

Controllo digitale firmware per un microcontrollore per un caricabatteria ultrafast per veicoli elettrici

Candidato: Simone CAIAZZA Relatore: Prof. Radu BOJOI Correlatore: Matteo GREGORIO

Abstract—L'obiettivo del presente progetto di tesi è la scrittura del firmware di configurazione delle periferiche di un μC , il cui scopo è quello di controllare un raddrizzatore attivo di un prototipo di ultrafast charger per veicoli a propulsione totalmente elettrica (BEV) o ibrida (HEV). Il lavoro, svolto in collaborazione del centro di ricerca interdipartimentale PEIC, è stato strutturato nel seguente modo: analisi dell'applicazione in esame con conseguente scelta del μC più adatto; analisi della programmazione e debug di un μC dual core; gestione dei segnali PWM; gestione delle comunicazioni SPI ed I2C con componenti esterni per la gestione delle acquisizioni e di debug, gestione delle acquisizioni di corrente e di tensione e relativa sincronizzazione tra le varie periferiche; validazione sperimentale del firmware su scheda di controllo finale.

I. INTRODUZIONE

L'interesse nel mondo della mobilità elettrica sta aumentando con il passare degli anni, per questo motivo diventa sempre più di interesse tecnologico, la necessità di dover ricaricare il pacco batterie di un veicolo elettrico, o ibrido, nel minor tempo possibile. Esso può essere ricaricato o per mezzo di un caricabatteria presente all'interno del veicolo (on-board) o per mezzo di un caricabatteria esterno al veicolo (off-board). Nel caso in esame si ha un off-board UFC da 50 kW unidirezionale.

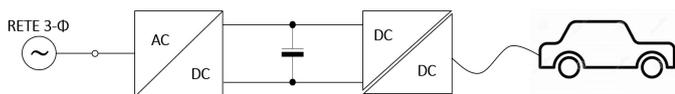


Figura 1: Sistema in esame: Rete + AFE + DC-DC + Veicolo

Il caricabatteria è costituito da due convertitori: una parte che si interfaccia con la rete (AFE T-Type) e da un convertitore DC-DC modulare ed isolato, come mostrato in Figura 1. In questo progetto mi sono occupato esclusivamente del raddrizzatore attivo, In particolare, **i miei contributi personali** sono stati:

- Scelta del μC e stesura pinout
- Studio del debug nel caso di un μC dual core
- Implementazione PWM 3 livelli e generazione dei segnali di gamba
- Gestione delle acquisizioni
- Comunicazioni con ADC SPI e DAC I2C esterni
- Sincronizzazione tra le diverse periferiche
- Stesura di report tecnici

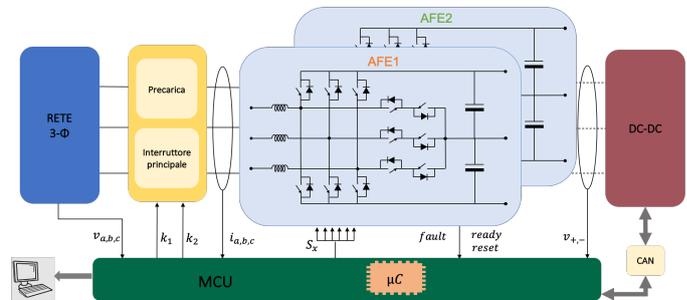


Figura 2: Configurazione AFE Interleaved

II. ACTIVE FRONT-END T-TYPE

Data l'alta potenza del caricabatteria supersfast, è stata scelta la configurazione "interleaved", ossia si utilizzano due unità in parallelo (Figura 2) introducendo una differenza di fase fra le due portanti del periodo di switching.

La Figura 2, inoltre, mette in evidenza i task principali della MCU, ossia deve occuparsi della generazione dei segnali PWM; acquisire le correnti di fase, le tensioni di rete e del DC-link; comandare il relè di precarica e l'interruttore principale, gestire i comandi di ready, reset e fault degli switch di potenza e comunicare con utente (PC) e DC/DC con protocollo CAN. La Figura 3 mostra le caratteristiche minime che deve avere un μC per applicazioni UFC.

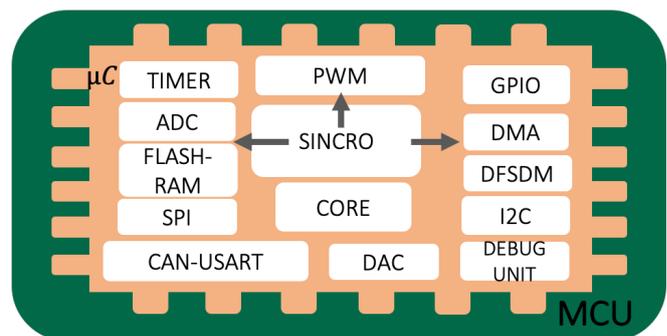


Figura 3: Principali caratteristiche di un μC per UFC

III. PROGRAMMAZIONE E DEBUG DUAL CORE

Data l'impossibilità di eseguire gli anelli di controllo alla frequenza di switching prefissata ($f_{sw} = 72\text{kHz}$) con un singolo core, ho utilizzato un μC dual core (μC STM32H745ZI), dotato di un core Cortex-M7 ed un Cortex-M4. Nello specifico, il CM7 esegue gli anelli di tensione e gestisce le comunicazioni;

mentre il CM4 si occupa degli anelli di corrente, dei segnali PWM e delle acquisizioni. Ho utilizzato il software STM32CubeMx per il codice di inizializzazione del sistema e delle periferiche necessarie; mentre, per la scrittura del firmware ed il debug del codice ho utilizzato l'IDE Keil μ Vision.

IV. GESTIONE PWM 3 LIVELLI E SINCRONIZZAZIONE PERIFERICHE

Per poter utilizzare i timer del μ C ho dovuto adattare il duty-cycle generato dagli anelli di controllo del convertitore, poichè la tecnica PWM 3 livelli prevede l'utilizzo di due portanti. La soluzione adottata, riportata in Figura 4, prevede l'aggiunta di un blocco che in ingresso ha il D^* della singola gamba e in uscita fornisce due duty-cycle equivalenti. Questi duty-cycle sono destinati alle periferiche timer ed è importante che i segnali PWM siano identici a quelli della PWM a 3 livelli.

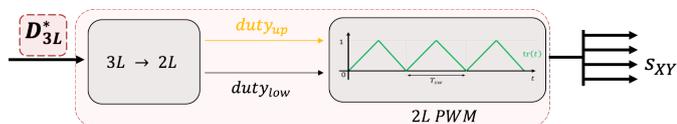


Figura 4: Schema a blocchi sistema in esame

Dopo essermi occupato della generazione dei segnali PWM, ho fatto in modo che tutto fosse sincronizzato. In particolare, è possibile settare un intervallo di tempo di startup, il delay tra l'AFE1 e l'AFE2 (Interleaving) ed il ritardo tra l'inizio dell'ISR e lo start delle acquisizioni

V. CONVERTITORI ANALOGICO-DIGITALE

Ho provveduto alla configurazione del DAC interno al μ C ed alla comunicazione con un DAC esterno I2C, per visualizzare in fase di debug delle forme d'onda analogiche. Inoltre, ho utilizzato la periferica DMA per alleggerire il lavoro della CPU spostando il dato dalla memoria alla periferica I2C.

VI. GESTIONE ACQUISIZIONI

Le acquisizioni di corrente sono effettuate per mezzo di un ADC SPI esterno. La configurazione (Figura 5) è stata scelta per compensare il ritardo introdotto dall'isolatore digitale ed assicurare il giusto sincronismo tra SCLK e DATA, aggiungendo un ritardo nel segnale di clock.

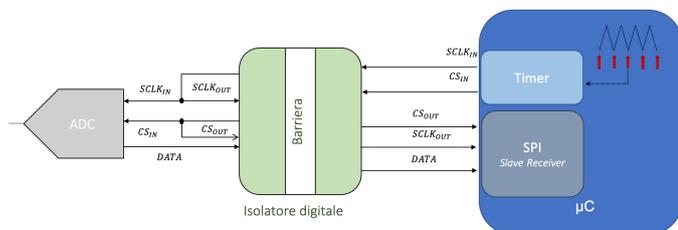


Figura 5: Implementazione acquisizioni di corrente

Le acquisizioni di tensione, invece, sono effettuate da un ADC Sigma-Delta esterno, il quale manda in ingresso al μ C

il dato ed il clock di ritorno. Il dato viene, successivamente, processato dalla periferica (DFSDM), la quale diminuisce il data rate e ne aumenta la risoluzione. Si riportano i test effettuati per dimostrare la corretta configurazione delle periferiche SPI e DFSDM.

Le Figure 6 e 7 sono ottenute confrontando i segnali imposti da generatore di funzioni in ingresso (CH1 e CH2) con le forme d'onda generate dal DAC interno (DAC1 e DAC2).

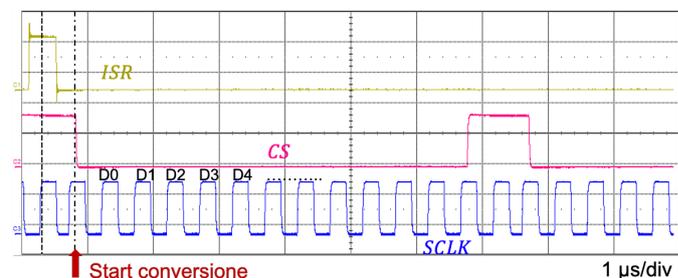


Figura 6: Segnali timer SPI. Dall'alto verso il basso: ISR (72 kHz), CS (144 kHz), SCLK (1 MHz).

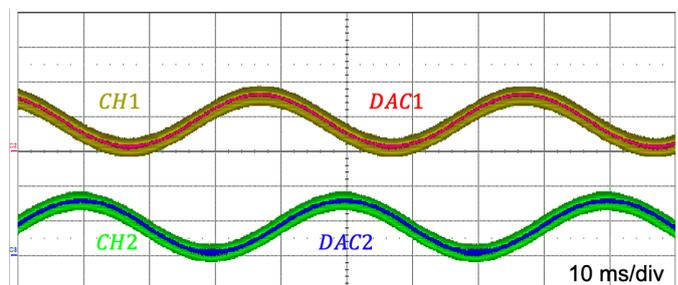


Figura 7: Validazione acquisizioni di corrente e di tensione.



Figura 8: Scheda MCU

VII. CONCLUSIONI

Durante il lavoro di tesi ho seguito tutte le fasi di progettazione del firmware di controllo. Sono partito dalla scelta e all'analisi della toolchain di un μ C dual core, passando per la configurazione delle periferiche necessarie fino ad arrivare alla fase di testing sperimentale sulla scheda MCU (Figura 8), in modo da validarne il funzionamento.