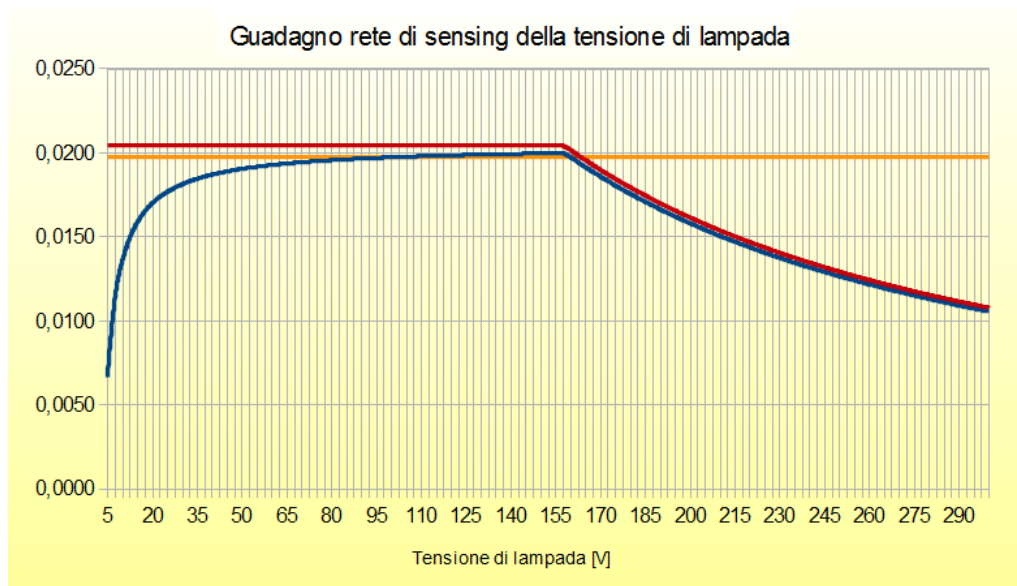


Figura 3.12: Andamento del guadagno della rete di sensing della tensione di lampada. In blu l'effettivo guadagno G_{TOT} , in rosso il guadagno della rete qualora venisse trascurata la caduta sui diodi, in giallo il valore di guadagno della rete impostato nel firmware.

Essendo di interesse solo tensioni di lampada comprese tra i 50 e 150 V (all'incirca), nel *firmware* è stato impostato un valore di $G_{TOT} = 0.0197$ (linea

gialla di figura 3.12) in quanto risulta quello che minimizza l'errore che il microcontrollore compie quando da V_{LAMP} (letta e convertita dal convertitore A/D) deve risalire alla reale tensione presente ai capi della lampada.

5.2 Reti di sensing di corrente

Il sensing della corrente di lampada viene effettuato mediante il resistore R90 posto tra source dei MOS inferiori del ponte e massa (figura 3.9). Prerogativa fondamentale di questo componente è la dissipazione di potenza oltre che un'induttanza parassita molto piccola ed una precisione la più grande possibile per ottenere così una conversione in tensione controllata. La scelta è ricaduta sul resistore WNE1R0FET della *OHMITE*: la resistenza è di $1\ \Omega$ con tolleranza del 1% e potenza dissipata di 5 W, mentre le lettere WN indicano che il componente non è induttivo.

Il segnale I_{LAMP} risulterà una tensione proporzionale (con coefficiente di proporzionalità unitario) alla corrente che circola sul ponte durante le fasi di energizzazione. Durante le fasi di ricircolo il resistore non è percorso da corrente perciò si leggerà I_{LAMP} nulla come mostrato in figura 3.13.

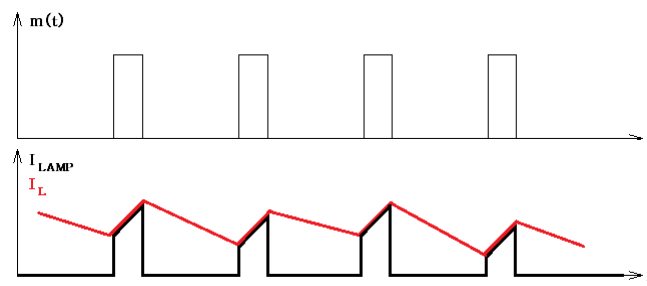


Figura 3.13: Esempio di andamento del segnale I_{LAMP} in nero, mentre in rosso è riportato l'andamento della corrente che effettivamente circola sul ponte ($m(t)$ è la legge di commutazione applicata alla lampada).

Il segnale così ottenuto viene mandato al microcontrollore e ai circuiti analogici asincroni di protezione che devono prontamente intervenire nel caso di superamento della soglia di protezione generando i segnali I_{OK} e I_{MAX} .

La sonda posta su I_{LAMP} ha evidenziato un segnale molto rumoroso e affetto da rumore e ciò ha suggerito di apporre su ogni ingresso che utilizza questo segnale una rete RC opportunamente dimensionata affinché venga in

parte ripulito. Allo stesso tempo la logica di protezione che genera I_{OK} deve poter intervenire tempestivamente e non è quindi consentito utilizzare una rete RC con un τ troppo grande che porterebbe ad una carica lenta del condensatore e ad una rilevazione della sovracorrente ritardata con conseguente latenza per lo spegnimento dei MOS.

L'ingresso positivo del comparatore I_{OK} proviene dal microcontrollore e quindi può assumere massimo il valore di 3.3 V e ciò vuol dire che il comparatore porterà bassa l'uscita tutte le volte che la corrente sul ponte sarà superiore di 3.3 A avendo impiegato un resistore di sensing da 1 Ω . Questo risulta molto limitativo in quanto la dinamica di corrente può assumere valori ben più alti a causa del ripple che presenta. Si è deciso perciò di introdurre un partitore resistivo sull'ingresso positivo del comparatore di I_{OK} e I_{MAX} e sul piedino del microcontrollore per la lettura di I_{LAMP} . In seguito a calcoli sul ripple di corrente al variare del duty cycle e della corrente media si è deciso di impostare il limite massimo di corrente sul ponte a 4.8 A: il partitore costituito da $R_{high} = 10 \text{ K}\Omega$ e $R_{low} = 22 \text{ K}\Omega$ ha un guadagno di 0.6875 ed è in grado di scalare la dinamica di I_{LAMP} all'interno del range tra 0 e 3.3 V. In figura 3.14-[A-B-D] non è presente la resistenza R_{low} in quanto aggiunta in un secondo momento da me.

Analizzando l'evaluation board proposta da *ST* si è scelto un condensatore da 100 pF (C70 e C83) per le reti RC in ingresso ai due comparatori, mentre per quanto riguarda l'ingresso al microcontrollore di I_{LAMP} si è scelto C30=100 nF che genera un polo a circa 230 Hz. In questo modo la rete RC, oltre che pulire in prima istanza il segnale da rumorosi spike, opera anche da filtro passa basso in grado di estrapolare il valore medio del segnale I_{LAMP} che ha una dinamica ad alta frequenza (il doppio della frequenza di commutazione HF su leg). Dividendo quanto letto dal microcontrollore per il duty cycle di lampada¹² si ottiene il valore della *corrente media di lampada*, utile a determinare la potenza erogata dalla lampada (una volta nota V_{LAMP}). Inizialmente i tre condensatori C70, C83 e C30 erano stati scelti tutti da 100 nF, ma questa scelta si è rivelata errata in quanto, in seguito ad alcune prove, ci si è accorti che tale valore era troppo elevato e bisognava attendere un paio di cicli di lampada affinché il condensatore si caricasse. Scalando di

¹²che ha valore compreso tra 0 e 1

un fattore 1000 solo i condensatori di I_{OK} e I_{MAX} si è risolto il problema.

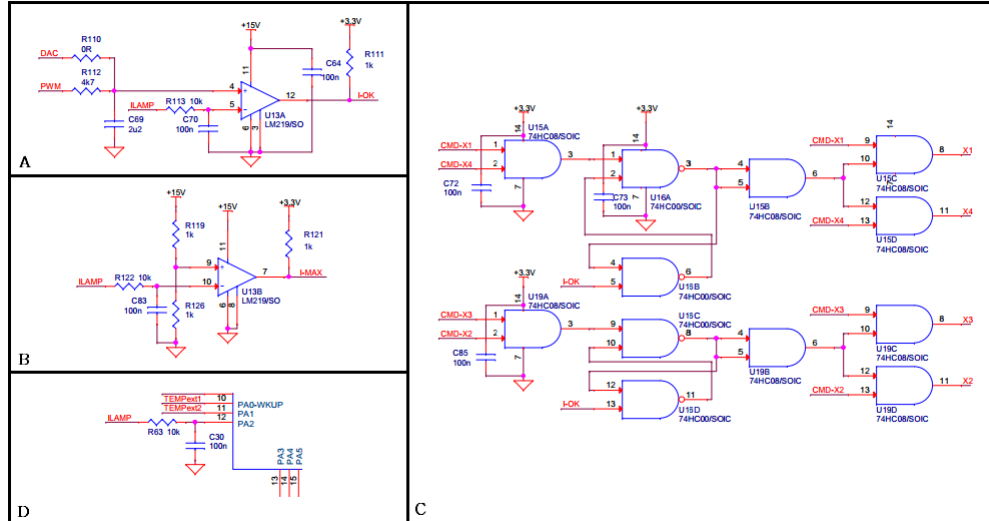


Figura 3.14: Schematico del sensing della corrente di lampada: A) anello di protezione che genera il segnale I_{OK} ; B) anello di protezione che genera il segnale I_{MAX} ; C) rete di condizionamento dei segnali di pilotaggio dei MOS mediante flip-flop realizzati a NAND; D) collegamento di I_{LAMP} ad un ingresso del microcontrollore mediante filtro RC.

Le logiche di protezione sono realizzate mediante i comparatori *high speed* dell'integrato LM219 che ha le uscite in open-collector; viene perciò utilizzata la resistenza di pull up R111 e R121 da 1 K Ω per avere le uscite a 3.3 V qualora la soglia non venisse superata. La soglia per il comparatore che genera I_{MAX} viene realizzata mediante un partitore resistivo con due resistenze da 1 K Ω , R119 e R126, che quindi realizzano una soglia di 7.5 V. Per realizzare la soglia del comparatore che genera I_{OK} si possono seguire due strade, entrambe richiedono l'utilizzo del microcontrollore: la prima grazie all'uscita del convertitore digitale-analogico all'interno del microcontrollore; la seconda mediante un segnale PWM filtrato con una rete RC che presenta un polo a 15 Hz in grado di restituirne solo il valor medio (proporzionale al T_{ON} del segnale PWM). Nel progetto del ballast realizzato si è utilizzata la seconda soluzione essendo il microcontrollore scelto privo di DAC: il segnale PWM adibito alla generazione della soglia ha frequenza di 24 KHz, abbondantemente oltre i 15 Hz.

Avendo la soglia generata dal microcontrollore si può implementare un *firmware* che modifica la soglia sulla base di algoritmi ben programmati che

gestiscono la fase di warm-up in limitazione di corrente e in regolazione di potenza. Questa possibile soluzione sarà implementata anche nel firmware realizzato in questo progetto.

In ultimo è stata aggiunta una rete di pull-down costituita da 4 resistenze di valore 100 K Ω in grado di mantenere bassi i segnali di pilotaggio dei MOS qualora il microcontrollore mettesse i piedini ad essi associati in alta impedenza, ad esempio in seguito ad una procedura di resettaggio (voluta o imprevista). Infatti ponendo i piedini in alta impedenza i segnali di pilotaggio risultano flottanti, ma vengono comunque elaborati dalla rete di flip-flop a NAND mostrato in figura 3.14-[C] che di tutta risposta pone tutte quattro le uscite ad un livello logico alto, attivando contemporaneamente tutti i MOS del ponte e provocando così un corto circuito tra 400 V e massa. L'aggiunta delle 4 resistenze di pull-down invece mantiene a massa i quattro segnali di pilotaggio dei MOS qualora i piedini fossero in alta impedenza (essendo state aggiunte in un secondo momento da me in seguito alle mie verifiche queste resistenze non compaiono in figura 3.14).

6 Dimensionamento flyback converter

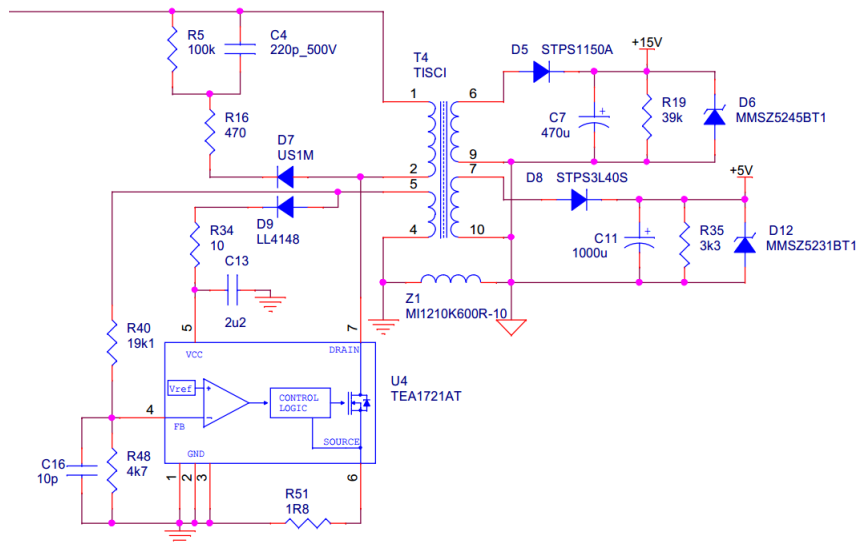


Figura 3.15: Schematico del flyback converter utilizzato nel progetto.

Il flyback converter presente nel progetto utilizza come ingresso la tensione presente sulla capacità del DC-Link (400 V) restituendo in uscita le tensioni continue di 5 V e di 15 V. Una rete di *snubber RCD* viene posta in parallelo al primario per limitare le sovratensioni sul dispositivo di controllo all'atto dell'apertura del MOS¹³: un diodo di ricircolo D7, le resistenze R5 e R16 e la capacità C4 sono state dimensionate seguendo i suggerimenti proposti nell'*application note* AN11060.

Al secondario del trasformatore un collegamento multipresa costituito da due avvolgimenti separati a diverso numero di spire permette di disaccoppiare i due circuiti per la generazione dei 5 V e 15 V: il diodo blocca la corrente nella fase di energizzazione del trasformatore, durante la quale la tensione sul carico d'uscita viene fornita dal condensatore elettrolitico. Infine il diodo Schottky funge da *clamber* per evitare il verificarsi di tensioni negative sulle uscite.

Infine viene utilizzata una ferrite MI1210K600R-10 della *Laird Technologies* per disaccoppiare la massa di potenza dalla massa di segnale¹⁴ questa precauzione risulta fondamentale in quanto in assenza della ferrite le reti di sensing restituirebbero valori affetti da molto rumore e potrebbero essere mal interpretati dal microcontrollore.

6.1 L'integrato del flyback converter: TEA1721AT

L'integrato TEA1721AT della *NXP*, utilizzato come controller, possiede al suo interno un MOSFET che opera come switch in grado di regolare l'energizzazione al primario del trasformatore isolato della *Tisci*. Diversi sono i *modi operativi* dell'integrato che, sulla base della tensione letta sul pin di *feedback* FB, stabilisce quale scegliere. I tre modi in ordine di decremento dell'impedenza d'uscita sono: *Constant Voltage Burst mode* (CVB), *Constant Voltage mode* (CV), *Constant Current mode* (CC).

Ad accensione avvenuta il controller lavora in modalità *Constant Voltage with Burst* nella quale la tensione di source è nell'intorno dei 120 mV, la frequenza di switching è costante a 22.5 KHz e la regolazione del duty cycle

¹³L'interruzione improvvisa della corrente su un carico induttivo genera sovratensioni ai capi del dispositivo che ne ha causato l'interruzione.

¹⁴La resistenza massima in corrente continua di questa ferrite è di 35 mΩ.

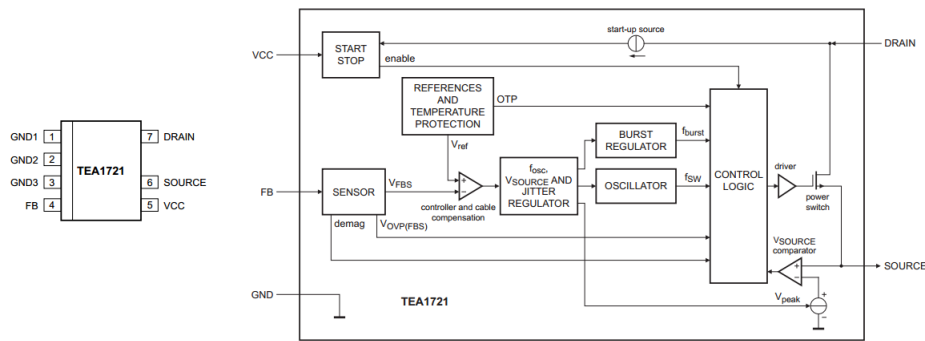


Figura 3.16: Piedinatura e diagramma a blocchi del controller TEA1721AT.[9]

del burst viene fatta sulla base di quanto letto dal pin FB. Dopo i primi due impulsi di corrente (ad ampiezza prestabilita I_{pk}) che circolano sul primario viene monitorata la tensione dal piedino FB in prossimità del secondo impulso. Se la tensione è superiore o al più uguale a 2.5 V non vengono generati altri impulsi di energizzazione del primario e il controller entra in modalità di risparmio energetico, mentre se la tensione è inferiore altri impulsi sono generati. Durante il periodo tra due impulsi è il condensatore d'uscita che deve gestire qualsiasi variazione di energia richiesta dal carico. Se l'uscita del flyback è a vuoto il *rate* con il quale si ripetono gli impulsi è basso¹⁵, mentre quando il carico d'uscita aumenta il numero di impulsi all'interno del burst aumenta per poter trasferire sufficiente energia. Quando l'intero periodo di burst viene riempito di impulsi a frequenza fissa di 22 KHz, il controller entra in modalità di Constant Voltage: l'integrato continua alla frequenza minima di 22.5 KHz con impulsi di corrente ad ampiezza fissa I_{pk} , ma se viene richiesta più energia dal carico la frequenza viene mantenuta e l'ampiezza dell'impulso aumentata. La tensione misurata sul source passa dai precedenti 120 mV a più di 500 mV. Il caso di lavoro a corrente costante non è stato preso in esame in quanto non di interesse per questa applicazione.

Va notato che l'integrato predispone al suo interno di una protezione da temperature troppo elevate (OTP) in grado di disabilitare la logica di controllo al superamento di una soglia di 150 °C.

¹⁵Maggiore è la capacità d'uscita, maggiore sarà il periodo che intercorre tra due impulsi di energizzazione.

PIN 1-2-3: GND

A questo piedino è collegata la massa dell'integrato. Essendo un dispositivo di potenza con un MOS in commutazione i pin 1 e 3 fungono anche da buoni conduttori termici quindi è consigliabile collegarli ad un piano di massa in rame per una più efficace dissipazione di temperatura.

PIN 4: FB

L'ingresso FB permette di conoscere, mediante un opportuno partitore resistivo, qual'è la tensione presente sull'avvolgimento ausiliario presente al primario, utilizzato per conoscere la tensione sul secondario. Il partitore resistivo è costituito da R40 di 19.1 K Ω e da R48 di 4.7 K Ω (con in più il condensatore C16 di livellamento). Al suo interno l'integrato predispone di una rete compensatrice sul valore di tensione letto dal piedino FB e di una protezione da sovratensioni (OVP).

PIN 5: VCC

All'accensione viene caricata, mediante una sorgente interna di corrente che preleva corrente dal piedino di DRAIN, la capacità interna di VCC fin tanto che non viene superata una soglia nell'intorno dei 17 V, dopo di che la sorgente viene disattivata. Successivamente all'accensione l'alimentazione viene fornita dall'avvolgimento ausiliario del primario mediante la rete composta dal diodo D9, dalla resistenza R34 da 10 Ω e dalla capacità C13 da 2.2 μ F con il compito di limitare gli spike. Un detector di UnderVoltage Protection (UVP) posto su questo piedino pone l'integrato al riparo da tensioni inferiori ai 8.5 V che fanno lavorare il dispositivo in condizioni non ottimali. Qualora si verificasse questo, viene disabilitata la rete logica interna e si ripete l'accensione tramite sorgente di corrente interna.

PIN 6: SOURCE

A questo pin è collegato il source del MOS interno. Per monitorare la corrente che circola sul MOS si utilizza una resistenza di sensing posta su questo piedino: nel progetto è stata scelta la resistenza R51 da 1.8 Ω . Il comparatore interno confronta la tensione di picco di source con una soglia V_{peak} regolata dalla logica interna dell'integrato; l'uscita del comparatore fornisce informa-

zioni utili al modo operativo del controller.

PIN 7: DRAIN

Rappresenta la connessione del drain del MOS interno (che presenta una tensione di *breakdown*) e viene perciò collegata direttamente al primario del trasformatore senza ulteriori reti di condizionamento. Attraverso questo piedino passa la corrente di carica del condensatore all'avvio del dispositivo.

Capitolo 4

Firmware

1 Il microcontrollore STM32F103CB

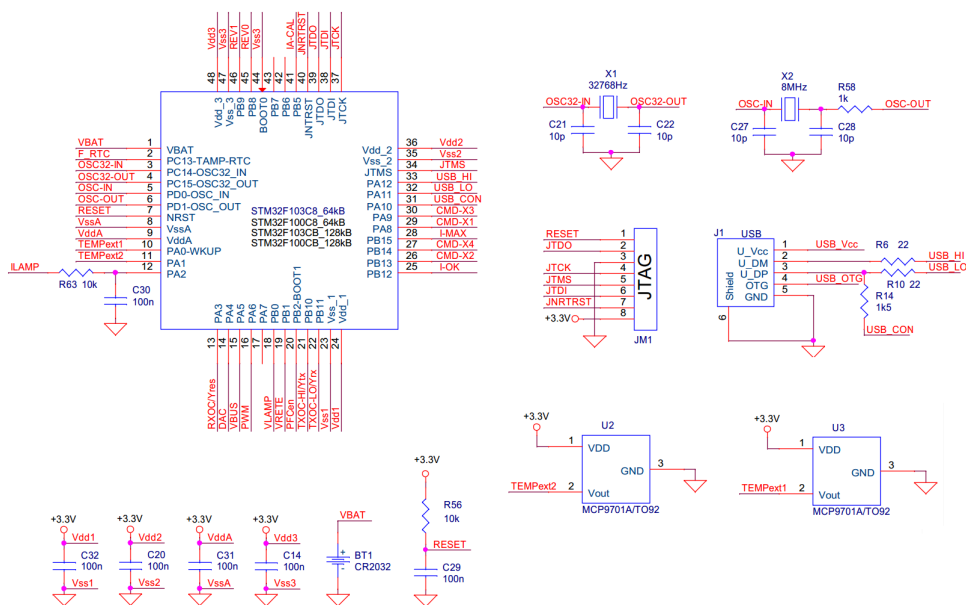


Figura 4.1: Schematico del microcontrollore STM32F103CB utilizzato nel progetto, delle sue alimentazioni, del connettore USB, del JTAG, dei due oscillatori al quarzo e dei due sensori di temperatura

Il microcontrollore utilizzato nel progetto è il STM32F103CB della *ST-Microelectronics* con *package* LQFP48 (la piedinatura e i segnali ad essa associata sono mostrati in figura 4.1). Questo microcontrollore dispone di un

core *CORTEX* a 32 bit della *ARM* in grado di operare a frequenze di 72 MHz. Contiene al suo interno una SRAM da 20 KByte e flash memory da 128 KByte identificata dalle lettere “CB” della sigla (il microcontrollore con memoria da 64 KByte riporta nella sigla l’identificativo “C8”) che lo classifica nella categoria dei microcontrollori *medium-density*. L’alimentazione del dispositivo di 3.3 V viene fornita in uscita dal regolatore di tensione AP7215 con il consueto condensatore di bypass, ma nel caso di guasti su questo stadio (o i suoi precedenti) vi è una batteria che funge da alimentazione tampone, continuando a tener attivo il microcontrollore.

Due oscillatori al quarzo esterni da 32 KHz e 8 MHz (con in aggiunta le capacità da 10 pF) servono al microcontrollore per generare il proprio clock interno: l’oscillatore a 32 KHz viene utilizzato per gestire la rete RTC¹, mentre l’oscillatore a 8 MHz viene moltiplicato tramite un *prescaler* per generare il clock a 72 MHz utilizzato dal microcontrollore come frequenza di lavoro. Le periferiche presenti all’interno del dispositivo:

- 4 *timer* con contatori a 16 bit, dei quali uno - TIM1 - con possibilità di esser utilizzato per generare segnali PWM, dead-time, ecc...;
- *DMA controller* a 7 canali;
- 2 *convertitori analogico/digitali* a 10 canali e risoluzione a 12 bit dei quali il primo può esser gestito mediante DMA;
- *interfacce di comunicazione*: 2 interfacce I²C, 3 USART, 2 SPI, una interfaccia CAN ed una USB;
- *porte general-purpose*: il dispositivo possiede 2 porte a 16 canali, porta A e porta B, per un totale di 37 porte I/O;
- modalità di *debug* tramite porta di serial wire debug JTAG;

All’interno del dispositivo è presente un sensore di temperatura che si aggiunge ai due sensori di temperatura esterni che tengono monitorata la temperatura del ballast. I convertitori analogici-digitale hanno l’alimentazione propria, differente da quella del dispositivo per fornire al convertitore una tensione di riferimento la più stabile possibile. Nel progetto questa tensione viene impostata a 3.3 V.

¹il Real-Time Clock fornisce un insieme di contatori in esecuzione continua anche in caso di alimentazione da batteria, utilizzati perciò per generare funzioni quali calendario, orologio, ecc...

2 Il timer **TIM1** per comandare i MOS

Lo studio e la corretta implementazione del timer TIM1 ha rappresentato uno step molto importante della programmazione del microcontrollore in quanto tramite esso - TIM1 - vengono generati i 4 segnali di comando dei MOS del ponte. La dinamicità con la quale la lampada può essere facilmente comandata passa proprio dall'immediatezza con la quale alcuni parametri riguardanti il timer vengono impostati all'avvio e modificati *runtime*. La principale forma di interazione con la lampada avviene quindi andando a modificare questi parametri, perciò TIM1 va analizzato approfonditamente.

Il timer TIM1, a differenza degli altri 3 timer presenti, realizza segnali PWM mediante una struttura interna al microcontrollore in grado di generare un segnale di duty-cycle voluto ed il suo complementare, introducendo tra le commutazioni il *dead-time*, la durata del quale viene impostata dal programmatore che può scegliere anche di disabilitare tale funzione. Questa caratteristica di TIM1 risulta molto utile e versatile per generare i segnali di comando dei MOS appartenenti allo stesso leg (M1 e M2 oppure M3 e M4). Saranno perciò 2 i segnali PWM, uno per il leg sinistro e uno per il leg destro, anche se, come spiegato nel paragrafo successivo, tramite la programmazione *centerAligned* i due segnali risultano dipendenti l'uno dall'altro.

Infine settando il periodo di conteggio di TIM1 è possibile impostare a quale frequenza far commutare i MOS².

2.1 Inizializzazione di **TIM1** nel file **myBRIDGE.c**

La configurazione di TIM1 viene eseguita all'interno del file `myBRIDGE.c` (Appendice A) proprio ad indicare come il settaggio dei parametri del timer sia in stretto collegamento con i segnali di pilotaggio del ponte. Mediante la funzione `myBRIDGE_Init` viene in prima istanza abilitato il clock sia nel timer TIM1 sia nelle porte A e B; successivamente viene inizializzata la struttura di configurazione dei pin `GPIO_InitStructure` ed infine vengono inizializzate le strutture costituenti il timer: `TIM_TimeBaseStructure`, `TIM_BDTRInitStructure`, `TIM_OCInitStructure`.

²Tale parametro risulta fondamentale nella determinazione della potenza dissipata per commutazione.

GPIO_InitStructure

Questa struttura “collega” i segnali generati da TIM1 ai piedini del microcontrollore; nello specifico:

- GPIO_Pin_8, PORT A = **CMD-X1** = piedino 28 → PWM1
- GPIO_Pin_9, PORT A = **CMD-X3** = piedino 29 → PWM2
- GPIO_Pin_13, PORT B = **CMD-X2** = piedino 26 → PWM1N
- GPIO_Pin_14, PORT B = **CMD-X4** = piedino 27 → PWM2N

Le quattro porte I/O sopraindicate vengono configurate in modalità *push-pull* con velocità di commutazione impostata a 50 MHz (il massimo selezionabile).

TIM_TimeBaseStructure

Il clock interno del microcontrollore - `SystemCoreClock` - è di 72 MHz; tale frequenza risulta troppo elevata come base dei tempi per l’incremento del contatore associato a TIM1. Si è perciò deciso di utilizzare il `Prescaler` del timer per scalare di un fattore 3 la frequenza di aggiornamento del contatore ottenendo così una più agevole frequenza di 24 MHz ($T_{CK} = 41.67$ ns).

Vi sono essenzialmente 3 tipi di conteggio nel timer: `Up` incrementa il contatore ad ogni T_{CK} fino al raggiungimento del valore presente nel registro `ARR` (Auto-Reload Register) per poi azzerarsi e ricominciare l’incremento; `Down` inizia il suo conteggio dal valore di `ARR` e si decrementa ad ogni clock; infine la modalità `CenterAligned` fa partire il contatore da 0 e lo incrementa fino ad arrivare al valore $ARR-1$ ³ per poi iniziare il decremento fino al raggiungimento dello 0. Nel progetto viene implementata quest’ultima modalità di conteggio in quanto tramite essa c’è la possibilità di generare dei segnali PWM centrati (come il nome suggerisce) nel centro di T_{ON} e questo svincola del tutto dalla necessità di creare segnali PWM sfalsati tra loro di $T/2$ come spiegato nel paragrafo 3.1 del capitolo 2.⁴ L’implementazione `CenterAligned` genera segnali nei quali il periodo inizia a metà T_{ON} per concludersi al successivo; in questo modo i segnali X1 e X3 (e i rispettivi negati X2 e X4) risultano in fase tra loro e ciò che distingue i due segnali PWM è unicamente il duty cycle, come mostrato in figura 4.2.

³Il “-1” è dovuto al fatto che il conteggio parte da 0.

⁴In tal caso si parlerebbe di implementazione `EdgeAligned`.

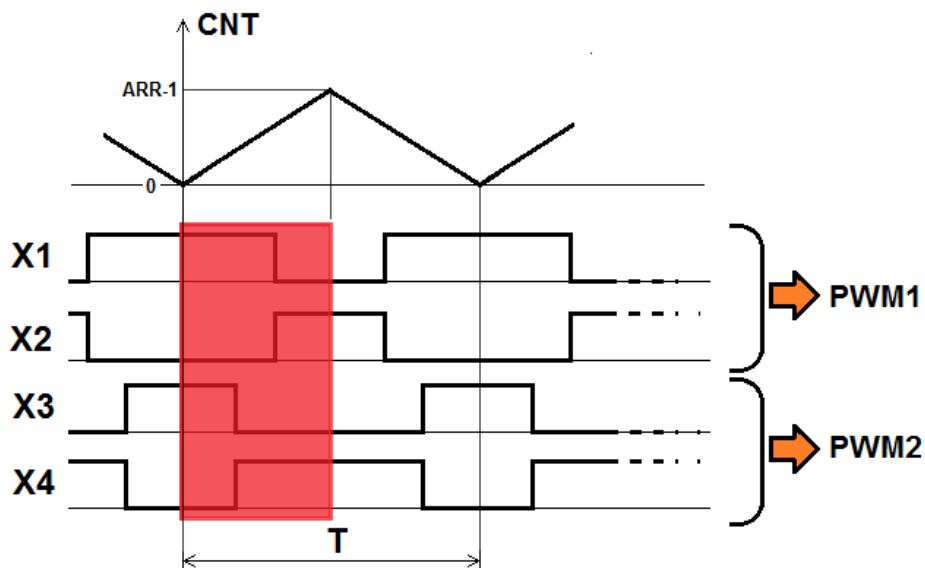


Figura 4.2: Rappresentazione del conteggio CenterAligned per la generazione dei 4 segnali di pilotaggio dei MOS: il duty cycle di PWM1 (che genera X1 e X2) risulta *complementare* a quello di PWM2 (che genera X3 e X4).

Risulta ancora più chiaro se ci si sofferma su metà periodo come evidenziato in figura 4.2: una volta assegnato a X1 il duty cycle desiderato, X2 rappresenta il suo negato mentre X3 ha il duty cycle complementare a X1. Infine X4 altro non è che il negato di X3. Stabilito dunque il duty cycle del primo segnale, i restanti 3 segnali sono determinati di conseguenza.

Si è scelto una frequenza di commutazione dei segnali PWM di 24 KHz ($T = 41.67 \mu s$) ponendosi così al di fuori della banda audio. Tale valore può essere impostato facilmente in quanto viene passato come ingresso alla funzione di inizializzazione `myBRIDGE_Init` sotto il nome `PWM_frequency`. Questa scelta fa sì che all'interno del periodo PWM siano contenuti 1000 T_{CK} : è possibile stabilire il valore da assegnare al registro ARR come il numero di clock che devono intercorrere per completare un semi-periodo PWM; nel progetto implementato ARR conterrà il valore 500. Il registro ARR viene settato agendo sulla variabile `TIM_Period` della struttura.

Scegliendo la modalità di conteggio `CenterAligned3` viene generato il bit di interrupt ad ogni *overflow* e *underflow* del contatore di TIM1.

TIM_OCInitStructure

Se con la precedente struttura si è inizializzato il contatore del timer TIM1 specificandone il periodo, la tipologia di conteggio e il passo T_{CK} di incremento (e decremento) del contatore, ora con la seguente struttura si definisce il duty cycle dei segnali PWM, lo stato delle uscite quando si è in *idle*, l'abilitazione delle uscite, ecc...

Dovendo generare due segnali PWM (seppur correlati tra loro come detto in precedenza) sono state implementate due strutture di *Output Compare* ognuna delle quali ha il compito di generare un segnale PWM e il suo negato.

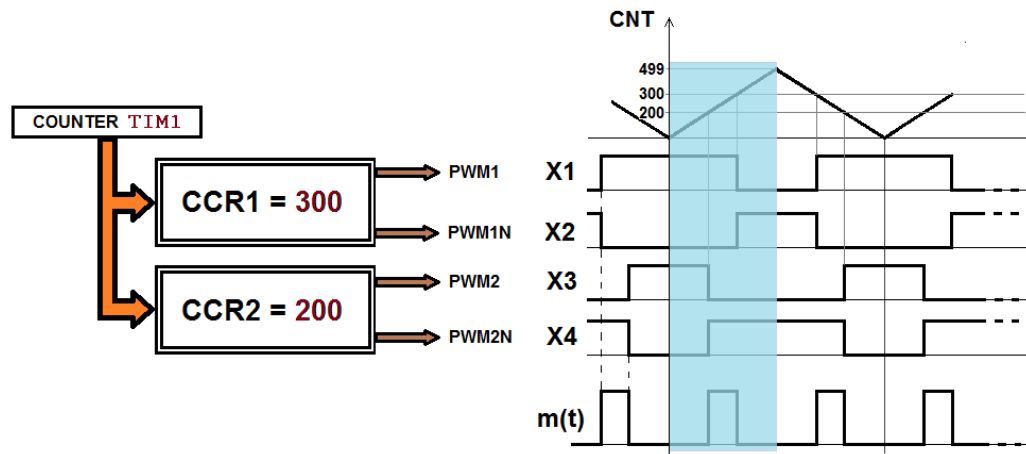


Figura 4.3: I Capture & Compare Register delle due strutture di Output Compare realizzano i due segnali PWM e i rispettivi negati.

Nel Capture Compare Register (CCR) viene settato a quale valore del contatore di TIM1 deve avvenire il cambio polarità dell'uscita. Per farlo si va ad agire sulla variabile `TIM_Pulse` della struttura di Output Compare che, impostando il CCR, stabilisce la pulsazione (o duty cycle) del segnale PWM.⁵ Scegliendo la modalità dell'Output Compare come `TIM_OCMode_PWM1` l'uscita viene tenuta a polarità alta fin tanto che il contatore - CNT - è ad un valore inferiore al CCR, mentre viene mantenuta bassa quando il contatore è a valori superiori.⁶ Assegnare al CCR valori superiori a quelli di ARR (500 in questo progetto) risulta inefficace in quanto la struttura di Output Compare manterrà sempre alta la polarità del segnale PWM e, parimenti, manterrà

⁵In automatico viene generato anche il segnale PWM negato.

⁶La modalità `TIM_OCMode_PWM2` opera in maniera opposta.

ad una polarità bassa il segnale PWM negato; una buona programmazione di TIM1 perciò impone che ai registri CCR siano assegnati valori compresi tra 0 e ARR-1.

`DutyCycle` è la variabile passata alla funzione di inizializzazione del ponte tramite la quale viene specificato il duty cycle (espresso in millesimi) del segnale PWM1 che pilota il MOS M1;

Per conoscere a quale valore impostare il registro CCR1 ci si serve di una semplice proporzione:

$$DutyCycle : 1000 = CCR1 : ARR \implies CCR1 = \frac{DutyCycle \cdot ARR}{1000} \quad (4.1)$$

Il CCR2 è complementare al primo perciò viene settato come segue:

$$CCR2 = ARR - CCR1 \quad (4.2)$$

duty cycle LAMPADA δ [%]	duty cycle PWM1 $D1$ [%]	duty cycle PWM2 $D2$ [%]	CCR1	CCR2
+1000	1000	0	500	0
+500	750	250	375	125
+200	600	400	300	200
+100	550	450	275	225
+40	520	480	260	240
0	500	500	250	250
-40	480	520	240	260
-100	450	550	225	275
-200	400	600	200	300
-500	250	750	125	375
-1000	0	1000	0	500

Tabella 4.1: Relazione tra duty cycle (di lampada e PWM) e CCR: i valori dei duty cycle, così come all'interno del firmware, sono espressi in millesimi: si evita così l'utilizzo di variabili in virgola mobile, cioè `float`.

Nella struttura di Output Compare è possibile definire l'abilitazione e la polarità delle uscite ed infine specificarne il loro stato quando il microcontrollore è in *idle*.

Servendosi della formula 2.2 del paragrafo 3.1 del capitolo 2 che lega il duty cycle di lampada δ con il duty cycle di leg D e servendosi delle formule 4.1 e 4.2 che permettono di ricavare i valori da assegnare ai Captur Compare Register delle due strutture di Output Compare sulla base del duty cycle di leg è possibile ricavare quanto mostrato in tabella 4.1.

Analizzando ad esempio il caso limite di energizzazione della lampada mediante un duty cycle di lampada del 100% ciò vuol dire tenere sempre accesi i MOS sulla diagonale positiva (M1 e M4) e per farlo il CCR1 viene settato a 500 mentre il CCR2 è settato a 0.

TIM_BDTRInitStructure

Con questa struttura vengono principalmente gestiti il *dead-time* e la funzione di *break*: quest'ultima abilita oppure disabilita la possibilità di ricevere come input delle interruzioni in grado di forzare l'uscita a livelli logici alti o bassi. In questo progetto la funzione di break viene disabilitata.

Il dead-time risulta facilmente programmabile mediante un registro a 10 bit contenente il numero di clock che devono intercorrere tra il cambio di polarità del segnale PWM e il cambio di polarità del suo negato durante il quale le uscite sono mantenute entrambe basse. È bene precisare che il numero di clock è aggiornato con frequenza di sistema pari a 72 MHz, in assenza quindi di *prescaler* come nel caso di TIM1. Salvando nella variabile DEADBAND_TIME_NANOSEC il valore in secondi del dead-time, il registro viene settato di conseguenza:

```
(uint16_t) (DEADBAND_TIME_NANOSEC * SystemCoreClock)
```

dove si è utilizzato il cast della variabile a intero a 16 bit senza segno per mantenere solo la parte intera del prodotto.

Nel progetto si è scelto di implementare un dead-time di 100 ns, perciò alla variabile DEADBAND_TIME_NANOSEC si è assegnato il valore di 100E-9.

A completamento della funzione myBRIDGE_Init () di inizializzazione del ponte è stato abilitato e inizializzato il Nested Vectored Interrupt Controller (NVIC) che è il vettore di gestione degli interrupt. Nel caso specifico di TIM1

gli interrupt da gestire riguardano gli *update* che avvengono sul contatore, cioè tutte le volte che il contatore va in *underflow* e *overflow* viene generato e “ascoltato” l’interrupt.

Una volta inizializzato il ponte e volendo generare, ad esempio, un duty cycle di lampada del +35% (corrispondente a duty cycle di PWM1 del 67.5% e duty cycle di PWM2 del 32.5%) i quattro segnali di pilotaggio dei MOS, visualizzati mediante l’oscilloscopio *Textronix* presente in UMPI sono mostrati in figura 4.4.

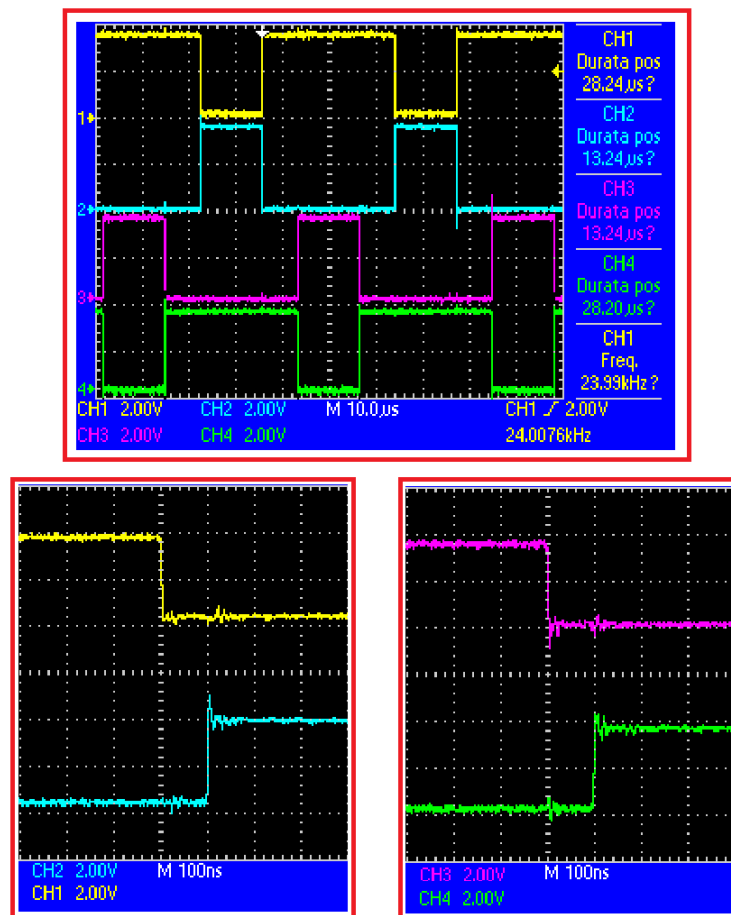


Figura 4.4: I quattro segnali di pilotaggio generati dal microcontrollore e prelevati sugli appositi piedini. Nei due riquadri inferiori lo zoom mostra il dead time di 100 ns introdotto dal microcontrollore all’atto della commutazione di PWM1 (X1 e X2) e di PWM2 (X3 e X4).

2.2 Funzioni di controllo del ponte

In seguito alla funzione di inizializzazione del ponte discussa nel paragrafo precedente verranno ora analizzate tutte le altre funzioni di controllo e modifica di stato del ponte che sono state implementate nel file `myBRIDGE.c` (Appendice A) e poi utilizzate nell'algoritmo di accensione e gestione della lampada.

myBRIDGE_Stop(void)

con questa funzione vengono disabilitate tutte le uscite PWM generate da TIM1 e quindi i segnali di pilotaggio dei MOS sono posti a livello logico basso. Anche gli interrupt generati da TIM1 vengono disabilitati, mentre ciò che non viene fermato è il contatore che quindi continua il suo conteggio.

myBRIDGE_Start(uint16_t polar)

questa funzione permette di abilitare le uscite PWM. La necessità di far funzionare il ponte in regime discontinuo durante la fase di accensione per portare ai capi della lampada la tensione sufficiente a far entrare in funzione lo spark-gap ha portato all'introduzione della variabile d'ingresso `polar` mediante la quale si sceglie uno dei tre modi di utilizzo del ponte: *polarizzazione positiva*, *polarizzazione negativa* oppure *tutti attivi*. L'idea iniziale era quella, ad esempio in *polarizzazione positiva*, di mantenere spenti i MOS M2 e M3 appartenenti alla diagonale secondaria e di far funzionare solo i MOS M1 e M4 che, seppur con un duty cycle di lampada non eccessivamente grande, riescono a portare in uscita dal ponte tutta la tensione del DC-Link. Le prove svolte hanno però dimostrato che ciò non è possibile in quanto se il MOS M2 viene sempre mantenuto spento la capacità di bootstrap del leg sinistro non riuscirà mai a caricarsi, perciò il MOS M1 risulterà sempre spento e la tensione d'uscita nulla. La soluzione finale prevede di abilitare, in caso di *polarizzazione positiva*, i segnali di pilotaggio dei MOS M1, M2 e M4 del ponte e di applicare un duty cycle molto grande del segnale PWM1 in modo che nel breve tempo nel quale è attivo solo il MOS M2 si ha la carica della capacità di bootstrap, fondamentale per la successiva accensione di M1.

Il caso di *polarizzazione negativa* è del tutto analogo con la differenza che il MOS ad esser mantenuto costantemente spento è M1 e la tensione del DC-

Link viene portata al carico mediante i MOS M2 e M3 (questa volta sarà il segnale PWM2 ad avere un duty cycle molto grande).

Infine la modalità di funzionamento del ponte con le uscite tutte abilitate viene utilizzata a lampada accesa per regolare il flusso di corrente che circola sul ponte.

Al termine della funzione vengono controllati i due CCR e sulla base dei loro valori viene settata la variabile `polarity`⁷ che tiene traccia dell'attuale polarizzazione. Tale variabile deve essere opportunamente aggiornata ogni qualvolta si applica l'inversione di polarità del ponte.

`myBRIDGE.ChangeDuty(uint8_t byState, uint16_t step)`

questa funzione viene utilizzata per incrementare o decrementare il duty cycle di lampada. La variabile d'ingresso `byState` specifica se vi deve essere incremento o decremento del duty cycle, mentre la variabile `step` indica quale deve essere il passo di modifica: ciò che vengono incrementati e decrementati sono i registri CCR delle due strutture di Output Compare. Se, ad esempio, si volesse incrementare di 5 step il duty cycle partendo dalla situazione mostrata in figura 4.3 si otterrà il CCR1 a 305 mentre il CCR2 sarà a 195 con una modifica totale su lampada di 10 step.

Il caso di incremento/decremento minimo di 1 step su leg porta a variazioni di 2 step su lampada, corrispondenti a modifiche del duty cycle di lampada del 0.4%.

Risulta fondamentale sapere qual'è la polarizzazione del ponte nel momento in cui viene chiamata questa funzione in quanto incrementare il duty cycle in caso di polarizzazione negativa vuol dire decrementare CCR1 e quindi incrementare CCR2. Per questo motivo il primo controllo da fare è sulla variabile `polarity` per poter poi agire di conseguenza.

Una volta instaurato l'arco elettrico tra gli elettrodi e che la lampada risulta accesa, avere duty cycle di lampada superiori al 40% è pressoché impossibile: con la tensione di DC-Link a 400 V e un duty cycle del 40% ai capi della lampada sono presenti 160 V e questa situazione è difficile che si

⁷A `polarity = 1` corrisponde polarizzazione positiva mentre a `polarity = 0` corrisponde polarizzazione negativa.

verifichi, anche nel caso di lampada in avanzato stato d'usura e verso il fine vita.

Un altro inconveniente riscontrabile consiste nell'avere un indesiderato cambio di polarità e di duty cycle a fronte di un decremento non controllato: se ad esempio il ponte si trovasse nella condizione di *polarizzazione positiva* al +2%, ciò significa che il CCR1 è a 255 mentre il CCR2 è a 245; se ora venisse chiamata questa funzione con la richiesta di decrementare il duty cycle di 20 step il registro CCR1 assumerebbe il valore di 235 mentre il CCR2 il valore di 265 corrispondenti ad una *polarizzazione negativa* con duty cycle di lampada di -6%. Il risultato quindi, oltre che un cambio di polarità, porta un incremento del duty cycle benché la richiesta fosse quella di decremento.

Sono stati perciò introdotti dei controlli al fine di porre dei limiti (superiori e inferiori) sui valori che assumono i registri CCR in seguito alla loro modifica, oltre che resettare la variabile `polarity`.

myBRIDGE_ChangePolarity(void)

questa funzione permette di invertire la polarità del ponte mantenendo inalterato il duty cycle. Per farlo basta semplicemente scambiare tra loro i due CCR.

myBRIDGE_SetDuty(int16_t NewDuty)

questa funzione permette di settare il duty cycle di lampada mediante la variabile `NewDuty` passata in ingresso alla funzione che esplicita il duty cycle espresso in millesimi da assegnare a PWM1. Per conoscere il duty cycle di lampada e la polarizzazione ci si affida all'equazione 4.3:

$$DutyLampada = NewDuty - (1000 - NewDuty) \quad (4.3)$$

Alcuni esempi sono riportati in tabella 4.1.

TIM1_UP_IRQHandler(void)

questa procedura è richiamata ogni volta che TIM1 genera l'interrupt e cioè ad ogni *underflow event* e ogni *overflow event*: viene incrementata la variabile `myBRIDGE_PWM_Tick` che tiene traccia del numero di volte che la procedura è stata richiamata. Grazie ad essa è possibile stabilire quando invertire la

polarità del ponte mediante la funzione `myBRIDGE.ChangePolarity()`. Essendo quindi la variabile `myBRIDGE_PWM_Tick` aggiornata ogni mezzo periodo PWM, il numero di tick che devono trascorrere tra un'inversione e l'altra è espresso nella variabile `tick_of_inversion`:

$$tick_of_inversion = \frac{myBRIDGE_FREQUENCY}{LOW_FREQUENCY} \quad (4.4)$$

dove `myBRIDGE_FREQUENY` è la frequenza PWM settata nella funzione di inizializzazione del ponte, mentre `LOW_FREQUENCY` è la frequenza alla quale si vuole l'inversione di polarità. Nel progetto quest'ultima frequenza è stata scelta di 80 Hz, ottenendo `tick_of_inversion` pari a 300. Ad ogni inversione la variabile `myBRIDGE_PWM_Tick` viene azzerata.

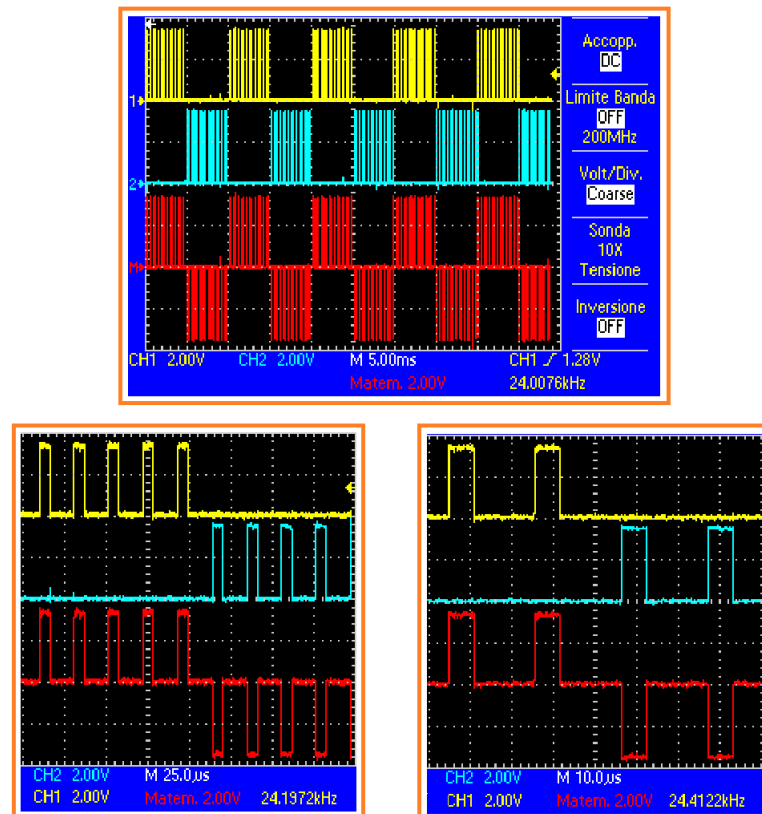


Figura 4.5: A frequenza di inversione di 100 Hz: in giallo il prodotto tra X1 e X4, in azzurro il prodotto tra X2 e X3; in rosso viene riportata la differenza dei due segnali, rappresentante la legge di commutazione $m(t)$ applicata alla lampada (nei riquadri inferiori due diversi zoom effettuati in prossimità dell'inversione).

La funzione `lamp_update_value()` (eseguita quando sono trascorsi la metà dei tick necessari all'inversione, cioè 150) verrà analizzata nel paragrafo 5.3, così come la funzione `lamp_freezing_value()` (eseguita assieme alla funzione di cambio polarità).

3 Gestione degli ingressi analogici

Gli ingressi analogici controllati dal microcontrollore vengono gestiti all'interno del file `anal_input.c`. Mediante il convertitore analogico/digitale ADC1 vengono digitalizzati e successivamente elaborati 6 ingressi analogici:

- **TEMP_EXT1:** GPIO_Pin_0, PORT_A → ADC_Channel_0
- **TEMP_EXT2:** GPIO_Pin_1, PORT_A → ADC_Channel_1
- **VBUS:** GPIO_Pin_5, PORT_A → ADC_Channel_9
- **VLAMP:** GPIO_Pin_0, PORT_B → ADC_Channel_8
- **VRETE:** GPIO_Pin_1, PORT_B → ADC_Channel_5
- **ILAMP:** GPIO_Pin_2, PORT_A → ADC_Channel_2

Il convertitore ADC1, che dispone di 10 canali ai quali collegare 10 possibili ingressi analogici, viene gestito dal DMA controller che trasferisce continuamente i valori convertiti in memoria, senza l'ausilio del microcontrollore che quindi non viene "disturbato" e può continuare a svolgere le sue funzioni: il DMA controller utilizza uno dei suoi 12 canali per trasferire continuamente il dato presente nel registro d'uscita del convertitore in memoria occupandosi di sincronizzare e coordinare il trasferimento dei dati.

In memoria è stato implementato un buffer in grado di contenere 4 valori, ciclicamente aggiornati dal DMA, di tutte 6 le grandezze analogiche: il buffer è costituito perciò da 24 allocazioni di memoria (a sua volta ogni allocazione è costituita da 4 byte).

Prima però è bene specificare che il risultato della conversione viene memorizzato in un registro a 12 bit e ciò comporta una quantizzazione del segnale analogico in 4096 possibili valori. Una prima proporzione dunque consente di ricavare il valore in tensione degli ingressi una volta noto il risultato della conversione, avendo posto uguale a 3300 mV il campione 4095:

$$ConvValue : VoltValue = 4095 : 3300 \rightarrow VoltValue = \frac{ConvValue \cdot 3300}{4095}$$

(4.5)

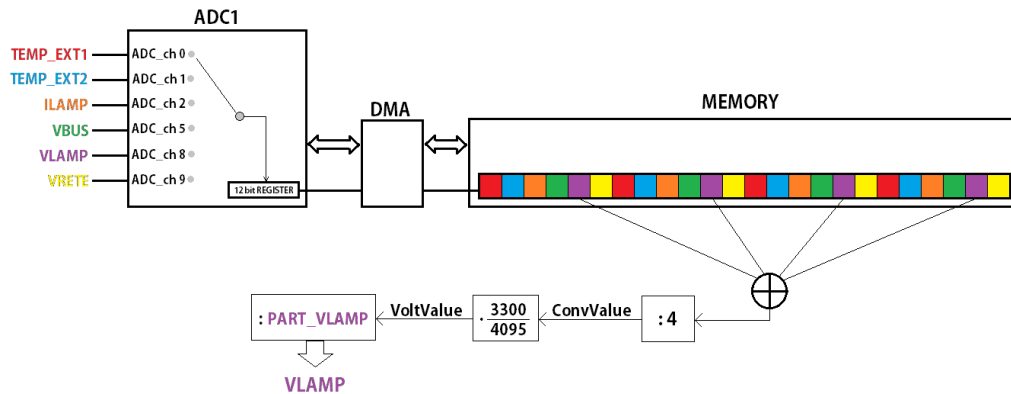


Figura 4.6: Conversione A/D ed elaborazione di una delle sei grandezze analogiche gestite dal microcontrollore; nell'esempio viene proposta la tensione di lampada *VLAMP*.

Attraverso la funzione `STM32_ADC_GetAvgValue` è possibile conoscere il valore di una delle sei grandezze analogiche gestite dal microcontrollore⁸: il primo passo consiste nella media dei 4 dati presenti nel buffer; successivamente viene fatta la conversione dal dato quantizzato al valore, espresso in mV, della grandezza analogica sul piedino del microcontrollore. In ultimo, dividendo il risultato ottenuto per il guadagno della rete resistiva che ha scalato il segnale in esame per portarlo ad un range di tensioni da 0 a 3.3 V, si ottiene la *reale* entità della grandezza elettrica.

Se ad esempio (come mostrato in figura 4.6) si volesse sapere qual'è la tensione ai capi della lampada e in memoria, nelle opportune allocazioni di memoria, fossero presenti 1875, 1936, 1867, 1955, allora `ConvValue` sarebbe 1908 e `VoltValue` varrebbe 1537 mV; essendo il guadagno della rete di sensing pari 19.7 come spiegato nel paragrafo 5.1 del capitolo 3⁹, la tensione di lampada risulta di 78 V.

⁸Quale delle sei viene specificato come parametro d'ingresso della funzione.

⁹Il guadagno della rete è stato moltiplicato per 1000 per ottenere così direttamente il risultato espresso in V e non in mV.

4 Ingressi e uscite general-purpose

All'interno del file `gpio.c` sono sviluppati e gestiti tre segnali: l'uscita *PWM* i due ingressi I_{OK} e I_{MAX} .

Il segnale PWM viene utilizzato per generare la soglia del limitatore di corrente analogico. Dunque in assenza del convertitore DAC si è proceduto con la generazione di un segnale PWM che mediante un filtro RC restituisce il valor medio. Per farlo è stato implementato e inizializzato un contatore: TIM3. La frequenza di aggiornamento del contatore è di 24 MHz ed è stato scelto una modalità di conteggio UP mentre la frequenza del segnale PWM generato è di 24 KHz.¹⁰ Come per TIM1 anche per questo contatore la struttura di Output Compare lavora in modalità `TIM_OCMode_PWM1`, quindi l'uscita viene mantenuta alta fin tanto che il contatore ha un valore inferiore a quanto scritto nel registro CCR, altrimenti viene posta bassa.

Per settare il registro CCR che stabilisce la pulsazione del segnale PWM e di conseguenza il suo valore medio è stata implementata la funzione `Set_ILampReference(curr_max)`. La variabile `curr_max` esplicita quale essere il riferimento o la soglia che si vuol avere sull'ingresso positivo del limitatore analogico. Per rendere coerente questo riferimento di corrente con il segnale *ILAMP* che è scalato di un fattore 0.6875 dal partitore resistivo si moltiplica la variabile `curr_max` per 0.6875. Questo permette di scalare il range di variazione di PWM da 0-4800 (reali valori di corrente espressi in mA che circola sul ponte) a 0-3300 (soglia espressa in mV in ingresso al limitatore analogico). Mediante la proporzione 4.6 è possibile stabilire a quale valore settare il registro CCR della struttura. Avendo una frequenza di aggiornamento del contatore a 24 MHz e la frequenza del segnale PWM impostata a 24 KHz, ARR vale 1000.

$$V_{ref} : 3300 = CCR : ARR \implies CCR = \frac{V_{ref} \cdot 1000}{3300} \quad (4.6)$$

Un controllo effettuato sul valore di CCR che deve essere inferiore o al più uguale a 1000 pone al riparo il corretto funzionamento della rete logica nel caso in cui fosse richiesta una `curr_max` superiore a 4800 mA.

I due segnali I_{OK} e I_{MAX} vengono invece gestiti come segnali di interrupt e sono collegati alla struttura EXTI (External Interrupt) a sua volta collega-

¹⁰Il polo del filtro RC è a 15 Hz.

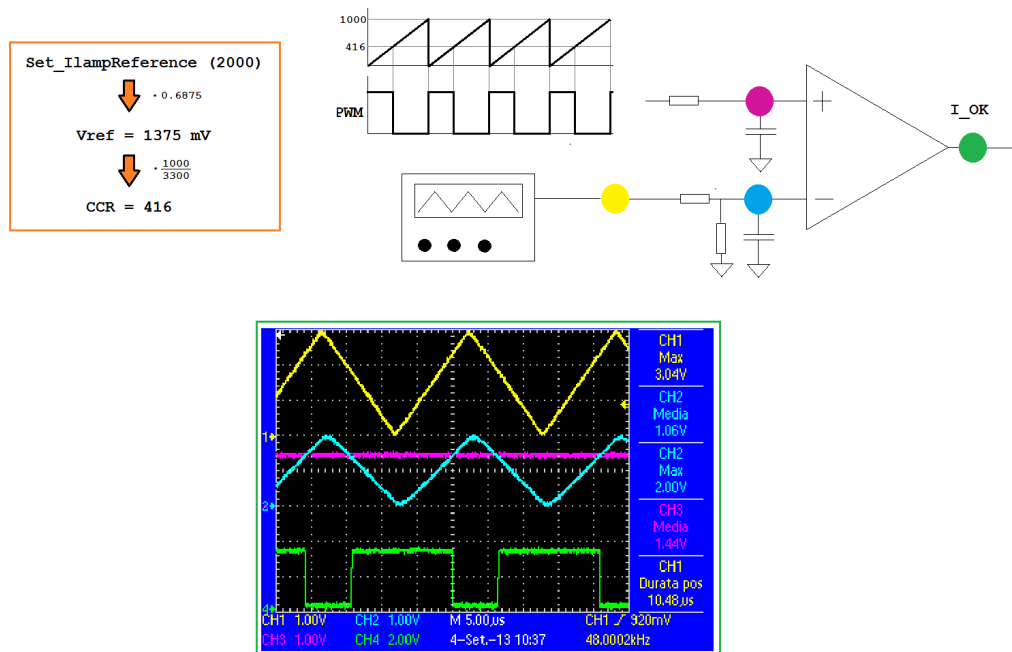


Figura 4.7: Test effettuato alla logica di protezione avendo impostato come corrente di soglia 2000 mA ed avendo simulato la corrente $ILAMP$ come onda triangolare mediante il generatore di funzione.

ta alla struttura NVIC (controller degli interrupt): il microcontrollore deve avviare determinate procedure solo al momento della loro commutazione da livello logico alto a livello logico basso. Entrambi i segnali provengono dall'uscita di comparatori quindi non sono previsti livelli di tensione intermedi. Nella loro configurazione iniziale i due piedini vengono posti come ingressi flottanti e collegati alla struttura EXTI:

- **I_OK**: GPIO_Pin_12, PORT_B → EXTI_Line12
- **I_MAX**: GPIO_Pin_15, PORT_B → EXTI_Line15

Essendo d'interesse la commutazione da livello logico alto a livello logico basso (situazione che si verifica quando la protezione entra in funzione) il trigger della struttura EXTI è di tipo `EXTI_Trigger_Falling`. La procedura di gestione dell'interrupt è chiamata `EXTI15_10_IRQHandler()`: ogni volta che viene richiamata, un primo controllo identifica quale delle due linee ha generato l'interrupt: se l'interrupt proviene dalla linea 15 si è

preso come provvedimento quello di fermare il ponte mediante la funzione `myBRIDGE_Stop()`, se invece l'interrupt proviene dalla line 12 viene incrementata la variabile `CURRENT_LIMIT_REACH` che tiene conto di tutte le volte che è scattata la protezione. Tale variabile sarà poi utilizzata nell'algoritmo di accensione e gestione della lampada descritti nel paragrafo successivo.

5 Accensione della lampada

In questo capitolo viene analizzato l'algoritmo eseguito dal microcontrollore all'accensione del ballast. La procedura di *startup* del microcontrollore riguarda il settaggio dei registri SP e PC, oltre che l'inizializzazione del *vector table* e la configurazione del clock di sistema (*System Clock*). Una volta eseguito questo viene chiamata la funzione `main()` che contiene a sua volta nel suo corpo la chiamata ad altre due funzioni: `init()` e `run()`.

Tramite la funzione `init()` implementata insieme a `Kernel_Init()` e `ErrorsMng_Init()` all'interno del file `inizializza.c` si inizializzano tutte le strutture utilizzate nel firmware come mostrato nel listato 4.1.

Listato 4.1: `inizializza.c`

```
void init(void)
{
    KERNEL_Init();
    ErrorsMng_Init();
    init_process(); // faccio iniziare l'applicazione
}

void init_process(void)
{
    // inizializzo le porte analogiche e gpio
    anal_input_Init();
    gpio_Init();
    // Aspetta per stabilizzare le porte
    Delay_ms(250);
    // inizializzazione ponte con duty di lampada del 50%
    // N.B. è solo inizializzazione, non vengono attivate le uscite PWM.
    myBRIDGE_Init(PWM_FREQUENCY, 500);
    // inizializzazione lampada
    lamp_Init();
    // inizializzazione test delle tensioni di alimentazione
    volt_check_Init();
}
```

La seconda funzione presente nel main, `run()`, manda in esecuzione l'algoritmo di accensione e controllo della lampada. Il listato 4.2 mostra come

tale algoritmo sia composto semplicemente da un ciclo `while(1)`: una volta che il microcontrollore è entrato in questo loop non ne esce più a meno di chiamate ad interrupt o *reset* del microcontrollore. Oltre alle tre funzioni di controllo della lampada, ogni 60 secondi, viene aggiornato un array contenente le variabili di maggiore interesse.¹¹

Listato 4.2: `run.c`

```
void run(void)
{
    time_check=GetSystemTick();
    while (1)
    {
        voltage_check(); // faccio il check delle tensioni di rete e di bus

        lamp_Ignition(); // inizio la fase di iniezione

        lamp_Warmup(); // limitazione di corrente e regolazione di potenza

        // ogni 60 secondi aggiorni i valori fondamentali
        if(KERNEL_Delay_elapsed(time_check,TIME_CHECK_REFRESH))
        {
            param_check[0]=anal_input_get_vrete();
            param_check[1]=anal_input_get_vbus();
            param_check[2]=anal_input_get_tempext1();
            param_check[3]=anal_input_get_tempext2();
            param_check[4]=Get_lamp_current();
            param_check[5]=Get_lamp_voltage();
            param_check[6]=Get_lamp_power();
            param_check[7]=Get_time_Ignition();
            param_check[8]=Get_time_CurrentLimit();
            param_check[9]=Get_time_PowerRegol();
            param_check[10]=Get_polarity();
            param_check[11]=Get_TIM1CH1_Pulse();
            param_check[12]=Get_TIM1CH2_Pulse();
            param_check[13]=Get_lamp_Pulse();
            param_check[14]=Get_lamp_status(V_CHECK);
            param_check[15]=Get_lamp_status(IGNITION);
            param_check[16]=Get_lamp_status(WARMUP);

            time_check=GetSystemTick();
        }
    }
}
```

5.1 La funzione `voltage_check()`

Una volta che il ballast è stato acceso e tutte le strutture del microcontrollore sono inizializzate la prima funzione che viene eseguita è quella di controllo della tensione di linea e della tensione sul DC-Link. Questo check viene fatto per esser sicuri di lavorare con le appropriate tensioni d'alimentazione. Per farlo è stata creata la struttura `volt_check` costituita dalle

¹¹Si è pensato a questa soluzione nel caso si volesse monitorare lo stato della lampada mediante onde convogliate.

variabili `vrete_value` e `vbus_value` sulle quali applicare i dovuti controlli e dalla variabile `state` che identifica lo *stato* nel quale l'algoritmo si trova. Infatti l'approccio che si è dato a questa struttura, così come alla struttura `lamp` e `lamp_duty` che saranno analizzati in seguito è quello di macchina a stati.

Lo stato iniziale della struttura è quello di `VRETE_CHECK` nel quale viene prelevato ogni millisecondo il campione della tensione *VRETE*. Fin tanto che la tensione letta non è superiore della variabile `VRETE_OK`, settata a 280, lo stato non viene modificato. Se passati 100 campioni (`MAX_NUM_TENT`) la tensione letta non è mai risultata superiore a 280 lo stato della struttura diventa `VOLTAGE_FAILURE`, diversamente si passa a `VBUS_CHECK`, abilitando il `PFC_EN`.¹²

In maniera analoga viene testata la tensione *VBUS*, prelevando un campione ogni millisecondo e guardando se è maggiore di 380 (`VBUS_OK`) e dichiarandone l'errore al centesimo campione che ha dato esito negativo. Nel momento in cui la tensione letta risulta superiore a 380 viene modificato lo stato che diventa `VOLTAGE_OK`: qui viene settata `TRUE` la variabile booleana `voltageOK` che sancisce come il check delle tensioni sia andato a buon fine.

5.2 La funzione `lamp_ignition()`

Successivamente al test effettuato sulle tensioni di alimentazione vi è la funzione di *ignition* con la quale si porta la tensione ai capi dello spark-gap ad un livello tale per cui esso scatta ed iniziano i tentativi di accensione della lampada. Per questa funzione ci si è affidati alla struttura `lamp` implementata costituita dai parametri fondamentali riguardanti la lampada quali ad esempio la tensione (`voltage`), la corrente (`current`), la potenza (`power`), lo stato (`state`), ecc...

La funzione di *ignition* è stata strutturata affinché siano limitati al minimo gli stress alla lampada durante questa critica fase. Per questa ragione sono stati implementati due stratagemmi: il primo consiste nella casualità con la quale si decide il verso di polarizzazione della lampada. In questa maniera si

¹²`PFC_EN` viene posto uguale a 0.

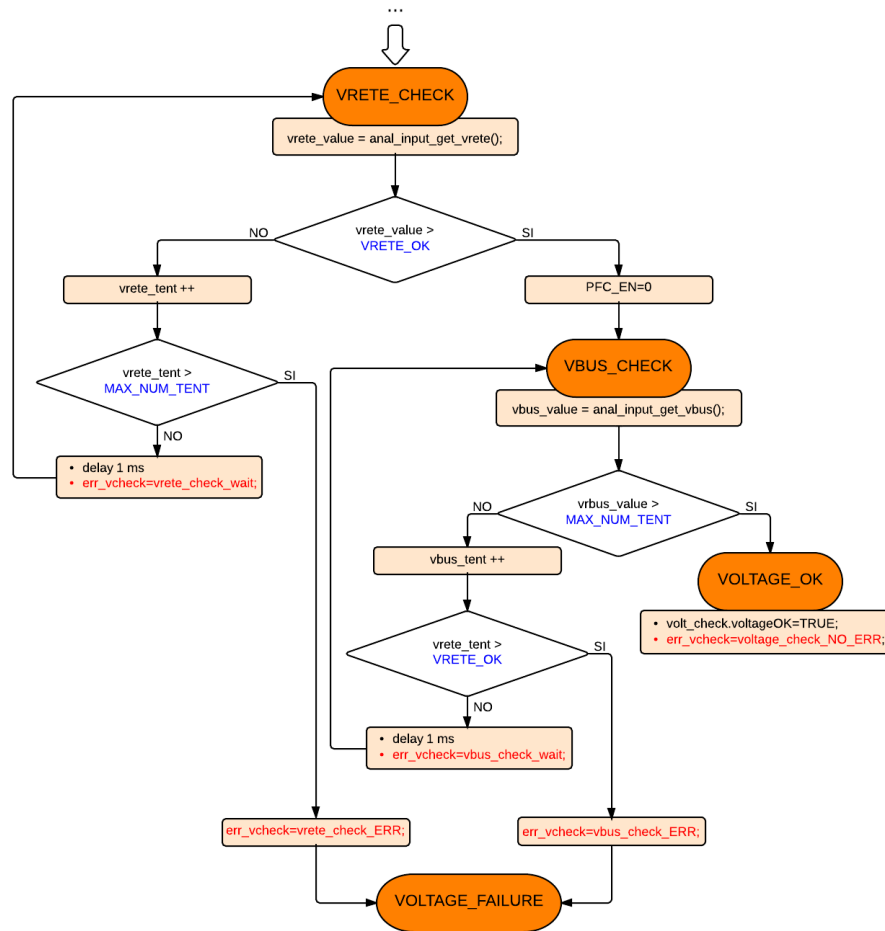


Figura 4.8: Flow chart della funzione di `voltage_check()`.

rende il più possibile uniforme lo stato d'usura degli elettrodi in quanto all'accensione l'elettrodo caldo (quello posto a potenziale alto) viene sollecitato maggiormente dal punto di vista termico per effetto della cataforesi.

La seconda misura di prevenzione contro usure precoci risulta l'evitare di tentar di accendere la lampada insistentemente per lunghi periodi in quanto il mancato innesco dell'arco elettrico può esser causato dal fatto che la lampada proviene da un precedente utilizzo e il gas all'interno del bulbo è ancora in parte ionizzato. Affinché ciò non si verifichi l'algoritmo di ignition prevede intervalli di tempi nei quali la lampada viene sollecitata e si tenta di accenderla, seguiti da intervalli più lunghi nei quali il ponte viene messo in stand-by. Al tentativo successivo, la polarizzazione viene invertita: in questo modo si pone rimedio alla possibilità che il mancato innesco sia cau-

sato dall'usura di uno dei due elettrodi che ne compromette l'accensione. I tentativi di accensione che il ballast esegue devono comunque essere limitati, dopodiché viene sancito lo stato di lampada bruciata e da sostituire.

Come visibile in figura 4.9 la funzione `ignition()` termina il suo processo ad accensione avvenuta, quando cioè viene rilevata una tensione di lampada inferiore ad una determinata soglia.

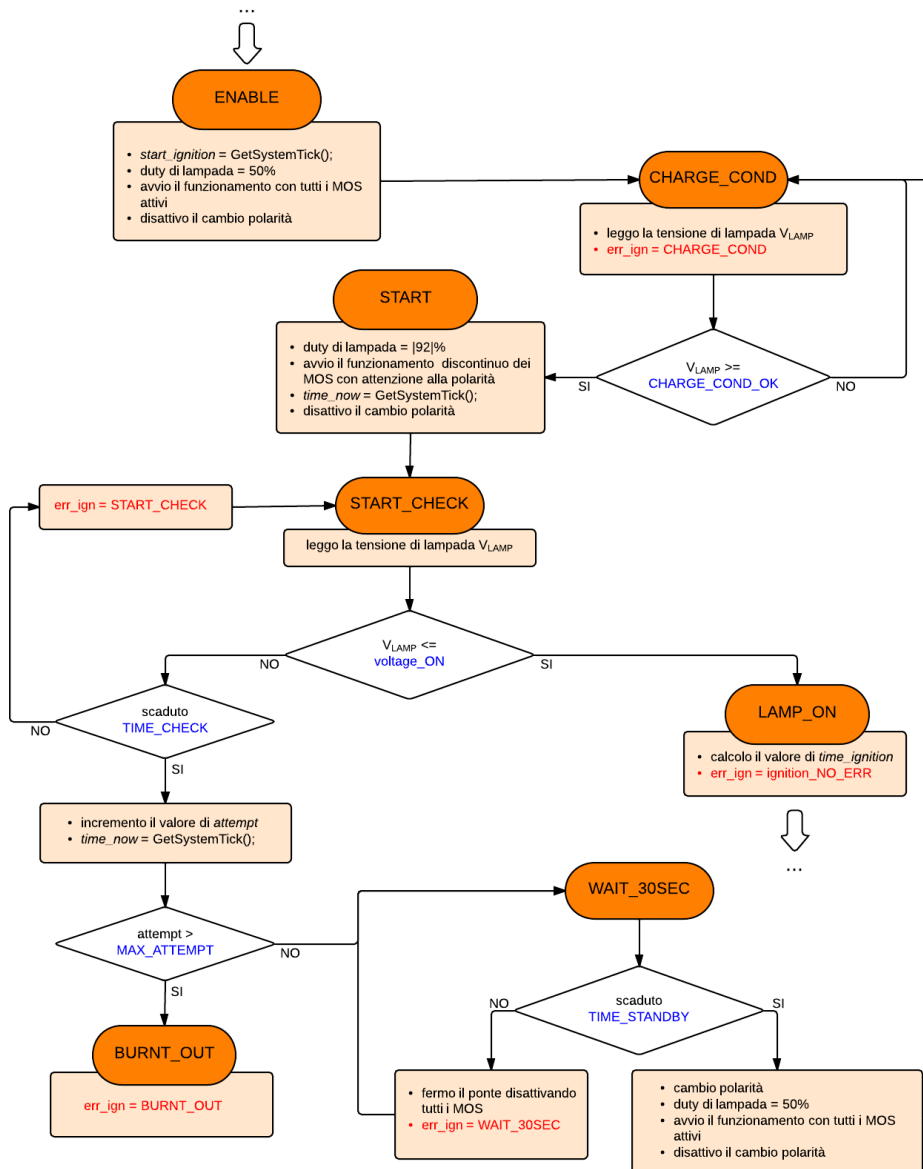


Figura 4.9: Flow chart della funzione di `lamp_ignition()`.

Nello stato `ENABLE`, dopo aver controllato che la variabile `voltage_OK`

sia *TRUE*, viene settato un duty cycle di lampada del 50% con tutti i MOS attivi per poter caricare la capacità in parallelo alla lampada; viene letta continuamente la tensione finché non risulta superiore alla soglia `CHARGE_COND_OK` (138 V), condizione per la quale lo stato diventa quello di `START`.

La polarizzazione con la quale cercare di accendere la lampada viene scelta andando a vedere il resto della divisione per 2 della tensione *VRETE* letta nella fase di inizializzazione della struttura `lamp`: se è 1 la polarizzazione è positiva, se è 0 la polarizzazione è negativa. Sulla base di questo valore viene settato un duty cycle di lampada del 92% facendo attenzione al tipo di polarità e scegliendo di conseguenza il tipo di utilizzo del ponte.¹³ Nello stato `START_CHECK` viene ripetutamente controllata la tensione di lampada e quando questa diventa inferiore alla soglia `voltage_ON` (100 V) la lampada risulta accesa: lo stato diventa `LAMP_ON`, la funzione di ignition è terminata e viene calcolato `time_ignition` che è l'intervallo di tempo trascorso (espresso in millisecondi) all'interno di questa fase di accensione della lampada. Mediante la variabile `TIME_CHECK`, che vale 3000, si selezionano i millisecondi nei quali la funzione deve continuare a tentar di accendere la lampada trascorsi i quali si incrementa di un'unità la variabile `attempt` che conteggia i tentativi di accensione eseguiti. Qualora i tentativi svolti fossero più di 5 (`MAX_ATTEMPT`) la lampada viene dichiarata bruciata e lo stato diventa `BURNT_OUT`, mentre invece se sono disponibili ancora altri tentativi lo stato della lampada diventa `WAIT_30SEC` dove appunto il ponte viene spento per 30 secondi (`TIME_STANDBY`); terminato il tempo d'attesa viene invertita la polarità di energizzazione e lo stato ritorna ad essere `CHARGE_COND`: è possibile fare ora un altro tentativo.

Utilizzando la variabile `err_ign`, aggiornata ogni volta che viene cambiato stato alla lampada è possibile tener traccia del comportamento dell'algoritmo. Come mostrato in figura 4.10 viene riportata la tensione ai capi del condensatore C52 durante la fase di ignition prelevata e visualizzata con l'oscilloscopio durante le prove di accensione della lampada.

¹³`POL_NEG` oppure `POL_POS` nella funzione `myBRIDGE_Start()`.

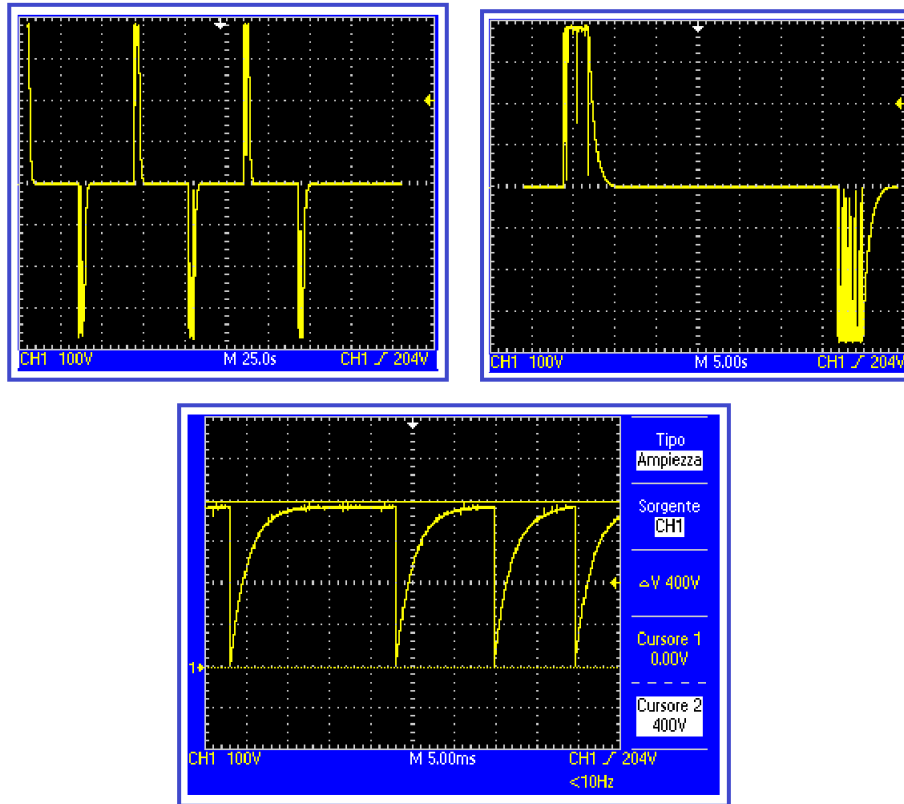


Figura 4.10: Tensione ai capi del condensatore C52 durante la fase di ignition. Nei tre riquadri sono riportate tre diverse visualizzazioni della tensione al variare del *time division* dell'oscilloscopio (25 s, 5 s e 5 ms).

5.3 La funzione `lamp_warmup()`

L'innesco dell'arco elettrico modifica lo stato della struttura `lamp` che diventa `LAMP_ON` e quindi, non essendo tale stato contemplato nella funzione `lamp_ignition()`, il controllo del microcontrollore esce dalla funzione per eseguire quella successiva: `lamp_warmup`. Durante una prima fase verrà applicata la limitazione della corrente che circola sul ponte fino al raggiungimento della potenza nominale di esercizio. A questo punto inizierà una seconda fase di `warmup` nella quale ciò che viene regolata è la potenza; la terminazione di questa seconda fase è dettata dalla invarianza della tensione ai capi della lampada. Ciò che differenzia le due fasi è la variabile da monitorare (in un caso la corrente, in un caso la potenza), ma in entrambi i casi risulta solo una la variabile sulla quale agire: il duty cycle di lampada.

Per prima cosa viene portato il duty cycle di lampada al 40%¹⁴ con MOS tutti attivi evitando il cambio di polarità che potrebbe spegnere l’arco appena innescato: questo valore è stato stabilito a fronte di alcune prove effettuate in quanto un duty cycle troppo piccolo non avrebbe garantito il mantenimento dell’arco elettrico, mentre un duty cycle troppo elevato può risultare rischioso. Qualora il duty cycle risultasse comunque troppo alto si lascia alla rete di protezione il compito di intervenire e interrompere l’energizzazione. A tal proposito la massima corrente ammessa a circolare sul ponte viene scelta essere pari a 2200 mA, per cui la soglia della protezione analogica viene settata a 2200 mA tramite la funzione `Set_IlampReference()`. Infine viene attivato l’interrupt di TIM1 che permette di invertire la polarità del ponte ad una frequenza di 80 Hz (cioè ogni 6.25 ms) e viene modificato lo stato che diventa `LAMP_WAIT`. Qui vi rimane fin tanto che la funzione di `freezing_value()`, lanciata dall’interrupt di TIM1 in corrispondenza del cambio polarità, non ne impone il cambio in `LAMP_ELABORATION`.

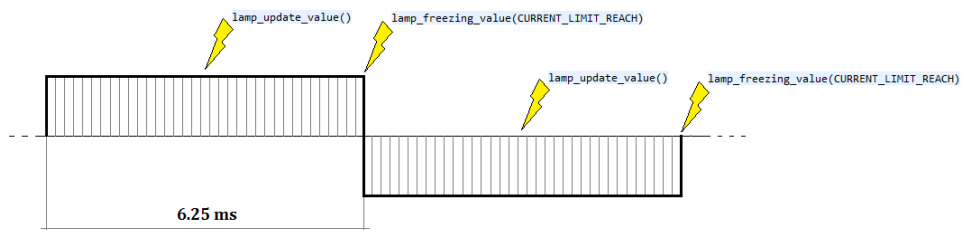


Figura 4.11: Sequenza delle chiamate delle funzioni `lamp_freezing_value()` e `lamp_update_value()` in riferimento alla legge di commutazione $m(t)$.

Tramite la funzione `lamp_freezing_value()` si ha il “congelamento” in voltage della tensione di lampada, in `current_read` della corrente letta sul piedino del microcontrollore e in `count` di `CURRENT_LIMIT_REACH` corrispondente al numero di volte che la protezione è scattata dall’ultimo aggiornamento del duty cycle. All’interno dello stato `LAMP_ELABORATION` e successivamente di `LAMP_POWER_CALCULATE` vengono poi *elaborati* questi dati controllando di quanto la corrente si scosta dalla corrente di riferimento (`current_reference`) e in secondo luogo se la potenza raggiunta dalla lampada è quella nominale di esercizio. I calcoli che vengono svolti portano

¹⁴durante la fase di ignition era al $\pm 92\%$

come risultato finale il settaggio delle variabili `step` (di quanto modificare il duty cycle) e `move` (indica se aumentare o diminuire il duty cycle); lo stato di `lamp` viene riportato in `LAMP_WAIT` in attesa che la funzione `lamp_update_value()`, richiamata anch'essa dall'interrupt di `TIM1`, metta in pratica le modifiche da fare e sia corretto il duty cycle mediante `myBRIDGE_ChangeDuty(move, step)`.

Nello stato `ELABORATION` la prima variabile ad essere controllata è `count`: qualora il suo valore risultasse superiore a 2 l'algoritmo abbassa di uno step il duty cycle (`step=1` e `move=DOWN`) in quanto questo dato è indice di un duty cycle troppo grande che fa scattare ripetutamente la protezione analogica.

Se invece `count` è minore di 2 si può procedere con il calcolo della corrente di lampada: non essendo intervenuta la protezione a bloccare l'energizzazione (cosa che di fatto modifica il duty cycle), nota la corrente `current_read` per conoscere la corrente media di lampada risulta sufficiente dividere tale valore per il duty cycle di lampada applicato. A questo punto si aggiornano gli array contenenti i valori di corrente e di tensione. Queste strutture, di dimensione pari a 160 celle, servono per poter calcolare la potenza su valore mediati di corrente e tensione così da non incappare in errate interpretazioni causate ad esempio da spike. Alla centosessantesima¹⁵ lettura viene calcolata la corrente media (`current_avg`) tra i 160 valori e analogamente la tensione media (`voltage_avg`): il prodotto tra le due quantità determina la potenza di lampada. Il `current_reference` viene modificato di conseguenza:

- se la potenza risulta al di fuori dell'isteresi di ± 1 W centrata in 150 W (`POWER_STEADY_VALUE`) allora viene aggiornato il riferimento di corrente come il rapporto tra `POWER_STEADY_VALUE` e la tensione di lampada, prestando attenzione a non impostare valori superiori a quelli di limitazione di corrente.
- se la potenza è compresa all'interno dell'isteresi non vengono attuate modifiche al riferimento di corrente.

Qualora ad esempio l'algoritmo si trovasse in limitazione di corrente con il riferimento di corrente settato a 2 A, la tensione di lampada ha un valore

¹⁵L'aggiornamento dell'array avviene ogni 6.25 ms, perciò il centosessantesimo elemento viene aggiornato ogni secondo.

sicuramente inferiore a 75 V ed infatti il prodotto è inferiore a 150 W; l'aggiornamento del riferimento imporrebbe valori superiori a 2 A ma ciò non è ammesso per la stessa definizione di *limitazione di corrente*. Invece nel caso in cui la potenza di lampada si trovasse ad esser superiore a 150 W, ciò implicherebbe l'ingresso in *regolazione di potenza* da parte dell'algoritmo e il riferimento di corrente verrebbe aggiornato con valori sicuramente inferiori a quelli di limitazione di corrente.

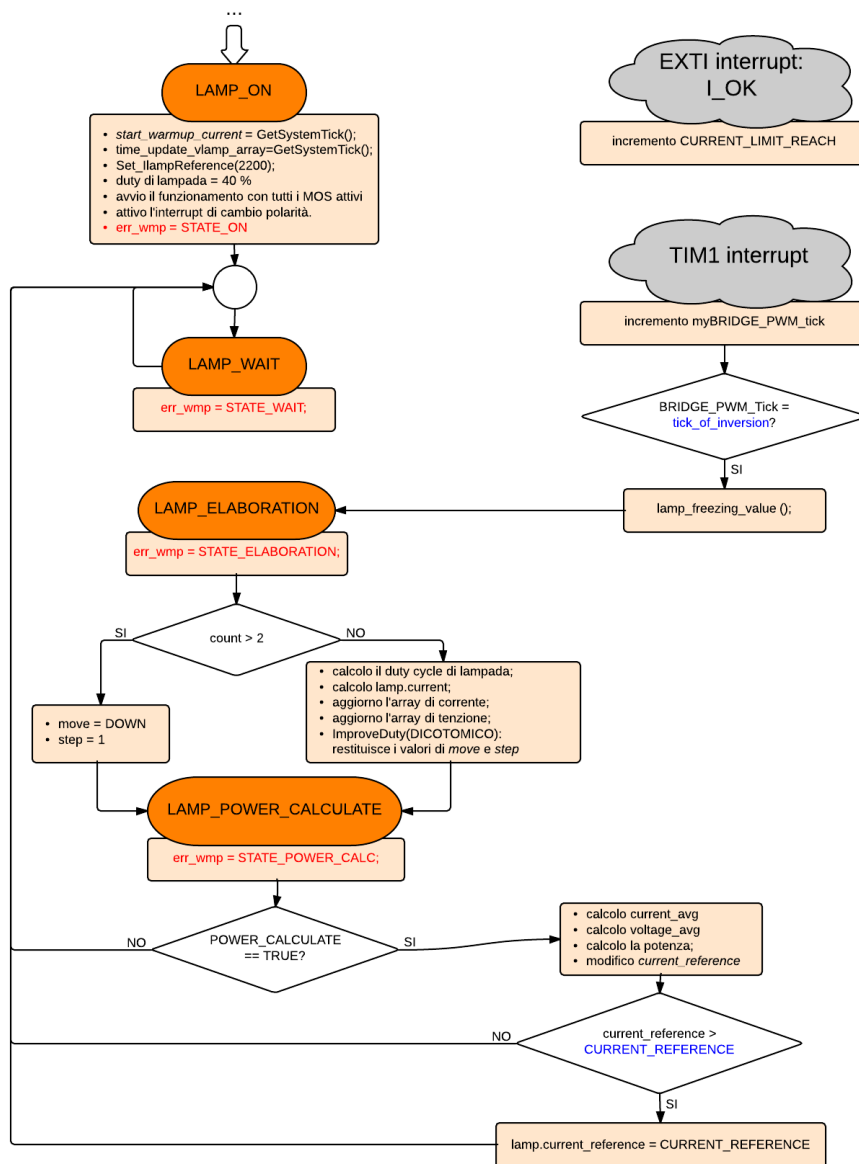


Figura 4.12: Flow chart della funzione di lamp_warmup () .

La durata della fase di warmup in *limitazione di corrente* viene misurata dall'istante in cui viene innescato l'arco elettrico all'istante in cui per la prima volta viene raggiunto il valore di potenza `LAMP_STEADY_VALUE`. Tale entità viene salvata all'interno della variabile `time_CurrentLimit`. Quanto dura invece la fase di warmup in *regolazione di potenza* viene salvato nella variabile `time_PowerRegol` che conta i millisecondi che intercorrono dalla fine della fase di *limitazione di corrente* alla fase di *steady*, identificata dalla invarianza della tensione di lampada. Per farlo viene costituito un array di 6 valori, aggiornato ogni 5 secondi; quando l'ultimo valore letto non si discosta in maniera considerevole dalla media dei 5 precedenti allora la tensione di lampada può considerarsi arrivata a regime.

5.4 Ricerca del duty ottimale

La modifica del duty cycle avviene mediante la funzione `ImproveDuty()`, richiamata ogni volta che l'algoritmo entra nello stato `LAMP_ELABORATION` e può avvenire in modalità `ONE_STEP` oppure `DICOTOMICO`. L'obiettivo di tale funzione è quello di migliorare il duty cycle di lampada affinché la corrente letta sia il più possibile affine a quella che viene impostata come di riferimento, `current_reference`. Nella fase di warmup in limitazione di corrente la `current_reference` viene settata al valore di `CURRENT_REFERENCE`, 2000 mA in questo caso, che rappresenta il valore al quale si vuole tenere limitata la corrente, mentre quando la lampada entrerà in regolazione di potenza il riferimento verrà via via modificato a livelli chiaramente sempre più bassi fino al raggiungimento della fine della fase di warmup nel quale `current_reference` non subirà più sostanziali modifiche. Al termine dell'algoritmo `ImproveDuty()` sono due le variabile che vengono settate: la prima, `move`, stabilisce se aumentare o diminuire il duty cycle, mentre la seconda, `duty_step`, fissa il passo di modifica del duty cycle. L'ausilio di una isteresi centrata sul riferimento di corrente consente di apporre variazioni al duty cycle solo al di fuori di essa. Nel caso di ricerca *dicotomica* del duty cycle ottimale tale isteresi viene scelta di ± 10 mA (`DELTA_DUTY_DICOT`), mentre viene scelta di ± 5 mA nel caso di ricerca *one step*. Nel caso venga utilizzato l'algoritmo in modalità `ONE_STEP`, il parametro `step` sarà posto

uguale a 1 nel caso ci si ritrovi con un valore di corrente al di fuori dell'isteresi, altrimenti viene posto a 0.

L'algoritmo di ricerca *dicotomica* invece è un po' più complesso in quanto la finalità di questa implementazione è quella di una ricerca più veloce del duty cycle ottimale. Per farlo è stata istituita la struttura `lamp_duty` costituita da tre variabili continuamente aggiornate: `direction`, `enable` e `delta_duty`. La logica dell'algoritmo prevede di considerare non tanto il valore del duty cycle, ma bensì l'ultima variazione attuata del duty cycle (`delta_duty`) e in quale direzione (UP, DOWN). Per rendere più veloce il processo si impone una modifica del `delta_duty` per passi ogni volta raddoppiati rispetto al caso precedente. Quando la corrente calcolata ricade all'interno dell'isteresi il duty cycle di lampada non viene modificato e quindi `duty_step` viene posto a 0, mentre il `delta_duty` della struttura viene regolato 1: questo settaggio è stato pensato per evitare il rischio di non far convergere l'algoritmo. Infine la variabile `enable` viene utilizzata per non generare oscillazioni troppo elevate nell'intorno del valore ottimale ricercato: qualora vi fosse la necessità di invertire la direzione di ricerca del duty cycle, questa viene fatta dimezzando il valore di `delta_duty` e invertendo `direction`; viene eseguito un secondo tentativo a parametri inalterati (`enable=0`) ed infine, settando `enable` a 1, l'algoritmo dicotomico torna al suo normale utilizzo.

Il principio di funzionamento dell'algoritmo dicotomico viene mostrato in figura 4.13: l'ipotesi iniziale è quella di avere un `CCR1`¹⁶ di valore 288 e di trovarsi nello stato DOWN1 con un `delta_duty` pari a 8. Nel caso in cui il valore di `CCR1` ottimale da raggiungere fosse 265, l'algoritmo dicotomico impiega 7 cicli, equivalenti a poco più di 40 ms¹⁷. Qualora fosse stato utilizzato l'algoritmo ONE_STEP sarebbero dovuti trascorrere 23 cicli, corrispondenti a circa 140 ms.

¹⁶Nell'esempio vengono riportati i valori del `CCR1` in quanto la funzione `myBRIDGE_ChangeDuty()` viene utilizzata per modificare questo registro, ovvero la pulsazione del PWM1; la pulsazione di PWM2 e il duty cycle di lampada sono modificati di conseguenza.

¹⁷Come mostrato in figura 4.11 il duty cycle viene aggiornato ogni 6.25 ms mediante la funzione `lamp_update_value()`.

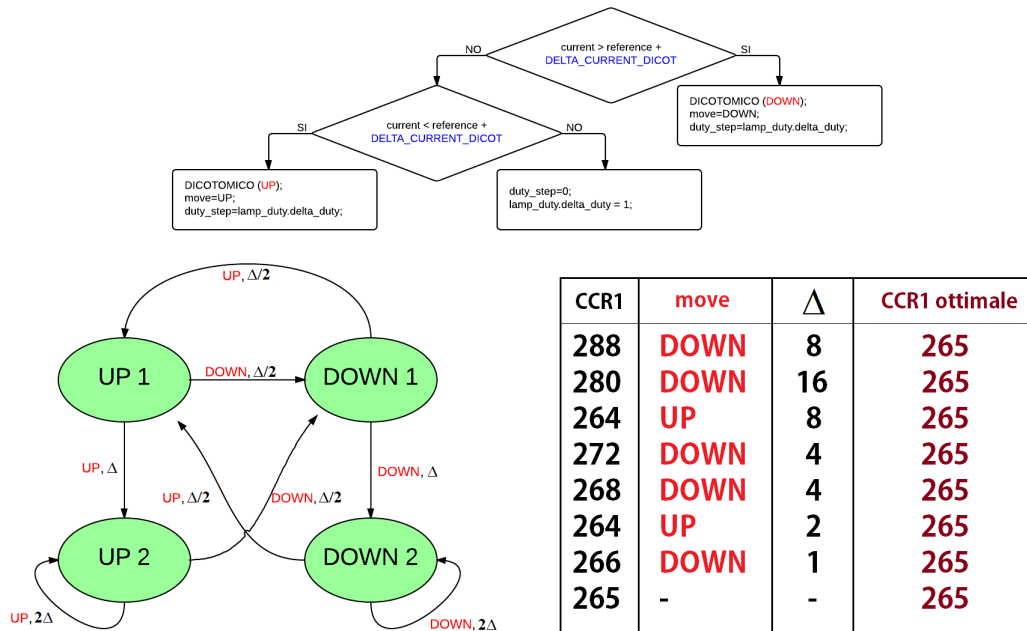


Figura 4.13: Diagramma a stati dell'algoritmo di ImproveDuty() in modalità DICOTOMICO.

6 Prove di accensione della lampada

In questo paragrafo sono riportate le immagini delle prove effettuate sul ballast progettato al variare di alcuni parametri del firmware ed anche al variare della lampada utilizzata (in un caso una lampada da 150 W, in un caso una lampada da 100 W). Certe immagini risultano “incomplete” in quanto durante la simulazione l'algoritmo è stato interrotto per eseguire calcoli e considerazioni al seguito. Tutte le prove sono state eseguite nel laboratorio di UMPI R&D con sede a Rimini che disponeva della strumentazione adeguata all'effettuare misure, calcoli e rilevamenti grafici mediante oscilloscopio oltre che alimentatori e *variac* per l'alimentazione del ballast.

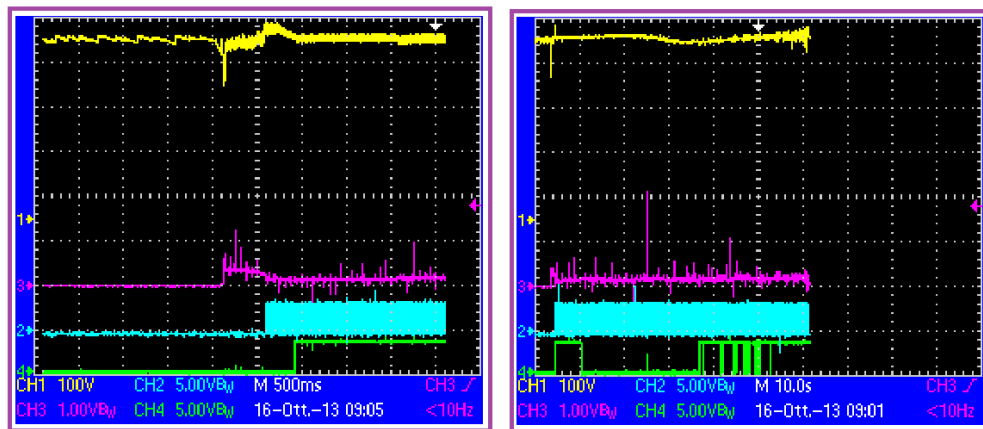


Figura 4.14: (Giallo: tensione al DC-Link; azzurro: va in toggle quando non entra in funzione la protezione e il duty cycle viene modificato dicotomicamente; rosa: corrente di lampada; verde: alto quando l'algoritmo dicotomico è UP, basso quando è DOWN). In entrambe le prove il riferimento di corrente quando si è in limitazione di corrente viene messo a 2000 mA e la protezione entra in funzione a 2500 mA. La prima prova è stata fatta con una lampada da 100 W, la seconda con una da 150 W. Nella prima immagine si può notare bene come nei secondi subito dopo l'accensione la corrente venga limitata dalla protezione e il duty cycle sia decrementato di una unità ad ogni ciclo.

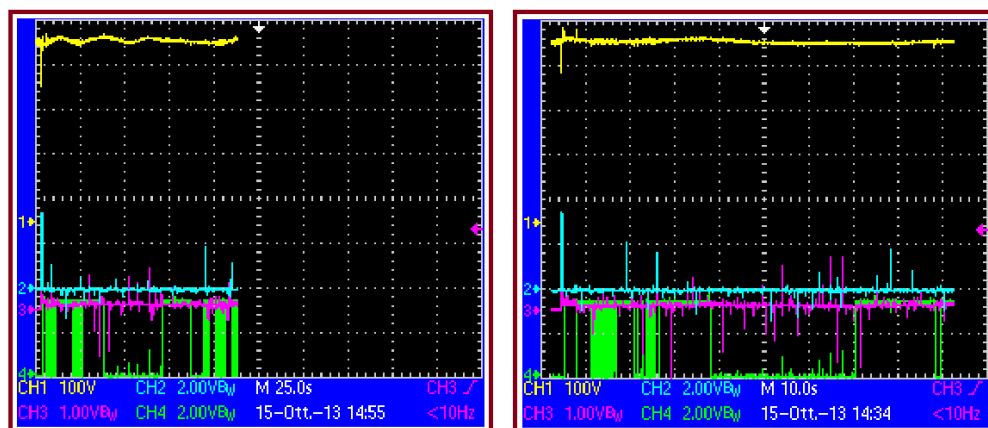


Figura 4.15: (Giallo: tensione al DC-Link; azzurro: diventa alto quando viene superata la soglia di protezione più di due volte; rosa: corrente di lampada (volt-division molto alto); verde: alto quando l'algoritmo dicotomico è UP, basso quando è DOWN). Nella prima prova è stata utilizzata una lampada da 150 W e un time-division di 25s, nella seconda la lampada è da 100 W e time-division di 10 s. Benché la lampada non sia ancora arrivata a regime la temperatura raggiunta dai MOS è di 40 °C e la potenza attinta dalla rete risulta di 75 W (lettura dal wattmetro). Visibile un leggero sfarfallio, da analizzare meglio.

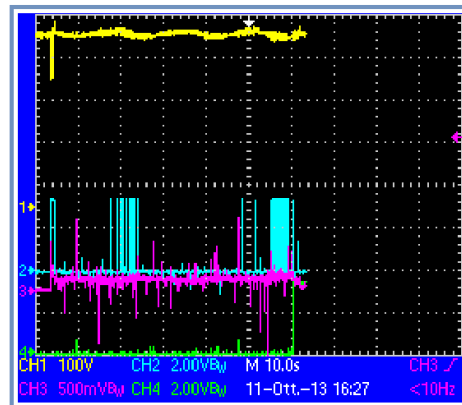


Figura 4.16: (Giallo: tensione al DC-Link; azzurro: diventa alto quando viene superata la soglia di protezione più di due volte; rosa: corrente di lampada; verde: diventa alto quando si entra in regolazione di potenza). Questa volta la regolazione del duty cycle avviene in maniera ONE_STEP. Visibile un leggero sfarfallio, da analizzare meglio.

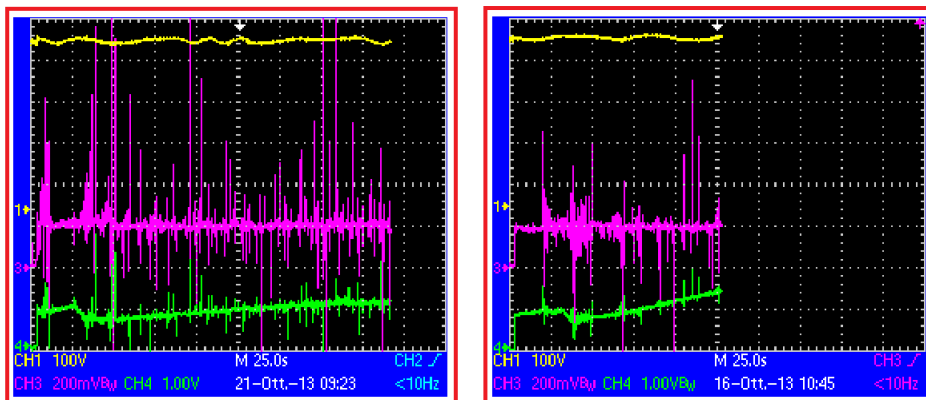


Figura 4.17: (Giallo: tensione al DC-Link; rosa: corrente di lampada; verde: tensione di lampada). La corrente risulta molto piccola e rumorosa ma si presume sia rumore raccolto dalle probe, mentre la tensione di lampada presenta un minimo di tensione a circa 30 secondi dall'accensione dell'arco: questo andamento, non documentato in letteratura, è un andamento tipico che è stato riscontrato per le lampade utilizzate e può essere utilizzato come "firma", la cui distorsione può essere di aiuto nell'identificazione del grado di usura. Nella prima prova, effettuata con lampada da 150 W la corrente viene mantenuta bene costantemente a 2000 mA e la tensione cresce abbastanza linearmente; la tensione sul DC-Link presenta piccole ondulazioni a frequenze molto basse. Nella seconda prova è stata utilizzata una lampada da 100 W mantenendo la limitazione in corrente a 2000 mA: la tensione cresce più velocemente. In entrambi i casi è visibile un leggero sfarfallio le cui cause devono essere analizzate meglio.

Conclusioni

La realizzazione di questo progetto si è sviluppata fondamentalmente in due fasi: una prima parte volta allo studio e al miglioramento di tutti i circuiti costituenti il ballast (*hardware*); una seconda fase di implementazione e stesura dell'algoritmo in grado di rispondere al meglio alle esigenze richieste dalla specifica applicazione (*firmware*).

Per ogni blocco è stato studiato il principio di funzionamento e le interazioni con le restanti parti del progetto andando a collaudare e verificare, mediante apposita strumentazione, che i segnali rispecchiassero gli andamenti previsti dalla teoria. Sono state rilevate e corrette possibili cause di malfunzionamento e aggiustate alcune non conformità esistenti nei primi prototipi realizzati.

Dopo aver verificato che l'*hardware* si comportasse nei modi previsti si è proceduto alla stesura del *firmware* eseguendo numerosi test. Il primo passo è stato quello della gestione del delicato momento dell'innesco dell'arco: una buona e decisa accensione della lampada lascia intuire che la fase di innesco dell'arco elettrico sia stata ben gestita. La successiva fase di warmup consiste sostanzialmente in un algoritmo che limita la corrente agendo sul duty cycle e avvalendosi di un circuito logico asincrono di protezione: la ricerca dicotomica implementata e precauzionalmente dimensionata affinché non si verificassero situazioni di divergenza dell'algoritmo ha prodotto una corretta limitazione della corrente.

Durante questa fase si è potuto verificare sperimentalmente una ben precisa dinamica della resistività dell'arco, non documentata in bibliografia, che potrebbe essere convenientemente sfruttata per dedurre informazioni sul grado di usura della lampada.

Dopo il raggiungimento della fase di regolazione di potenza, ma in molte

prove anche con intensità luminose consistenti, si è notato un leggero sfarfallio della luce emessa: bisogna investigare meglio l'origine, ma si ritiene che sia dovuto al fatto che l'intervento del microcontrollore per aggiustare il duty cycle avvenga troppo di rado. Rimane aperto anche il problema di una stranissima ondulazione sul DC-Link a frequenze straordinariamente basse (ordine di qualche secondo).

I test effettuati hanno mostrato come una basetta dissipativa posta sui 4 MOS del ponte sia necessaria per consentire un miglior scambio termico con l'ambiente del calore generato dai MOS (sono state toccate temperature tra gli 80 °C e i 100°C, con la lampada non ancora a regime).

Sviluppi futuri

Dal punto di vista hardware la scheda, con le modifiche che ho apportato, soddisfa pienamente le specifiche sia in condizioni nominali che in termini di sollecitazioni limite ammissibili. Rimangono però da quantificare meglio le perdite, sia sui dispositivi attivi che su quelli magnetici: questo però non può prescindere dalla frequenza di commutazione e, a valle del consolidamento del firmware, l'hardware dovrà essere ulteriormente raffinato per aumentarne il rendimento e lo smaltimento del calore optando per packaging e resistenze termiche più favorevoli.

Il firmware può essere sicuramente migliorato: allo stato attuale il controllo modifica il duty cycle ad ogni inversione (ogni 6.25 ms, solo 160 volte al secondo) e cioè ogni 300 cicli di energizzazione della lampada: per avere un controllo più efficace della lampada sarebbe bene controllare la corrente sul ponte ad ogni periodo di lampada; questo probabilmente eviterebbe lo sfarfallio ma costituisce certamente un notevole sovraccarico di lavoro per il microcontrollore che dedicherebbe la maggior parte delle risorse di calcolo al solo controllo della lampada.

Un ulteriore miglioramento è quello di identificare e memorizzare la sequenza di duty cycle più adatta per gestire il cambio di polarità della corrente sugli elettrodi al fine di minimizzare lo stress ed aumentare ulteriormente la vita della lampada rispetto a quanto accade con un ballast convenzionale.

Appendice A

Firmware: myBRIDGE.c

```

// Company: UMPI R&D s.r.l. - www.umpi.it
// E-mail: umpird@umpi.it
// All rights reserved.
//-----
/* Includes ----- */
#include "application/myBRIDGE.h"
#include "application/anal_input.h"
#include "application/lamp.h"
#include "STM32/STM32_ADC_MD.h"
#include "STM32/STM32_IO.h"

/* Private typedef ----- */
typedef struct myBRIDGE_ConfigDef
{
    TIM_OCInitTypeDef TIM_OCInitStructure1;
    TIM_OCInitTypeDef TIM_OCInitStructure2;
    TIM_BDTRInitTypeDef TIM_BDTRInitStructure;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
} myBRIDGE_ConfigDef;

/* Private define ----- */
/* Private macro ----- */
/* Private function prototypes ----- */
/* Private variables ----- */

// struttura per la gestione degli ingressi/uscite
GPIO_InitTypeDef GPIO_InitStructure;
// struttura per la gestione degli interrupt
NVIC_InitTypeDef NVIC_InitStructure;
// myBRIDGE è il nome che fa riferimento alla struttura myBRIDGE_ConfigDef
myBRIDGE_ConfigDef myBRIDGE;

uint16_t myBRIDGE_PWM_Tick;
uint16_t tick_of_inversion, half_tick_of_inversion;
uint16_t myBRIDGE_TIM1CH1_Pulse;
uint16_t myBRIDGE_TIM1CH2_Pulse;
uint16_t myBRIDGE_FREQUENCY;
uint16_t PrescalerValue;
uint16_t PWM_Period;

extern uint16_t polarity;
extern uint16_t CURRENT_LIMIT_REACH;

/* Public functions ----- */
/***** */
uint16_t myBRIDGE_Init(uint16_t PWM_Frequency, uint16_t DutyCycle)
{
    // GPIO_InitStructure Configuration
    //-----
    * inizializzo le uscite come push-pull, in
    * modalità alternative function, con velocità di
    * commutazione di 50 MHz. (guardando il datasheet):
    * GPIO_Pin_8 = X1 = piedino 28 --> PWM1
    * GPIO_Pin_9 = X3 = piedino 29 --> PWM2
    * GPIO_Pin_13 = X2 = piedino 26 --> PWM1N
    * GPIO_Pin_14 = X4 = piedino 27 --> PWM2N
    //----- */
}

```

```

// myBRIDGE_FREQUENCY assume il valore che gli passo in ingresso
myBRIDGE_FREQUENCY = PWM_Frequency;

//tick of inversion è il numero di tick da raggiungere per poi cambiare pol.
tick_of_inversion = myBRIDGE_FREQUENCY/LOW_FREQUENCY;
half_tick_of_inversion = tick_of_inversion/2;

// abilito il clock di TIM1
RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1, ENABLE);

// abilito il clock nelle porte GPIOA & GPIOB
RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA | RCC_APB2Periph_GPIOB |
RCC_APB2Periph_AFIO, ENABLE);

// GPIO A Configuration
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure);

// GPIO B Configuration
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13 | GPIO_Pin_14;
GPIO_Init(GPIOB, &GPIO_InitStructure);

// TimeBaseStructure Configuration
//-----
* SystemCoreClock è la frequenza interna del chip che per il STM32F103xx è di 72
* Mhz. Per ottenere una frequenza di conteggio per il TIM1 di 24 MHz, il Prescaler
* va calcolato come segue:
* Prescaler = (SystemCoreClock / TIM1 counter clock) - 1
* Per quanto riguarda invece la frequenza il calcolo è:
* Frequency = TIM1 counter clock/(ARR + 1)
* (in realtà nel nostro caso la frequenza sarà data, quello da determinare è ARR)
//----- */

// Calcolo il valore di Prescaler
PrescalerValue = (uint16_t) (SystemCoreClock / 24000000) - 1;
// myBRIDGE_PWM_Tick è la variabile che conteggia quanti istanti far passare
di cambiare polarità e viene settato a 0 all'inizializzazione
myBRIDGE_PWM_Tick = 0;
// configurazione del Timer TIM1
myBRIDGE.TIM_TimeBaseStructure.TIM_Period = (uint16_t)(24000000/PWM_Frequency*2));
// il *2 al denominatore è perchè stiamo facciamo un conteggio center aligned
myBRIDGE.TIM_TimeBaseStructure.TIM_Prescaler = PrescalerValue;
myBRIDGE.TIM_TimeBaseStructure.TIM_ClockDivision = 0;
// imposto il conteggio CenterAligned3 perchè genera il bit di interrupt per ogni
underflow e overflow event
myBRIDGE.TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_CenterAligned3;
myBRIDGE.TIM_TimeBaseStructure.
TIM_RepetitionCounter = 0;
TIM_TimeBaseInit(TIM1,
&myBRIDGE.TIM_TimeBaseStructure);
// abilito il conteggio per TIM1
TIM_Cmd(TIM1, ENABLE);

```



```

myBRIDGE.TIM_OCInitStruct2.TIM_OutputNState = TIM_OutputNState_Enable;
TIM_OC2Init(TIM1, &myBRIDGE.TIM_OCInitStruct2);

myBRIDGE.TIM1CH1_Pulse = TIM1->CCR1;
myBRIDGE.TIM1CH2_Pulse = TIM1->CCR2;
PWM_Period = myBRIDGE.TIM_TimeBaseStructure.TIM_Period;
// all'inizializzazione tengo disabilitate le uscite PWM
TIM_CtrlPWMOutputs(TIM1, DISABLE);

// NVIC_InitStructure Configuration
// abilito NVIC per la gestione degli interrupt
NVIC_InitStructure.NVIC_IRQChannel = TIM1_UP_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelSubPriority=1;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);

// Abilito il preload su OC1 e OC2 della periferica TIM1
TIM_OC1PreloadConfig(TIM1, TIM_OCPreload_Enable);
TIM_OC2PreloadConfig(TIM1, TIM_OCPreload_Enable);

// Abilito il registro preload su ARR della periferica TIM1
TIM_ARRPreloadConfig(TIM1, ENABLE);

// per ora tengo disabilitato l'interrupt di TIM1
TIM_ITConfig(TIM1, TIM_IT_Update, DISABLE);

TIM_ITConfig(TIM1, TIM_IT_CC1, DISABLE);
TIM_ITConfig(TIM1, TIM_IT_CC2, DISABLE);
TIM_ITConfig(TIM1, TIM_IT_CC3, DISABLE);
TIM_ITConfig(TIM1, TIM_IT_CC4, DISABLE);

return BRIDGE_ERR_NoError;
}

/***** */
int16_t myBRIDGE_Stop(void)
{
// disabilito le uscite PWM
TIM_CtrlPWMOutputs(TIM1, DISABLE);
// disabilito l'interrupt per il polarità
TIM_ITConfig(TIM1, TIM_IT_Update, DISABLE);
// azzerò il tick
myBRIDGE_PWM_Tick = 0;
return BRIDGE_ERR_NoError;
}
/***** */

```

```

// BDRInitStructure Configuration
// Off-state selection of run mode disabilitato
myBRIDGE.TIM_BDRInitStructure.TIM_OSSRState = TIM_OSSRState_Enable;
// Off-state selection of idle mode disabilitato
myBRIDGE.TIM_BDRInitStructure.TIM_OSSIState = TIM_OSSIState_Enable;
// protezione contro errori software disabilitato
myBRIDGE.TIM_BDRInitStructure.TIM_LOCKLevel = TIM_LOCKLevel_OFF;
// imposto il dead-time: trovo il valore di conteggio al quale deve arrivare il
contatore che gestisce il dead time. Per farlo basta moltiplicare
DEADBAND_TIME_NANOSec per la frequenza di sistema, cioè 72 MHz. Scegliendo
DEADBAND_TIME_NANOSec = 100 ns il contatore del dead-time si ferma a 7.
myBRIDGE.TIM_BDRInitStructure.TIM_DeadTime = (uint16_t)
(DEADBAND_TIME_NANOSec * SystemCoreClock);
// break disabilitato
myBRIDGE.TIM_BDRInitStructure.TIM_Break = TIM_Break_Disable;
//polarità bassa del break
myBRIDGE.TIM_BDRInitStructure.TIM_BreakPolarity = TIM_BreakPolarity_Low;
myBRIDGE.TIM_BDRInitStructure.TIM_AutomaticOutput = TIM_AutomaticOutput_Disable;
TIM_BDRConfig(TIM1, &myBRIDGE.TIM_BDRInitStructure);

// ocInitStruct Configuration
/*-----*/
* Channel 1, Channel 1N, Channel 2, Channel 2N
* Configuration in PWM1 mode: quando il conteggio è UP il canale 1 è attivo
* (OC1REF="1") fin tanto che TIMx_CNT<TIMx_CCR1, altrimenti è inattivo
* (OC1REF="0"). Quando il conteggio è DOWN il canale è inattivo fin
* tanto che TIMx_CNT>TIMx_CCR1, altrimenti diventa attivo
----- */
myBRIDGE.TIM_OCInitStruct1.TIM_OCMode = TIM_OCMode_PWM1;
// setto il duty cycle. Il valore Pulse viene poi caricato nel CCR, quindi il
valore Pulse può assumere valori da 0x0000 a 0xFFFF.
N.B.: c'è una divisione di mezzo che può portare numeri con la virgola, ma tanto
arrotonda in automatico.
myBRIDGE.TIM_OCInitStruct1.TIM_Pulse =
(DutyCycle * myBRIDGE.TIM_TimeBaseStructure.TIM_Period)/1000;
// definisco la polarità alta di entrambe le uscite
myBRIDGE.TIM_OCInitStruct1.TIM_OCpolarity = TIM_OCpolarity_High;
myBRIDGE.TIM_OCInitStruct1.TIM_OCNPolarity = TIM_OCNPolarity_High;
myBRIDGE.TIM_OCInitStruct1.TIM_OCIdleState = TIM_OCIdleState_Reset;
myBRIDGE.TIM_OCInitStruct1.TIM_OCIdleState = TIM_OCIdleState_Reset;
myBRIDGE.TIM_OCInitStruct1.TIM_OutputState = TIM_OutputState_Enable;
myBRIDGE.TIM_OCInitStruct1.TIM_OutputNState = TIM_OutputNState_Enable;
TIM_OC1Init(TIM1, &myBRIDGE.TIM_OCInitStruct1);

// OCInitStruct2 Configuration
myBRIDGE.TIM_OCInitStruct2.TIM_OCMode = TIM_OCMode_PWM1;
// il secondo canale è sempre PWM, ma il duty cycle è complementare al primo PWM,
quindi avrò dutycycle2 = 1000 - dutycycle1
myBRIDGE.TIM_OCInitStruct2.TIM_Pulse =
myBRIDGE.TIM_TimeBaseStructure.TIM_Period-myBRIDGE.TIM_OCInitStruct1.TIM_Pulse;
myBRIDGE.TIM_OCInitStruct2.TIM_OCpolarity = TIM_OCpolarity_High;
myBRIDGE.TIM_OCInitStruct2.TIM_OCNPolarity = TIM_OCNPolarity_High;
myBRIDGE.TIM_OCInitStruct2.TIM_OCIdleState = TIM_OCIdleState_Reset;
myBRIDGE.TIM_OCInitStruct2.TIM_OCIdleState = TIM_OCIdleState_Reset;
myBRIDGE.TIM_OCInitStruct2.TIM_OCIdleState = TIM_OCIdleState_Reset;
myBRIDGE.TIM_OCInitStruct2.TIM_OutputState = TIM_OutputState_Enable;
myBRIDGE.TIM_OCInitStruct2.TIM_OutputState = TIM_OutputState_Enable;

```

```

int16_t myBRIDGE_Start(uint16_t polar)
{
    switch(polar)
    {
        case (POL_POS):
            myBRIDGE.TIM_OCInitStructel.TIM_OCMode = TIM_OCMode_PWM1;
            myBRIDGE.TIM_OCInitStructel.TIM_Pulse = myBRIDGE_TIM1CH1_Pulse;
            TIM_OCInit(TIM1, &myBRIDGE.TIM_OCInitStructel);
            myBRIDGE.TIM_OCInitStructel2.TIM_OCMode = TIM_OCMode_PWM2;
            myBRIDGE.TIM_OCInitStructel2.TIM_Pulse = myBRIDGE_TIM1CH2_Pulse;
            TIM_OC2Init(TIM1, &myBRIDGE.TIM_OCInitStructel2);

            TIM_CxxCmd(TIM1, TIM_Channel_1, TIM_CcX_Enable);
            TIM_CcXnCnd(TIM1, TIM_Channel_1, TIM_CcXN_Enable);
            TIM_CcxCmd(TIM1, TIM_Channel_2, TIM_CcX_Disable);
            TIM_CcXnCnd(TIM1, TIM_Channel_2, TIM_CcXN_Enable);
            break;

        case (POL_NEG):
            myBRIDGE.TIM_OCInitStructel.TIM_OCMode = TIM_OCMode_PWM2;
            myBRIDGE.TIM_OCInitStructel.TIM_Pulse = myBRIDGE_TIM1CH1_Pulse;
            TIM_OCInit(TIM1, &myBRIDGE.TIM_OCInitStructel);
            myBRIDGE.TIM_OCInitStructel2.TIM_OCMode = TIM_OCMode_PWM1;
            myBRIDGE.TIM_OCInitStructel2.TIM_Pulse = myBRIDGE_TIM1CH2_Pulse;
            TIM_OC2Init(TIM1, &myBRIDGE.TIM_OCInitStructel2);

            TIM_CcxCmd(TIM1, TIM_Channel_1, TIM_CcX_Disable);
            TIM_CcXnCnd(TIM1, TIM_Channel_1, TIM_CcXN_Enable);
            TIM_CcxCmd(TIM1, TIM_Channel_2, TIM_CcX_Enable);
            TIM_CcXnCnd(TIM1, TIM_Channel_2, TIM_CcXN_Enable);
            break;

        case (ALL_ACTIVE):
            myBRIDGE.TIM_OCInitStructel.TIM_OCMode = TIM_OCMode_PWM1;
            myBRIDGE.TIM_OCInitStructel.TIM_Pulse = myBRIDGE_TIM1CH1_Pulse;
            TIM_OCInit(TIM1, &myBRIDGE.TIM_OCInitStructel);
            myBRIDGE.TIM_OCInitStructel2.TIM_OCMode = TIM_OCMode_PWM1;
            myBRIDGE.TIM_OCInitStructel2.TIM_Pulse = myBRIDGE_TIM1CH2_Pulse;
            TIM_OC2Init(TIM1, &myBRIDGE.TIM_OCInitStructel2);

            TIM_CcxCmd(TIM1, TIM_Channel_1, TIM_CcX_Enable);
            TIM_CcXnCnd(TIM1, TIM_Channel_1, TIM_CcXN_Enable);
            TIM_CcxCmd(TIM1, TIM_Channel_2, TIM_CcX_Enable);
            TIM_CcXnCnd(TIM1, TIM_Channel_2, TIM_CcXN_Enable);
            break;
    }

    // abilito le uscite PWM
    TIM_CtrlPWMOutputs(TIM1, ENABLE);
    myBRIDGE_PWM_Tick = 0;
}

-----
// risetto polarity guardando CCR1 e CCR2
if(TIM1->CCR1 > TIM1->CCR2)
{
    polarity = 1;
}
else
{
    polarity = 0;
}
-----
return BRIDGE_ERR_NoError;
}

-----
/*****
int16_t myBRIDGE_ChangedDuty(uint8_t byState, uint16_t step)
{
    // N.B.: step è il valore di tempi di clock da aggiungere/togliere al PWM1 e PWM2.
    Ricordandosi che se step è 5, i clock sulla lampada sono 10.
    uint16_t out;

    switch (byState)
    {
        case BRIDGE_DUTY_UP:
            if (polarity == 1)
            {
                // energizzazione +1
                TIM1->CCR1 = (TIM1->CCR1) + step;
                TIM1->CCR2 = (TIM1->CCR2) - step;
                out = BRIDGE_ERR_NoError;
                if (TIM1->CCR1 >= PWM_Period)
                {
                    TIM1->CCR1=(PWM_Period)-1;
                    TIM1->CCR2=1;
                    out = ERR_CCR1_OVERFLOW;
                }
            }
            else
            {
                // energizzazione -1
                TIM1->CCR2 = (TIM1->CCR2) + step;
                TIM1->CCR1 = (TIM1->CCR1) - step;
                out = BRIDGE_ERR_NoError;
                if (TIM1->CCR2 >= PWM_Period)
                {
                    TIM1->CCR2=(PWM_Period)-1;
                    TIM1->CCR1=1;
                    out = ERR_CCR2_OVERFLOW;
                }
            }
        case BRIDGE_DUTY_DOWN:
            if (polarity == 1)
            {
                // energizzazione +1
                TIM1->CCR1 = (TIM1->CCR1) - step;
                TIM1->CCR2 = (TIM1->CCR2) + step;
                out = BRIDGE_ERR_NoError;
                if (TIM1->CCR1 <= 0)
                {
                    TIM1->CCR1=(PWM_Period)-1;
                    TIM1->CCR2=1;
                    out = ERR_CCR1_OVERFLOW;
                }
            }
            else
            {
                // energizzazione -1
                TIM1->CCR2 = (TIM1->CCR2) - step;
                TIM1->CCR1 = (TIM1->CCR1) + step;
                out = BRIDGE_ERR_NoError;
                if (TIM1->CCR2 <= 0)
                {
                    TIM1->CCR2=(PWM_Period)-1;
                    TIM1->CCR1=1;
                    out = ERR_CCR2_OVERFLOW;
                }
            }
    }
}
-----
}

```

```

{
    TIM1->CCR1=(PWM_Period/2)+1;
    TIM1->CCR2=(PWM_Period/2)-1;
    out = ERR_CCR2_OVERFLOW;
}

}
else
{
    // energizzazione -1
    TIM1->CCR2 = (TIM1->CCR2) - step;
    TIM1->CCR1 = (TIM1->CCR1) + step;
    out = BRIDGE_ERR_NoError;
    if (TIM1->CCR2 <= (PWM_Period/2)+1)
    {
        TIM1->CCR1= (PWM_Period/2)-1;
        TIM1->CCR2= (PWM_Period/2)+1;
        out = ERR_CCR1_OVERFLOW;
    }
}
break;
}
}

//-----
// limito il duty di lampada ad assumere valori a l massimo del 40%
// CCR1 = 350 & CCR2 = 150 --> duty_lamp = +40%
// CCR2 = 350 & CCR1 = 150 --> duty_lamp = -40%
if (TIM1->CCR1 > 350)
{
    TIM1->CCR1=350;
    TIM1->CCR2=150;
}
}

if (TIM1->CCR2 > 350)
{
    TIM1->CCR2=350;
    TIM1->CCR1=150;
}
//-----
// risetto polarity guardando CCR1 e CCR2
if (TIM1->CCR1 > TIM1->CCR2)
{
    polarity = 1;
}
else
{
    polarity = 0;
}
//-----
myBRIDGE_TIM1CH1_Pulse = TIM1->CCR1;
myBRIDGE_TIM1CH2_Pulse = TIM1->CCR2;
return out;
}
}

```

```

/***** myBRIDGE_ChangePolarity (void) *****/
int16_t myBRIDGE_ChangePolarity (void)
{
    // introduco una variabile ausiliaria dummy alla quale passo il valore di CCR1
    uint16_t dummy = TIM1->CCR1;
    // setto il valore di CCR1 con il valore di CCR2
    TIM1->CCR1 = TIM1->CCR2;
    // setto il valore di CCR2 con il valore di dummy, cioè CCR1
    TIM1->CCR2 = dummy;
    //-----
    // risetto polarity guardando CCR1 e CCR2
    if (TIM1->CCR1 > TIM1->CCR2)
    {
        polarity = 1;
    }
    else
    {
        polarity = 0;
    }
    //-----
    myBRIDGE_TIM1CH1_Pulse = TIM1->CCR1;
    myBRIDGE_TIM1CH2_Pulse = TIM1->CCR2;
    return BRIDGE_ERR_NoError;
}
/***** myBRIDGE_SetDuty (int16_t NewDuty) *****/
void myBRIDGE_SetDuty (int16_t NewDuty)
{
    // N.B.: NewDuty è il valore del duty (espresso in millesimi) passato a PWM1,
    quello del PWM2 è calcolato di conseguenza. Se NewDuty è maggiore di 500 ho
    energizzazione +1. Se invece è minore di 500 ho energizzazione -1. (vedi stesso
    discorso myBRIDGE_Init()). Controllo in caso di immissione di valori sbagliati e
    in questo caso viene impostato il valore di PWM1 a DUTY_FAILURE (50%).

    if (NewDuty>1000 | NewDuty<0)
    {
        TIM1->CCR1 = myBRIDGE.TIM_TimeBaseStructure.TIM_Period/2;
        TIM1->CCR2 = myBRIDGE.TIM_TimeBaseStructure.TIM_Period-(TIM1->CCR1);
    }
    else
    {
        TIM1->CCR1 = (NewDuty * myBRIDGE.TIM_TimeBaseStructure.TIM_Period) /
        1000;
        TIM1->CCR2 = myBRIDGE.TIM_TimeBaseStructure.TIM_Period - (TIM1->CCR1);
    }
    //-----
    // risetto polarity guardando CCR1 e CCR2
    if (TIM1->CCR1 > TIM1->CCR2)
    {
        polarity = 1;
    }
    else
    {
        polarity = 0;
    }
    //-----
    myBRIDGE_TIM1CH1_Pulse = TIM1->CCR1;
    myBRIDGE_TIM1CH2_Pulse = TIM1->CCR2;
}
}

```

```

}
/*****
void TIM1_UP_IRQHandler (void)
{
    if (TIM_GetITStatus(TIM1, TIM_IT_Update) != RESET)
    {
        TIM_ClearITPendingBit(TIM1, TIM_IT_Update);
        myBRIDGE_PWM_Tick++;

        // ----- TICK = 150 -----
        if (myBRIDGE_PWM_Tick == half_tick_of_inversion)
        {
            lamp_update_value();
            CURRENT_LIMIT_REACH=0;
        }

        // ----- TICK = 300 -----
        if (myBRIDGE_PWM_Tick == tick_of_inversion)
        {
            lamp_freezing_value(CURRENT_LIMIT_REACH);
            myBRIDGE_PWM_Tick = 0;
            CURRENT_LIMIT_REACH=0;
            myBRIDGE_ChangePolarity();
        }
    }
}
/*****
uint16_t Get_TIMICH1_Pulse(void)
{
    float var;
    var=(myBRIDGE_TIMICH1_Pulse*1000)/PWM_Period;
    return var;
    // ritorna il valore del duty di PWM1 espresso in millesimi
}
/*****
uint16_t Get_TIMICH2_Pulse(void)
{
    float var;
    var=(myBRIDGE_TIMICH2_Pulse*1000)/PWM_Period;
    return var;
    // ritorna il valore del duty di PWM2 espresso in millesimi
}
/*****
uint16_t Get_lamp_Pulse(void)
{
    uint16_t var;
    if (TIM1->CCR1 > TIM1->CCR2)
    {
        var=((uint32_t)(myBRIDGE_TIMICH1_Pulse-
            myBRIDGE_TIMICH2_Pulse)*1000)/PWM_Period;
    }
    else
    {
        var=((uint32_t)(myBRIDGE_TIMICH2_Pulse-
            myBRIDGE_TIMICH1_Pulse)*1000)/PWM_Period;
    }
    return var;
    // Ritorna il valore del DUTY DI LAMPADA espresso in millesimi
}

```

Appendice B

Firmware: `lamp.c`

```

// Company:  UMPI R&D s.r.l. - www.umpi.it
// E-mail:  umpird@umpi.it
//
// All rights reserved.
//
//-----
/* Includes -----
#include "application/lamp.h"
#include "application/anal_input.h"
#include "application/gpio.h"
#include "application/myBRIDGE.h"
#include "STM32/STM32_IO.h"
#include "system/Kernel.h"
#include "inizializza.h"

/* Private typedef -----
/* Private define -----
/* Private macro -----
/* Private variables -----

DWORD time_now;
DWORD start_ignition, start_warmup_current, start_warmup_power, time_array_steady;
DWORD time_ignition, time_CurrentLimit, time_PowerRegol;

uint16_t polarity, count;
uint16_t move, duty_step, duty_value;
uint16_t vlamp_array[VLAMP_ARRAY_LENGTH];
uint16_t plamp_array[PLAMP_ARRAY_LENGTH];
uint16_t ilamp_array[ILAMP_ARRAY_LENGTH];
uint16_t dutylamp_array[DUTYLAMP_ARRAY_LENGTH];
uint16_t array_steady[ARRAY_STEADY_LENGTH];
BOOL POWER_CALCULATE,vrete_NO_OK, FIRST_POWER_EXCEED, START_STEADY, UPDATE_ENABLE;
uint16_t err_vcheck, err_ign, err_wmp;
uint16_t vlamp_array_pos, plamp_array_pos, ilamp_array_pos, dutylamp_array_pos,
uint16_t array_steady_pos;

VOLTCHECK_STRUCT volt_check;
LAMP_STRUCT lamp;
ILAMP_DUTY_STRUCT lamp_duty;

/* Public functions -----
/*****
void volt_check_Init(void)
{
    volt_check.vrete_value=0;
    volt_check.vbus_value=0;
    volt_check.vrete_tent=0;
    volt_check.vbus_tent=0;
    volt_check.voltageOK=FALSE;
    volt_check.state=VRETE_CHECK;
}
//-----
void lamp_Init(void)
{
    uint16_t randVarADC;
    err_vcheck=0;
    err_ign=0;
}
/*****
err_wmp=0;
lamp.current=0;
lamp.voltage=0;
lamp.attempt=0;
lamp.power=0;
lamp.current_read=0;
lamp.current_avg=0;
lamp.voltage_avg=0;
lamp.current_max=3000;
lamp.current_reference=CURRENT_REFERENCE;
lamp.state=LAMP_ENABLE;

lamp_duty.delta_duty=DELTA_DUTY_START;
lamp_duty.direction=0;
lamp_duty.enable=0;

move=0;
duty_step=0;
//-----ARRAY VLAMP-----
for (vlamp_array_pos=0; vlamp_array_pos<VLAMP_ARRAY_LENGTH; vlamp_array_pos++)
{
    vlamp_array[vlamp_array_pos]=0;
}
vlamp_array_pos=0;
//-----ARRAY PLAMP-----
for (plamp_array_pos=0; plamp_array_pos<PLAMP_ARRAY_LENGTH; plamp_array_pos++)
{
    plamp_array[plamp_array_pos]=0;
}
plamp_array_pos=0;
//-----ARRAY ILAMP-----
for (ilamp_array_pos=0; ilamp_array_pos<ILAMP_ARRAY_LENGTH; ilamp_array_pos++)
{
    ilamp_array[ilamp_array_pos]=0;
}
ilamp_array_pos=0;
//-----ARRAY DUTYLAMP-----
for (dutylamp_array_pos=0; dutylamp_array_pos<DUTYLAMP_ARRAY_LENGTH;
dutylamp_array_pos++)
{
    dutylamp_array[dutylamp_array_pos]=0;
}
dutylamp_array_pos=0;
//-----ARRAY STEADY-----
for (array_steady_pos=0; array_steady_pos<ARRAY_STEADY_LENGTH; array_steady_pos++)
{
    array_steady[array_steady_pos]=0;
}
array_steady_pos=0;

FIRST_POWER_EXCEED=TRUE;
START_STEADY=FALSE;
UPDATE_ENABLE=FALSE;
POWER_CALCULATE=FALSE;

randVarADC = anal_input_get_vrete();
polarity = randVarADC%2;
}
/*****

```

```

/*****
void voltage_check(void)
{
    switch (volt_check.state)
    {
        case VRETE_CHECK:
            volt_check.vrete_value = anal_input_get_vrete();
            if (volt_check.vrete_value > VRETE_OK)
            {
                // Abilita il PFC Controller
                STM32_IO_PortInit(PFC_EN, IO_TYPE_OUT_PUSH_PULL);
                STM32_IO_Set(PFC_EN, 0);
                volt_check.state = VBUS_CHECK;
            }
            else
            {
                volt_check.vrete_tent++;
                if (volt_check.vrete_tent > MAX_NUM_TENT)
                {
                    vrete_NO_OK = TRUE;
                    volt_check.state = VOLTAGE_FAILURE;
                }
                Delay_ms(1);
                err_vcheck = vrete_check_wait;
            }
            break;

        case VBUS_CHECK:
            volt_check.vbus_value = anal_input_get_vbus();
            if (volt_check.vbus_value > VBUS_OK)
            {
                // voltege_check andato bene
                volt_check.state = VOLTAGE_OK;
            }
            else
            {
                volt_check.vbus_tent++;
                if (volt_check.vbus_tent > MAX_NUM_TENT)
                {
                    volt_check.state = VOLTAGE_FAILURE;
                }
                Delay_ms(1);
                err_vcheck = vbus_check_wait;
            }
            break;

        case VOLTAGE_OK:
            volt_check.voltageOK = TRUE;
            err_vcheck = voltage_check_NO_ERR;
            break;

        case VOLTAGE_FAILURE:
            if (vrete_NO_OK == TRUE)
            {
                err_vcheck = vrete_check_ERR;
            }
            else
            {
                err_vcheck = vbus_check_ERR;
            }
    }
}
/*****/

break;
}
}
//-----
void lamp_ignition(void)
{
    switch (lamp.state)
    {
        case LAMP_ENABLE:
            start_ignition = GetSystemTick();
            if (volt_check.voltageOK == TRUE)
            {
                // do un duty di lampada del 50% --> 1/2 di Vcd-link per caricare
                // le capacità (carico e boosttrap)
                if (polarity == 0)
                {
                    myBRIDGE_SetDuty(250);
                }
                else
                {
                    myBRIDGE_SetDuty(750);
                }
            }
            myBRIDGE_Start(ALL_ACTIVE);
            TIM_ITConfig(TIM1, TIM_IT_Update, DISABLE);
            lamp.state = LAMP_CHARGE_COND;
        }
        break;

        case LAMP_CHARGE_COND:
            lamp.voltage = anal_input_get_vlamp();
            if (lamp.voltage >= CHARGE_COND_OK)
            {
                lamp.state = LAMP_START;
            }
            err_ign = CHARGE_COND;
            break;

        case LAMP_START:
            // Polarizzazione testa/croce del ponte
            if (polarity == 0)
            {
                // polarizzazione negativa
                myBRIDGE_SetDuty(DUTY_START_NEG);
                myBRIDGE_Start(POL_NEG);
            }
            else
            {
                // polarizzazione positiva
                myBRIDGE_SetDuty(DUTY_START_POS);
                myBRIDGE_Start(POL_POS);
            }
            TIM_ITConfig(TIM1, TIM_IT_Update, DISABLE);
            time_now = GetSystemTick();
            lamp.state = LAMP_START_CHECK;
            break;
    }
}

```

```

case LAMP_POWER_CALCULATE:
    if (POWER_CALCULATE)
        err_wmp = STATE_POWER_CALC;
    {
        POWER_CALCULATE=FALSE;
        lamp_current_avg = Get_lamp_current();
        lamp_voltage_avg = Get_lamp_voltage();
        lamp_power = ((uint32_t) lamp_current_avg * lamp_voltage_avg) / 1000;
        if (lamp_power >= POWER_STEADY_VALUE+1 ||
            lamp_power <= POWER_STEADY_VALUE-1)
        {
            lamp_current_reference = POWER_STEADY_VALUE / lamp_voltage;
            if (lamp_current_reference > CURRENT_REFERENCE)
            {
                lamp_current_reference = CURRENT_REFERENCE;
            }
        }
        if (FIRST_POWER_EXCEED == TRUE)
        {
            if (lamp_power > POWER_STEADY_VALUE)
            {
                FIRST_POWER_EXCEED=FALSE;
                time_CurrentLimit=KERNEL_elapsedTick
                (start_warmup_current);
                start_warmup_power = GetSystemTick();
            }
        }
        if (START_STEADY == FALSE)
        {
            if (KERNEL_Delay_elapsed
                (time_array_steady, TIME_UPDATE_ARRAY_STEADY))
            {
                time_array_steady = GetSystemTick();
                array_steady_update();
            }
        }
        lamp_state = LAMP_WAIT;
        break;
    }
}

//-----
void ImproveDuty (uint8_t value)
{
    switch (value)
    {
        case DICOTOMICO:
            UPDATE_ENABLE = TRUE;
            //-- calcolo come modificare il duty --
            if (lamp_current < lamp_current_reference + DELTA_CURRENT_DICOT)
            {
                if (lamp_current > lamp_current_reference - DELTA_CURRENT_DICOT)
                {
                    duty_step = 0;
                    lamp_duty.delta_duty = 1;
                }
                else
            }
        }
    }
}

```

-VII-

```

    {
        ImproveDutyDicotomico (UP);
        move = UP;
        duty_step = lamp_duty.delta_duty;
    }
}
else
{
    ImproveDutyDicotomico (DOWN);
    move = DOWN;
    duty_step = lamp_duty.delta_duty;
}
break;

case ONE_STEP:
    UPDATE_ENABLE = TRUE;
    //-- calcolo come modificare il duty --
    if (lamp_current < lamp_current_reference + DELTA_CURRENT_1STEP)
    {
        if (lamp_current < lamp_current_reference - DELTA_CURRENT_1STEP)
        {
            move = UP;
            duty_step = 1;
        }
        else
        {
            move = DOWN;
            duty_step = 0;
        }
    }
}
else
{
    move = DOWN;
    duty_step = 1;
}
break;
}

//-----
void ImproveDutyDicotomico (uint16_t versus)
{
    switch (versus)
    {
        case DOWN:
            if (lamp_duty.direction == 1)
            {
                lamp_duty.delta_duty =
                lamp_duty.delta_duty - (lamp_duty.delta_duty/2);
                lamp_duty.enable = 0;
            }
            else
            {
                if (lamp_duty.enable == 1)
                {
                    lamp_duty.delta_duty =
                    lamp_duty.delta_duty + lamp_duty.delta_duty;
                    lamp_duty.enable = 1;
                }
                lamp_duty.direction = 0;
            }
        }
    }
}

```

-VIII-


```

err_wmp = D_DOWN;
break;

case UP:
    if(lamp_duty.direction == 0)
    {
        lamp_duty.delta_duty =
        lamp_duty.delta_duty - (lamp_duty.delta_duty/2);
        lamp_duty.enable = 0;
    }
    else
    {
        if(lamp_duty.enable == 1)
        {
            lamp_duty.delta_duty =
            lamp_duty.delta_duty + lamp_duty.delta_duty;
            lamp_duty.enable = 1;
        }
        lamp_duty.direction = 1;
        err_wmp = D_UP;
        break;
    }

    // metto un limite inferiore a delta_duty per far si che non si verifichino
    valori negativi
    if(lamp_duty.delta_duty < 1)
    {
        lamp_duty.delta_duty = 1;
    }

    // metto un limite superiore a delta_duty altrimenti il duty rischia di divergere
    limite imposto a DELTA_DUTY_LIMIT = 32 --> corrisponde a incrementi/decrementi
    del duty del 12.8%
    if(lamp_duty.delta_duty > DELTA_DUTY_LIMIT)
    {
        lamp_duty.delta_duty = DELTA_DUTY_LIMIT;
    }

    //-----
void lamp_update_value(void)
{
    if(UPDATE_ENABLE == TRUE)
    {
        UPDATE_ENABLE = FALSE;
        myBRIDGE_ChangeDuty(move,duty_step);
    }
}
//-----
void lamp_freezing_value(uint16_t num)
{
    count = num;
    lamp_voltage = anal_input_get_vlamp();
    lamp_current_read = anal_input_get_ilamp();
    lamp_state = LAMP_ELABORATION;
}
/*****
/*****/

```

```

void array_steady_update(void)
{
    float var;
    int i;
    var=0;
    array_steady[array_steady_pos]=lamp_voltage;
    array_steady_pos++;
    if(array_steady_pos==ARRAY_STEADY_LENGTH)
    {
        array_steady_pos=0;
    }
    for (i=0; i<ARRAY_STEADY_LENGTH;i++)
    {
        var=var+array_steady[i];
    }
    var=var/ARRAY_STEADY_LENGTH;
    if(lamp_voltage==var)
    {
        time_PowerRegol=KERNEL_elapsedTick(start_warmup_power);
        START_STEADY=TRUE;
    }
}
//-----
void vlamp_array_update(void)
{
    vlamp_array[vlamp_array_pos] = lamp_voltage;
    vlamp_array_pos++;
    if(vlamp_array_pos == VLAMP_ARRAY_LENGTH)
    {
        vlamp_array_pos = 0;
        POWER_CALCULATE = TRUE;
    }
}
//-----
void ilamp_array_update(void)
{
    ilamp_array[ilamp_array_pos] = lamp_current;
    ilamp_array_pos++;
    if(ilamp_array_pos == ILAMP_ARRAY_LENGTH)
    {
        ilamp_array_pos = 0;
        POWER_CALCULATE = TRUE;
    }
}
//-----
void plamp_array_update(void)
{
    plamp_array[plamp_array_pos] = (lamp_voltage*lamp_current)/1000;
    plamp_array_pos++;
    if(plamp_array_pos == PLAMP_ARRAY_LENGTH)
    {
        plamp_array_pos = 0;
    }
}
//-----

```

```

void dutylamp_array_update (void)
{
    dutylamp_array[dutylamp_array_pos] = lamp.duty;
    dutylamp_array_pos++;
    if(dutylamp_array_pos == DUTYLAMP_ARRAY_LENGTH)
    {
        dutylamp_array_pos = 0;
    }
}
/*****
/*****
WORD Get_lamp_voltage (void)
{
    float var;
    int i;
    var = 0;
    for (i=0; i<VLAMP_ARRAY_LENGTH; i++)
    {
        var = var+vlamp_array[i];
    }
    var = var/VLAMP_ARRAY_LENGTH;
    return var;
}
/*****
/*****
WORD Get_lamp_current (void)
{
    float var;
    int i;
    var = 0;
    for (i=0; i<ILAMP_ARRAY_LENGTH; i++)
    {
        var = var+ilamp_array[i];
    }
    var = var/ILAMP_ARRAY_LENGTH;
    return var;
}
/*****
/*****
WORD Get_lamp_power (void)
{
    float var;
    int i;
    var = 0;
    for (i=0; i<PLAMP_ARRAY_LENGTH; i++)
    {
        var = var+plamp_array[i];
    }
    var = var/PLAMP_ARRAY_LENGTH;
    return var;
}
/*****
/*****
DWORD Get_time_Ignition (void)
{
    return time_Ignition;
}
/*****
/*****
DWORD Get_time_CurrentLimit (void)
{
    return time_CurrentLimit;
}
/*****
/*****
DWORD Get_time_PowerRegol (void)
{
    return time_PowerRegol;
}
/*****
/*****
WORD Get_polarity (void)
{
    return polarity;
}
/*****
/*****
WORD Get_lamp_status (uint16_t variable)
{
    WORD err;
    switch (variable)
    {
        case V_CHECK:
            err=err_vcheck;
            break;
        case IGNITION:
            err=err_ign;
            break;
        case WARMUP:
            err=err_wmp;
            break;
    }
    return err;
}
/*****
/*****

```

```

case LAMP_START_CHECK:
    lamp_voltage = anal_input_get_vlamp();
    if (lamp_voltage <= voltage_ON)
    {
        lamp.state = LAMP_ON;
        time_ignition = KERNEL_elapsedTick(start_ignition);
    }
    if (KERNEL_Delay_elapsed(time_now, TIME_CHECK))
    {
        lamp.attempt++;
        if (lamp.attempt > MAX_ATTEMPT_START)
        {
            lamp.state = LAMP_BURNT_OUT;
        }
        else
        {
            time_now = GetSystemTick();
            lamp.state = LAMP_WAIT_30SEC;
        }
    }
    err_ign = START_CHECK;
    break;

case LAMP_WAIT_30SEC:
    if (KERNEL_Delay_elapsed(time_now, TIME_STANDBY))
    {
        // attivo il ponte con polarizzazione inversa
        if (polarity == 0)
        {
            myBRIDGE_SetDuty(750);
        }
        else
        {
            myBRIDGE_SetDuty(250);
        }
    }
    myBRIDGE_Start(ALL_ACTIVE);
    lamp.state = LAMP_CHARGE_COND;
}
else
{
    myBRIDGE_Stop();
}
err_ign = WAIT_30SEC;
break;

case LAMP_BURNT_OUT:
    myBRIDGE_Stop();
    err_ign = BURNT_OUT;
    break;
}
}
//

```

-V-

```

void lamp_warmup(void)
{
    switch (lamp.state)
    {
        case LAMP_ON:
            err_ign = ignition_NO_ERR;
            Set_lampReference(2200);
            if (polarity == 1)
            {
                myBRIDGE_SetDuty(DUTY_LAMP_ON);
            }
            else
            {
                myBRIDGE_SetDuty(1000-DUTY_LAMP_ON);
            }
            myBRIDGE_Start(ALL_ACTIVE);
            TIM_ITConfig(TIM1, TIM_IT_Update, ENABLE);
            start_warmup_current = GetSystemTick();
            time_array_steady = GetSystemTick();
            lamp.state = LAMP_WAIT;
            err_wmp = STATE_ON;
            break;
        case LAMP_WAIT:
            err_wmp = STATE_WAIT;
            break;
        case LAMP_ELABORATION:
            err_wmp = STATE_ELABORATION;
            if (count < 2)
            {
                lamp.duty = Get_lamp_Pulse();
                lamp.current = ((float)lamp.current_read/lamp.duty)*1000.0;
                vlamp_array_update();
                llamp_array_update();
                ImproveDuty(DICOTOMICO);
            }
            else
            {
                move=DOWN;
                duty_step=1;
                UPDATE_ENABLE = TRUE;
            }
            lamp.state = LAMP_POWER_CALCULATE;
            break;
    }
}

```

-VI-

Bibliografia

- [1] "Lampade a scarica di gas"
<http://www.arch.unige.it/did/l1/disegnoind/secondonau0203/fisica/dispense/sorgenti3.pdf>

- [2] "HID Ballast Application"
http://www.gelighting.com/na/business_lighting/education_resources/literature_library/ballast/downloads/198217_HID_Ballast_Application_Guide.pdf

- [3] IEEE International Symposium,
C. Moo S. Tang J. Chen, K. Ng and C. Lee.
"Accelerating startup of metal halide lamps. Industrial Electronics". 2006

- [4] "Metal Halide Lamps", OSRAM
http://www.osram.com/global/pdf/Professional/General_Lighting/High_pressure_discharge_lamps/106T020GB.pdf

- [5] Application Note AN3112, STMicroelectronics.
"Solution for designing a fixed off-time controlled PFC pre-regulator with the L6564". February 2011

- [6] Application Note AN2761, STMicroelectronics.
"Solution for designing a transition mode PFC preregulator with the L6562A". November 2009

- [7] Datasheet L6564, STMicroelectronics.
"10 pin transition-mode PFC controller". October, 2011

- [8] Datasheet FAN7382, Fairchild Semiconductor.
"High- and Low-Side Gate Driver". February, 2007

- [9] Datasheet TEA1721AT, NXP Semiconductors.
"HV start-up flyback controller with integrated MOSFET for 5 W applications, 430 Hz burst frequency". June, 2012
- [10] Reference Manual RM0008, STMicroelectronics.
"STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM-based 32-bit MCUs". October 2011
- [11] Application Note AN2747, STMicroelectronics.
"250 W HID metal halide electronic ballast". May 2008
- [12] Application Note AN2952, STMicroelectronics.
"35 W electronic ballast for HID lamps". September 2009
- [13] Application Note AN3159, STMicroelectronics.
"STEVAL-ILH005V2: 150 W HID electronic ballast". April 2011
- [14] Application Note AN3009, STMicroelectronics.
"How to design a transition mode PFC pre-regulator using the L6564". February 2011
- [15] Application Note AN6076, Fairchild Semiconductor.
"Design and Application Guide of Bootstrap Circuit for High-Voltage Gate-Drive IC". September 2008
- [16] Application Note AN11060, NXP Semiconductors.
"TEA172X 5 W to 11 W Power Supply/USB charger". June 2012
- [17] https://it.wikipedia.org/wiki/Ionizzazione_dei_gas
- [18] <http://cronologia.leonardo.it/invenzio/inv007.htm>
- [19] http://www.osram.it/osram_it/prodotti/lampade/lampade-a-scarica-ad-alta-intensita/lampade-ad-alta-pressione-a-vapori-di-sodio/index.jsp