



POLITECNICO DI TORINO
Dipartimento di Energia
Tesi di Laurea Magistrale in Ingegneria
Elettrica

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

Candidato: Simone Caiazza

Relatore: Radu Bojoi

Correlatore: Matteo Gregorio

Ottobre 2020

Ringraziamenti

Ringrazio il Prof. Bojoi per l'opportunità fornitami con il presente progetto di tesi e Matteo Gregorio che con estrema pazienza ha seguito il mio lavoro e che senza il quale non avrei raggiunto i risultati ottenuti. Grazie alla mia famiglia per il sostegno riservatomi durante questi anni, sia a livello didattico che nella vita. Un ringraziamento particolare va a Stefano, Gianluca e Adriana che mi supportano e sopportano costantemente da un pò di anni ormai; senza di loro non sarei qui a festeggiare questo traguardo così importante e quasi inimmaginabile per me. Grazie a tutti i miei amici, ad Iron e alle persone che mi sono accanto nella vita di tutti i giorni.

Indice

| | | |
|----------|--|-----------|
| 1 | Obiettivi della tesi | 1 |
| 2 | Introduzione | 3 |
| 2.1 | Caricabatterie on-board ed off-board | 3 |
| 2.2 | Standard di connessione alla rete | 5 |
| 2.3 | Specifiche applicazione | 8 |
| 3 | Active Front-End T-Type 3 livelli | 10 |
| 3.1 | Configurazione T-type | 10 |
| 3.2 | Controllo di corrente e di tensione | 12 |
| 3.2.1 | Anelli di corrente | 13 |
| 3.2.2 | Anello di tensione | 15 |
| 4 | MCU | 17 |
| 4.1 | Caratteristiche μC per ultra-fast charger | 17 |
| 4.2 | MCU STM32H745xI/G | 19 |
| 5 | Toolchain utilizzata | 24 |
| 5.1 | STM32CubeMX dual core programming | 25 |
| 5.2 | Keil μVision dual core programming | 28 |
| 5.2.1 | Options setting | 29 |
| 5.2.2 | Dual core debug Keil μVision | 35 |
| 6 | Convertitori Digitale-Analogico | 37 |
| 6.1 | Periferica DAC μC | 37 |
| 6.2 | DAC esterno | 39 |
| 7 | Gestione PWM | 46 |
| 7.1 | PWM Active Front-End 3LTT | 46 |
| 7.1.1 | Simulazione PLECS PWM 3LTT | 49 |
| 7.2 | Implementazione digitale PWM 3LTT | 50 |
| 7.2.1 | Simulazione PLECS implementazione digitale | 54 |

| | | |
|-----------|--|------------|
| 7.3 | Configurazione periferica TIMER | 59 |
| 7.3.1 | TIM4 | 61 |
| 7.3.2 | TIM1-TIM8 | 63 |
| 7.3.3 | TIM15 | 66 |
| 7.4 | Configurazione periferica HRTIM | 69 |
| 7.5 | Validazione segnali PWM | 73 |
| 8 | Gestione acquisizioni | 77 |
| 8.1 | Acquisizioni di corrente | 77 |
| 8.1.1 | Generazione CLOCK e CS SPI | 81 |
| 8.2 | Validazione acquisizioni di corrente | 86 |
| 8.2.1 | Validazione generazione CS e SCLK | 87 |
| 8.2.2 | Test SPI in laboratorio | 91 |
| 8.3 | Acquisizioni di tensione | 96 |
| 8.3.1 | Configurazione periferica DFSDM | 98 |
| 8.4 | Validazione acquisizioni di tensione | 101 |
| 9 | Assemblaggio e test AFE | 106 |
| 9.1 | Test segnali PWM su schede finali | 109 |
| 10 | Conclusioni | 112 |
| A | Valori di configurazione timer | 113 |

Elenco delle figure

| | | |
|-----|--|----|
| 2.1 | Overview funzionamento di ricarica con caricabatteria on-board | 4 |
| 2.2 | Overview funzionamento di ricarica con caricabatteria off-board | 4 |
| 2.3 | Tipi di connettori modo 3. a) Tipo 1, b) Tipo 2, c) Tipo 3a, d) Tipo 3b | 5 |
| 2.4 | Tipi di connettori modo 4, standard CHAdeMO | 6 |
| 2.5 | Tipi di connettori modo 4, standard CCS COMBO2 | 6 |
| 2.6 | Schema a blocchi sistema in esame | 8 |
| 2.7 | Costituzione caricabatteria modulare | 9 |
| 2.8 | SOA caricabatteria ultrafast | 9 |
| 3.1 | AFE T-type tre livelli | 11 |
| 3.2 | Configurazione con fase omologhe accoppiati con induttori di modo trasversale, con $x=a,b,c$ | 12 |
| 3.3 | Circuito equivalente modo longitudinale e modo trasversale . . | 13 |
| 3.4 | Controllo correnti di modo longitudinale | 13 |
| 3.5 | Controllo correnti di modo trasversale | 14 |
| 3.6 | Circuito equivalente DC-link | 15 |
| 3.7 | Anello controllo di tensione del DC-link | 16 |
| 4.1 | Schema a blocchi del sistema completo in esame | 18 |
| 4.2 | Blocchi principali di cui deve essere composto un microcon- trollore per un'applicazione come quella in esame | 19 |
| 4.3 | Datasheet STM32H745xI/G [10] | 20 |
| 4.4 | Schema a blocchi periferiche interne STM32H745xI/G [10] . . | 23 |
| 5.1 | Schermata principale STM32CubeMX (in alto). Pin reserva- tion (in basso) | 26 |
| 5.2 | Passaggio da STM32CubeMX a Keil μ Vision | 27 |
| 5.3 | Organizzazione delle cartelle output STM32CubeMX | 27 |
| 5.4 | Schermata principale Keil μ Vision | 28 |
| 5.5 | Tab per settare le opzioni del progetto in Keil | 29 |

ELENCO DELLE FIGURE

| | | |
|------|--|----|
| 5.6 | Opzioni da settare Cortex-M7, tab target | 31 |
| 5.7 | Opzioni da settare Cortex-M7, tab linker | 31 |
| 5.8 | Opzioni da settare Cortex-M7, tab debug | 32 |
| 5.9 | Opzioni da settare Cortex-M7, tab flash download | 32 |
| 5.10 | Opzioni da settare Cortex-M4, tab target | 33 |
| 5.11 | Opzioni da settare Cortex-M4, tab linker | 33 |
| 5.12 | Opzioni da settare Cortex-M4, tab debug | 34 |
| 5.13 | Opzioni da settare Cortex-M4, tab flash download | 34 |
| 5.14 | Debug dual core in Keil μ Vision [13] | 36 |
| 6.1 | Schema a blocchi periferica DAC [14] | 38 |
| 6.2 | Package DAC MCP4725 [15] | 40 |
| 6.3 | Package DAC MCP4728 [16] | 40 |
| 6.4 | Write commands per DAC input register e EEPROM[15] . . . | 41 |
| 6.5 | Setup sperimentale validazione DAC esterno | 42 |
| 6.6 | Andamenti di SDA e SCL della comunicazione I2C | 43 |
| 6.7 | Andamenti di: segnale analogico in input (linea blu) e uscita DAC (linea rossa), onda quadra a 50 Hz, DAC MCP4725 . . . | 43 |
| 6.8 | Andamenti di: segnale analogico in input (linea blu) e uscita DAC (linea rossa), onda quadra a 300 Hz, DAC MCP4725 . . | 44 |
| 6.9 | Andamenti di: segnale analogico in input (linea blu) e uscita DAC (linea rossa), onda sinusoidale a 50 Hz, DAC MCP4725 . | 44 |
| 6.10 | Andamenti di: segnale analogico in input (linea blu) e uscita DAC (linea rossa), onda triangolare a 50 Hz, DAC MCP4725 . | 45 |
| 7.1 | Convertitore 3 livelli T-Type monofase | 46 |
| 7.2 | Stato P, stato M e stato N | 47 |
| 7.3 | Portanti triangolari | 47 |
| 7.4 | $-\frac{V_{DC}}{2} < V_{XM} < 0$ | 48 |
| 7.5 | $0 < V_{XM} < +\frac{V_{DC}}{2}$ | 48 |
| 7.6 | $-\frac{V_{DC}}{2} < V_{XM} < 0 \rightarrow 0 < V_{XM} < +\frac{V_{DC}}{2}$ | 49 |
| 7.7 | Schematico PLECS utilizzato per la simulazione | 50 |
| 7.8 | Simulazione PLECS $-\frac{V_{DC}}{2} < V_{XM} < 0$ | 51 |
| 7.9 | Simulazione PLECS $0 < V_{XM} < +\frac{V_{DC}}{2}$ | 51 |
| 7.10 | Simulazione PLECS $-\frac{V_{DC}}{2} < V_{XM} < 0 \rightarrow 0 < V_{XM} < +\frac{V_{DC}}{2}$. | 52 |
| 7.11 | Overview counter timer | 52 |
| 7.12 | Schema a blocchi passaggio da PWM 3 livelli a PWM 2 livelli | 53 |
| 7.13 | Diagramma di flusso passaggio da PWM 3 livelli a PWM 2 livelli | 54 |
| 7.14 | $-\frac{V_{DC}}{2} < V_{XM} < 0$ | 55 |
| 7.15 | $0 < V_{XM} < +\frac{V_{DC}}{2}$ | 55 |

| | |
|--|----|
| 7.16 Schematico PLECS utilizzato per la simulazione duty-cycle equivalenti | 56 |
| 7.17 Simulazione PLECS $d(t) < 0.5$ | 57 |
| 7.18 Simulazione PLECS $d(t) > 0.5$ | 58 |
| 7.19 Simulazione PLECS con duty-cycle che passa da 0.2 a 0.8 | 58 |
| 7.20 Overview gestione PWM del convertitore 1 | 59 |
| 7.21 Schema a blocchi implementazione PWM | 60 |
| 7.22 Schema a blocchi general-purpose timers [14] | 62 |
| 7.23 Counter di TIM4 e generazione startup time | 63 |
| 7.24 Schema a blocchi advanced-control timers [14] | 64 |
| 7.25 Counter di TIM1 | 64 |
| 7.26 Esempio di generazione segnali PWM TIM1 | 65 |
| 7.27 Schema a blocchi TIM15 [14] | 67 |
| 7.28 Esempio di generazione segnali PWM TIM15 | 67 |
| 7.29 Meccanismo di funzionamento Combined PWM mode [17] | 68 |
| 7.30 Implementazione digitale PWM TIM15 | 68 |
| 7.31 Schema a blocchi High-resolution timer [14] | 70 |
| 7.32 Schema a blocchi unità di timing A..E [14] | 71 |
| 7.33 Logica di generazione segnali PWM HRTIM | 72 |
| 7.34 Esempio di generazione segnali PWM HRTIM | 72 |
| 7.35 Validazione ritardo tra AFE 1 e AFE 2 | 74 |
| 7.36 Validazione PWM con duty costante e ritardo nullo | 74 |
| 7.37 Validazione PWM con duty costante e ritardo pari a $5 \mu s$ | 74 |
| 7.38 Analisi teorica PWM con duty-cycle variabile e reset i_{pos} | 75 |
| 7.39 Validazione PWM con duty-cycle variabile e reset i_{pos} | 75 |
| 8.1 Overview gestione acquisizioni di corrente | 78 |
| 8.2 Schema a blocchi acquisizioni di corrente | 78 |
| 8.3 Schema a blocchi ADC SAR[18] | 79 |
| 8.4 Schema a blocchi AD7276 | 80 |
| 8.5 Diagramma temporale convertitore AD7276 | 80 |
| 8.6 Logica di generazione segnali PWM HRTIM | 81 |
| 8.7 Sincronizzazione segnali di CS e SCLK | 82 |
| 8.8 Sincronizzazione segnali di CS e SCLK | 83 |
| 8.9 TIM2 | 83 |
| 8.10 TIM3 | 84 |
| 8.11 TIM5 | 85 |
| 8.12 Overall generazione segnali di CS e SCLK | 85 |
| 8.13 Validazione generazione CS e SCLK. Dall'alto verso il basso: Andamenti del vertice alto di TIM8, vertice basso di TIM8, CS e SCLK, test 1 | 88 |

ELENCO DELLE FIGURE

| | | |
|------|---|-----|
| 8.14 | Verifica ritardo tra inizio ISR/CS e tra fronte di discesa del CS e fronte di discesa di SCLK, test 1 | 88 |
| 8.15 | Validazione generazione CS e SCLK. Dall'alto verso il basso: Andamenti del vertice alto di TIM8, vertice basso di TIM8, CS e SCLK, test 2 | 89 |
| 8.16 | Verifica ritardo tra inizio ISR/CS e tra fronte di discesa del CS e fronte di discesa di SCLK, test 2 | 89 |
| 8.17 | Validazione generazione CS e SCLK. Dall'alto verso il basso: Andamenti del vertice alto di TIM8, vertice basso di TIM8, CS e SCLK, test 3 | 90 |
| 8.18 | Verifica ritardo tra inizio ISR/CS e tra fronte di discesa del CS e fronte di discesa di SCLK, test 3 | 90 |
| 8.19 | Setup sperimentale validazione acquisizioni di corrente | 91 |
| 8.20 | Verifica ISR a 72 kHz e acquisizione di corrente a 144 kHz | 92 |
| 8.21 | Andamenti di: vertice basso di TIM8 (linea gialla), CS (linea rossa) e SCLK (linea blu), con ISR a 72 kHz | 92 |
| 8.22 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 50 Hz | 93 |
| 8.23 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 150 Hz | 94 |
| 8.24 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 2 kHz | 94 |
| 8.25 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), sinusoide a 50 Hz | 95 |
| 8.26 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda triangolare a 50 Hz | 96 |
| 8.27 | Overview gestione acquisizioni di tensione | 97 |
| 8.28 | Schema a blocchi ADS1205 | 97 |
| 8.29 | Schema a blocchi periferica DFSDM[19] | 98 |
| 8.30 | Schema a blocchi periferica DFSDM | 100 |
| 8.31 | Setup sperimentale validazione acquisizioni di corrente | 101 |
| 8.32 | Aggiornamento segnale DAC (linea blu) alla frequenza della ISR (linea gialla) 72 kHz | 102 |
| 8.33 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 50 Hz, ADC Sigma-Delta | 103 |
| 8.34 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 150 Hz, ADC Sigma-Delta | 103 |
| 8.35 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 2 kHz, ADC Sigma-Delta | 104 |

| | | |
|------|---|-----|
| 8.36 | Andamenti di: segnali analogici in input (linea gialla e linea verde) e uscita DAC (linea rossa e linea blu), onda triangolare a 50 Hz, ADC Sigma-Delta | 104 |
| 8.37 | Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda triangolare a 50 Hz, ADC Sigma-Delta | 105 |
| 9.1 | Scheda MCU con μC STM32h745XIH6 | 107 |
| 9.2 | Scheda Carrier e scheda MCU (in rosso) | 108 |
| 9.3 | Schede driver | 108 |
| 9.4 | Setup sperimentale test segnali PWM | 109 |
| 9.5 | Segnali PWM $d = 0.5$ | 110 |
| 9.6 | Segnali PWM $d > 0.5$ | 110 |
| 9.7 | Segnali PWM $d < 0.5$ con delay=10 μs | 111 |
| A.1 | Riassunto timer utilizzati | 114 |
| A.2 | Esempio valori ARR | 114 |

Elenco delle tabelle

| | | |
|-----|--|----|
| 2.1 | Principali specifiche dell'applicazione in esame | 8 |
| 8.1 | Input test $f_{SCLK} = 1MHz$ | 87 |
| 8.2 | Input test $f_{SCLK} = 250kHz$ | 89 |
| 8.3 | Input test $f_{SCLK} = 100kHz$ | 90 |

Capitolo 1

Obiettivi della tesi

Il presente lavoro di tesi si colloca all'interno di un progetto di ricerca del Politecnico di Torino, il cui scopo è quello di realizzare un prototipo di caricabatteria ultrafast per veicoli elettrici. Il suddetto prototipo è costituito da due convertitori, un raddrizzatore bidirezionale ed un DC/DC. Nello specifico, il lavoro di tesi si è concentrato sul raddrizzatore e si è provveduto alla scrittura del firmware di programmazione delle periferiche del microcontrollore. Nella prima parte della tesi, ho contribuito all'analisi del funzionamento di un microcontrollore dual core ed ho implementato la tecnica PWM per un convertitore a tre livelli. Successivamente, ho contribuito alla gestione delle sincronizzazioni tra PWM, ADC, DAC, EXTI ed alla gestione delle comunicazioni tramite protocolli SPI ed I2C. Nell'ultima parte del lavoro ho contribuito alla fase di testing delle configurazioni delle periferiche sulla scheda NUCLEO utilizzata ed effettuato il debug del firmware sopracitato. Ho inoltre partecipato alla stesura del pinout e di documentazione tecnica sotto forma di report, utile per comprendere il lavoro personale svolto.

La seguente tesi è stata suddivisa in 10 capitoli:

1. **Obiettivi della tesi:** esposizione dello scopo e della struttura dell'elaborato
2. **Introduzione:** funzionamento di un caricabatteria ultrafast per veicoli elettrici, standard di connessione alla rete e specifiche applicazione in esame
3. **Active Front-End T-Type 3 livelli:** studio del funzionamento di un AFE T-Type a 3 livelli, anelli di controllo di corrente e di tensione
4. **MCU:** specifiche richieste ad un microcontrollore per controllo digitale di un caricabatteria off-board e principali caratteristiche della famiglia di microcontrollori dual core STM-H745xI/G

5. **Toolchain utilizzata:** focus sui software utilizzati per la generazione del codice di inizializzazione delle periferiche e per la scrittura e debug del firmware di controllo, concentrandosi sul caso specifico delle periferiche del μC STM32H745ZI
6. **DAC:** implementazione pratica della periferica DAC interna al μC e comunicazione con DAC esterno tramite protocollo I2C
7. **Gestione PWM:** analisi teorica, implementazione e validazione della generazione dei segnali PWM
8. **Gestione acquisizioni:** analisi teorica, implementazione e validazione delle acquisizioni di corrente e di tensione
9. **assemblaggio e test AFE** assemblaggio e validazione sperimentale preliminare schede AFE, in particolare test PWM
10. **Conclusioni:** riassunto del lavoro svolto e cenni su possibili risvolti futuri

Capitolo 2

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 (HEV), nel minor tempo possibile. Il pacco batterie degli HEVs (Hybrid Electric Vehicles) e BEVs (Battery Electric Vehicles) può essere ricaricato o per mezzo di un caricabatteria presente all'interno del veicolo o per mezzo di un caricabatteria esterno al veicolo. Nel primo caso si parla di caricabatteria on-board, nel secondo caso di caricabatteria off-board.

2.1 Caricabatterie on-board ed off-board

Un on-board charger, or OBC, è la struttura, interna al veicolo, che converte in DC il flusso di energia elettrico AC proveniente dalla rete. Il vantaggio è dovuto al fatto che la ricarica può essere effettuata anche nella propria abitazione a discapito di un aumento dei tempi di ricarica poichè la potenza prelevabile dalla rete è limitata a valori per usi commerciali. Il tempo di ricarica può essere calcolato, in prima approssimazione, come segue:

$$T_{ricarica} = \frac{N_{celle} \cdot C_{cella,nominale} \cdot V_{cella,nominale}}{P_{ricarica}} \quad (2.1)$$

dove sono stati riportati i principali parametri che influenzano la velocità di ricarica: essi sono: la capacità nominale della singola cella in [Wh] $C_{cella,nominale}$, il numero di celle N_{celle} , la tensione nominale della cella $V_{cella,nominale}$ e la potenza di ricarica $P_{ricarica}$. Un tipico sistema di ricarica on-board è mostrato nella Fig. 2.1:

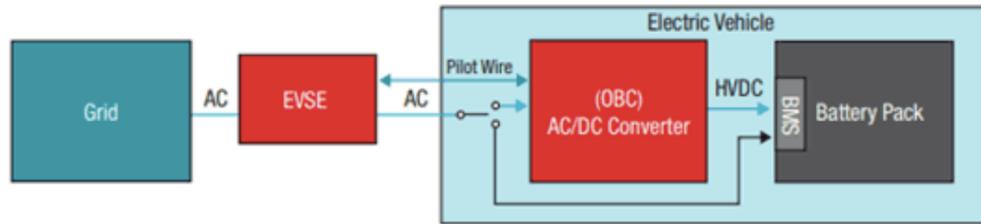


Figura 2.1: Overview funzionamento di ricarica con caricabatteria on-board

Il flusso di energia elettrica AC dalla rete arriva direttamente all'OBC attraverso l'EVSE, dove EVSE sta per Electric Vehicle Supply Equipment e non si riferisce nient'altro che alle stazioni di ricarica[1]. L'OBC poi lo converte in DC e carica il pacco batterie per mezzo del BMS, acronimo di Battery Management System. Il Pilot wire è usato per identificare il tipo di connettore connesso al veicolo e setta la corrente richiesta in input per l'on-board charger.

Nel caso di un caricabatteria off-board si ha a che fare con una stazione di ricarica in DC che preleva il flusso di energia AC dalla rete e lo converte sotto forma di grandezze continue. Essi sono anche chiamati caricatori di "Modo 4" e comunemente conosciuti come DC Fast Chargers (DCFC). Il vantaggio di questo tipo di ricarica è certamente la velocità a discapito di una maggiore complessità del sistema in quanto c'è bisogno di comunicare costantemente con il veicolo elettrico per caricare il proprio pacco batterie in maniera efficiente e sicura.

Un tipico sistema di ricarica off-board è mostrato nella Fig. 2.2, dove si può notare il by-pass dell'on-board charger.

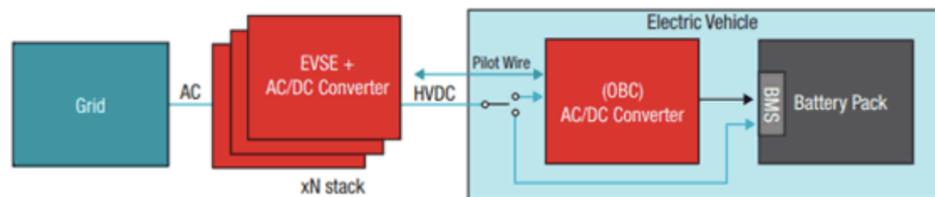


Figura 2.2: Overview funzionamento di ricarica con caricabatteria off-board

Si può quindi intuire che una delle soluzioni, per ridurre i tempi di ricarica delle batterie per veicoli elettrici e/o ibridi, è individuabile nel fatto di

aumentare la potenza assorbita dalla rete elettrica e destinata alle batterie, facendo uso di un caricabatteria superfast.

2.2 Standard di connessione alla rete

Secondo la normativa IEC 61851-1 relativa alle colonnine di ricarica, sono ammessi 4 Modi per la ricarica dei veicoli [2], e precisamente:

- Modo 1: Ricarica lenta (6-8 h) a 16 A, (corrente alternata) ammessa solamente in ambiente domestico privato. È possibile utilizzare una semplice presa domestica o una presa industriale fino a 32 A, il limite di potenza assorbita è di 4 kW
- Modo 2: Ricarica lenta (6-8 h) a 16 A, (corrente alternata) ammessa in ambiente domestico e pubblico. Sul cavo di alimentazione del veicolo è presente un dispositivo denominato Control Box che garantisce la sicurezza delle operazioni durante la ricarica. come nel caso del modo 1, le prese utilizzabili sono quelle domestiche o industriali fino a 32 A, le potenze sono dello stesso ordine del modo 1
- Modo 3: Ricarica lenta (6-8 h) a 16 A o mediamente rapida (30 min – 1 h) a 63 A, 400V (Modalità con sistema di sicurezza PWM), ammessa in ambiente domestico e pubblico. Esistono tre tipi di connettori che sono stati standardizzati ed è necessario installare una colonnina di ricarica dedicata. Le potenze tipiche sono fino a 22kW

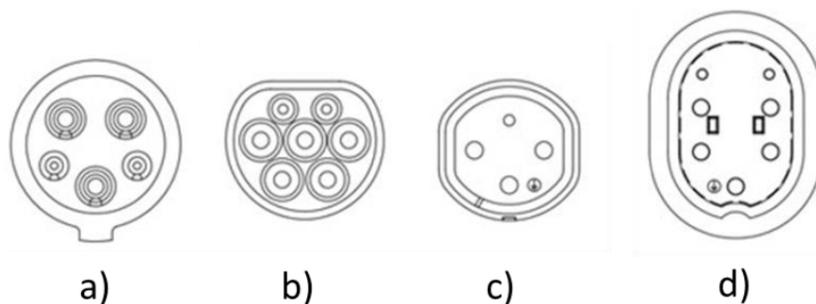


Figura 2.3: Tipi di connettori modo 3. a) Tipo 1, b) Tipo 2, c) Tipo 3a, d) Tipo 3b

- Modo 4: Ricarica ultra rapida in corrente continua ammessa solamente in ambiente pubblico. Con questo sistema è possibile ricaricare i veicoli in alcuni minuti, il caricabatterie è esterno al veicolo. L'associazione che promuove il Modo 4 è nata in Giappone ed è denominata CHAdeMO. Attualmente gli standard principali per questa modalità sono il CHAdeMO e CCS Combo2 [3] .

CHAdeMO



Figura 2.4: Tipi di connettori modo 4, standard CHAdeMO

CCS Combo2



Figura 2.5: Tipi di connettori modo 4, standard CCS COMBO2

Inoltre, le normative citate dividono i sistemi di ricarica in due tipologie, in base alla relativa potenza massima [4]:

1. La prima categoria (potenza Standard) comprende tutti i sistemi di ricarica in corrente alternata (AC - Modo 3 secondo CEI EN 61851) e i sistemi di ricarica in corrente continua di potenza fino a 22 kW (DC-Modo 4 secondo CEI EN 61851)
2. La seconda categoria (potenza Elevata) comprende tutti i sistemi di ricarica in corrente continua di potenza superiore a 22 kW (DC-Modo 4 secondo CEI EN 61851)

2.3 Specifiche applicazione

Si riportano, nella tabella 2.1 le specifiche principali dell'applicazione trattata all'interno del presente progetto di tesi.

| | |
|-------------------------------------|----------------|
| Potenza nominale | 50 kW |
| Tensione rete concatenata UE | 400 V |
| Massima tensione batteria | 1000 V |
| Minima tensione batteria | 250 V |
| Collocamento | Off-board |
| Struttura caricabatteria | Unidirezionale |
| Raffreddamento | Aria |
| Power factor corrector | Sì |
| Interfacce di comunicazione | CAN, USART |
| Isolato | Sì |

Tabella 2.1: Principali specifiche dell'applicazione in esame

Come mostrato in Fig. 2.6, per soddisfare le specifiche sopracitate, è necessario che il caricabatteria ultrafast sia costituito da due convertitori: una parte che si interfaccia con la rete (AFE), che ha lo scopo di convertire il flusso di energia elettrica alternata in forma continua, con fattore di potenza unitario; la restante parte è rappresentata da un convertitore DC-DC isolato che ha invece lo scopo di regolare il flusso di potenza elettrica richiesto dall'utenza. L'AFE ha un ulteriore compito di regolare la tensione del DC-link in maniera tale che il DC-DC lavori per più tempo possibile nel punto di massima efficienza, nel caso in esame quando la tensione del DC-link e la tensione di batteria sono allo stesso valore.

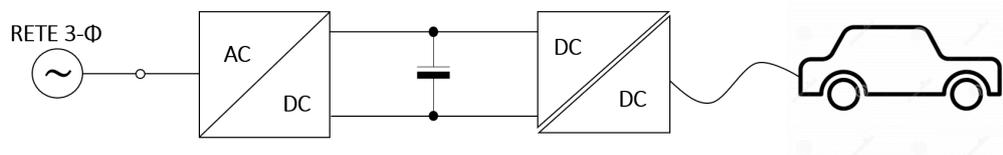


Figura 2.6: Schema a blocchi sistema in esame

A causa dell'elevata potenza richiesta, il convertitore DC/DC presenta una struttura modulare, sfruttando il DC-link diviso in due condensatori

di un AFE a 3 livelli. Un totale di quattro moduli LLC da 12.5 kW sono progettati e realizzati indipendentemente, suddividendo la potenza nominale e consentendo di spegnerne una o più moduli a carico leggero [5]. Inoltre, l'uscita di una coppia di moduli può essere riconfigurata per essere in parallelo o in serie con l'altra coppia, a seconda della richiesta in tensione della batteria, come mostrato in Fig. 2.7.

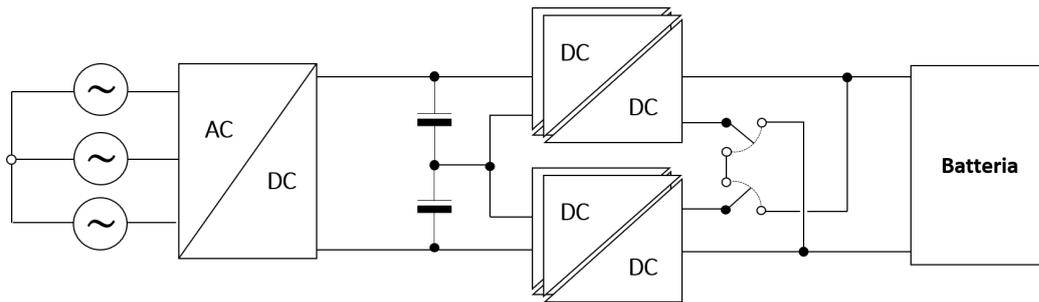


Figura 2.7: Costituzione caricabatteria modulare

Si riporta in Fig. 2.8 la safe operating area, la quale è delimitata dai valori nominali di corrente, tensione e potenza. Si distinguono inoltre due regioni; la regione in parallelo corrisponde alla posizione del deviatore che collega in parallelo le uscite dei moduli DC/DC, mentre la regione blu corrisponde alla configurazione serie. Nella configurazione parallelo le due coppie di moduli DC/DC sono connesse in parallelo e si è in grado di erogare il doppio della corrente del caso di configurazione serie.

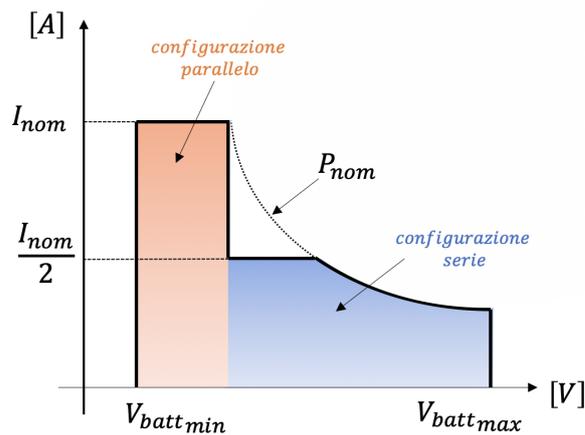


Figura 2.8: SOA caricabatteria ultrafast

Capitolo 3

Active Front-End T-Type 3 livelli

3.1 Configurazione T-type

In questo progetto ci si è occupati esclusivamente del convertitore a contatto con la rete trifase. Entrando sempre di più nello specifico, il convertitore in esame è costituito da un Active Front-End T-Type tre livelli riportato in Fig. 3.1, dove:

- per 'Active' si intende che la struttura di conversione è costituita da switch attivi, ossia che richiedono l'applicazione di una grandezza elettrica sul terminale di comando per poter svolgere il proprio compito di interruttore
- 'Front-End' in quanto è la parte che si interfaccia con la rete
- 'T-Type 3 livelli' è una delle possibili configurazioni di un convertitore a 3 livelli

Una struttura di conversione viene denominata a 'tre livelli' poichè è in grado di realizzare, istantaneamente, tre diverse tensioni sull'utenza: V_+ , 0 e V_- , prendendo come riferimento il punto medio M.

I convertitori a tre livelli offrono alcuni vantaggi rispetto ad un equivalente convertitore a due livelli come: la maggiore efficienza ad alta frequenza di commutazione, la necessità di minori induttanze in input ed una minore necessità di filtraggio per evitare disturbi elettromagnetici condotti [6]. Tra gli svantaggi di una struttura di conversione a tre livelli, invece, si sottolineano la maggiore complessità del circuito, dovuto ad un numero maggiore di

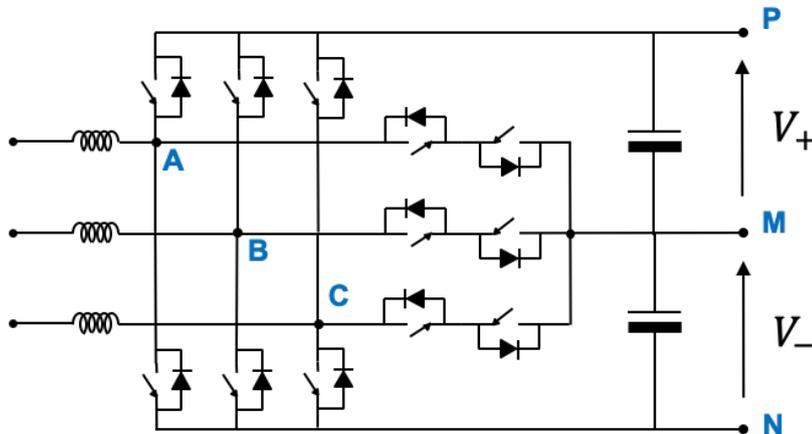


Figura 3.1: AFE T-type tre livelli

switch di potenza necessari a realizzare le prestazioni volute, ed alla necessità di controllare la tensione del punto medio M. La soluzione adottata per il presente lavoro di tesi, mostrata in Fig. 3.1, è caratterizzata da un basso fattore di distorsione armonica della corrente di rete e con cui è possibile ottenere un elevato fattore di potenza. Si può osservare che sia i diodi sia i transistor, collegati tra i punti a valle degli induttori di Boost (A, B, C) ed il punto medio M sono dimensionati per sostenere la metà della tensione del DC-link, a differenza dei restanti switch e diodi che devono sostenere per intero la tensione del DC-link, $(V_+ + V_-)$.

Data l'alta potenza del caricabatteria supersfast, è stata scelta la configurazione "interleaved", ossia si utilizzano due unità in parallelo (singola unità riportata in Fig. 3.1) introducendo una differenza di fase fra le due portanti del periodo di switching. È una pratica diffusa in media ed alta potenza poiché riduce lo stress termico sui semiconduttori e le dimensioni dei diodi e dei reattori di potenza [7]. Inoltre, con la tecnica dell'interleaving, il ripple di corrente totale in ingresso dovuto alla commutazione PWM può essere ridotto e, come risultato, si ha la riduzione del filtro EMI in entrata al convertitore poiché si ha minore distorsione di corrente rispetto al caso non interleaved. Oltre ai sopracitati vantaggi, essendo una soluzione con due convertitori in parallelo, bisogna considerare però un numero più alto di sensori di corrente e un aumento nella complessità del controllo PWM.

La struttura per cui si è predisposto il codice di controllo oggetto del presente elaborato presenta delle modifiche rispetto ad un AFE T-type tre livelli come quello presentato fino ad ora. In particolare il convertitore in esame non utilizza sei induttori di Boost (uno per fase, quindi essendo due

strutture in interleaved sei induttori in totale), ma si utilizzano tre induttori di Boost; infatti, ogni fase omologa delle due strutture in interleaved ha l'induttore in comune e si utilizzano tre coppie di induttori i cui avvolgimenti sono avvolti in verso opposto (Fig. 3.2). In questo modo, si ottiene il vantaggio di specializzare sia gli induttori di Boost che gli induttori accoppiati in quanto i primi vengono dimensionati per sostenere principalmente la corrente di carico, mentre gli induttori accoppiati per essere sollecitati da un flusso caratterizzato da tanto ripple e valor medio nullo. Infatti, se le due correnti differenziali sono uguali, i flussi generati dagli induttori accoppiati si compensano e di conseguenza "vedono" esclusivamente il ripple dovuto allo sfasamento delle due portanti. Inoltre, un ulteriore vantaggio è rappresentato dal fatto che gli induttori accoppiati si oppongono intrinsecamente al passaggio delle correnti di circolazione, cosa che deve esser fatta a livello software se non si utilizza la soluzione con induttori accoppiati.

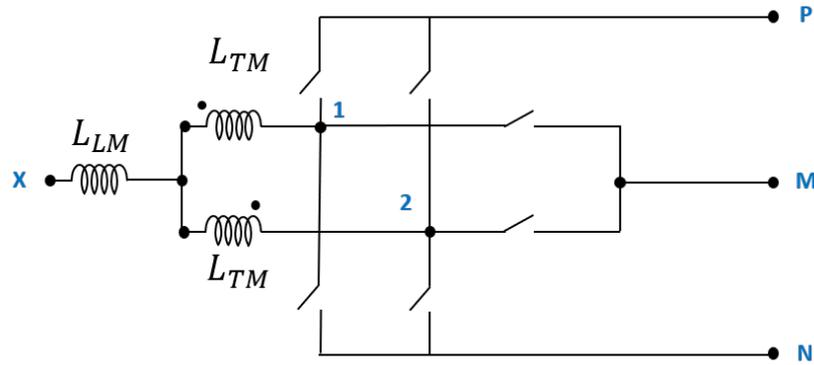


Figura 3.2: Configurazione con fase omologhe accoppiati con induttori di modo trasversale, con $x=a,b,c$

3.2 Controllo di corrente e di tensione

La soluzione utilizzata può essere rappresentata per semplicità con il circuito equivalente di Fig. 3.3 in cui v_{LM} e v_{TM} rappresentano rispettivamente la tensione di modo longitudinale e la tensione di modo trasversale, dove il modo longitudinale regola il trasferimento di potenza mentre quello trasversale tiene sotto controllo la corrente di circolazione tra le due gambe. Esse sono definite nel seguente modo:

$$v_{LM} = \frac{v_{1x} + v_{2x}}{2} \tag{3.1}$$

$$v_{TM} = \frac{v_{1x} - v_{2x}}{2} \quad (3.2)$$

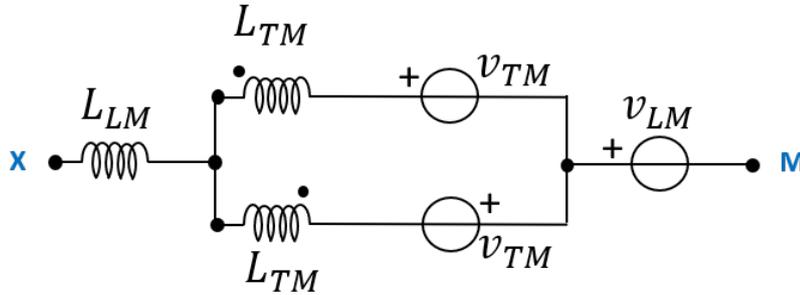


Figura 3.3: Circuito equivalente modo longitudinale e modo trasversale

3.2.1 Anelli di corrente

Il controllo di corrente di modo longitudinale è lo stesso di un controllo di corrente di un Active Front-end trifase senza induttori accoppiati, poichè solo due correnti sono indipendenti tra loro. L'unica differenza risiede nel fatto che, per fare in modo che il tempo di calcolo sia il più piccolo possibile, si evita la trasformata di rotazione la quale richiede di implementare nel μC operazioni trigonometriche quali seno e coseno, effettuando il controllo in assi $\alpha\beta$. Si accetta in questo caso il compromesso di lavorare con delle grandezze sinusoidali e quindi con errore non nullo a regime (con un regolatore PI), il quale è comunque trascurabile in quanto la frequenza di controllo utilizzata è nettamente superiore rispetto alla frequenza fondamentale delle grandezze da controllare.

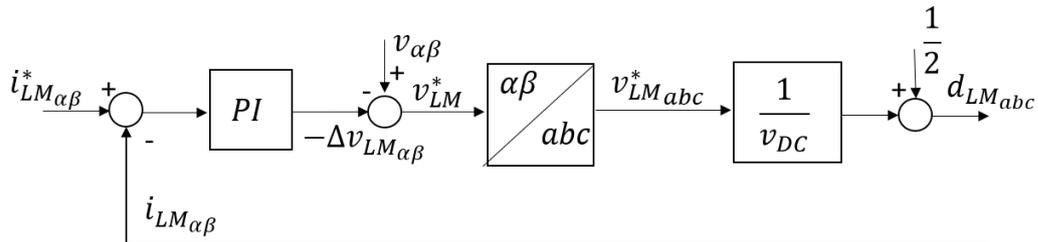


Figura 3.4: Controllo correnti di modo longitudinale

Per quanto riguarda il controllo di corrente di modo trasversale il discorso è differente. Infatti, essendo tali correnti tutte indipendenti tra loro, bisogna

implementare tre differenti anelli di controllo, imponendo come riferimento che le correnti di circolazione di modo trasversale siano nulle.

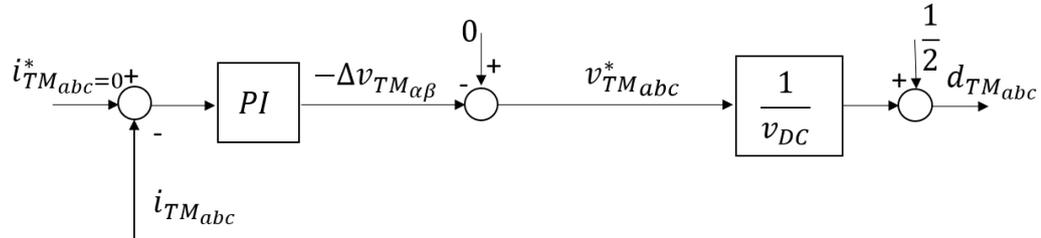


Figura 3.5: Controllo correnti di modo trasversale

Le figure Fig.3.4 e Fig.3.5 mostrano come ottenere i duty-cycle $d_{LM,abc}$ e $d_{TM,abc}$ da applicare al convertitore, ai quali bisogna ancora aggiungere il duty d_0 dovuto al riferimento di tensione di zero-sequence v_0^* . Esso viene aggiunto per compensare e tiene conto di molteplici fattori;

- Essa dipende dalla tecnica di modulazione adottata, in questo caso si è scelta la ZMPC (Zero Mid-Point Current) [8]
- Dipende dal fatto che si vuole $V_+ = V_-$. Per fare ciò si aggiunge un anello di controllo della tensione di mid-point, eseguito ogni 150 Hz [9]

3.2.2 Anello di tensione

Per quanto riguarda la tensione del DC-link si vuole che sia il più costante possibile e fare in modo che il DC-DC lavori per più tempo possibile nel punto di massima efficienza, per questo viene introdotto un anello di tensione. Dal punto di vista del controllo della tensione v_{DC} , gli anelli di controllo di corrente sono stati considerati per semplicità a guadagno unitario. L'anello di controllo di tensione è riportato in Fig. 3.7.

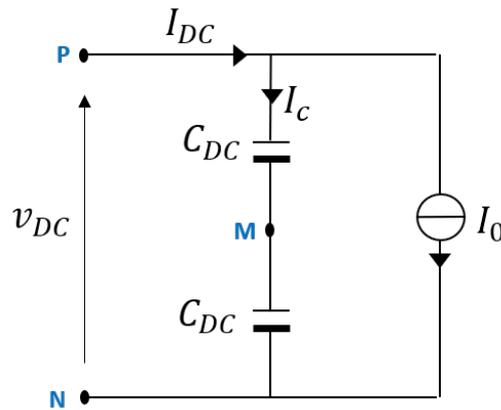


Figura 3.6: Circuito equivalente DC-link

Facendo riferimento alla figura sopra riportata (Fig. 3.6), che rappresenta il circuito equivalente visto dal DC-link, si possono scrivere le seguenti equazioni:

$$I_C = I_{DC} - I_0 = \frac{C_{DC}}{2} \cdot v_{DC} \dot{\quad} \quad (3.3)$$

da cui in Laplace, esplicitando la tensione istantanea presente sulla somma dei due condensatori, si può scrivere

$$v_{DC} = \frac{2I_C}{sC_{DC}} = \frac{2(I_{DC} - I_0)}{sC_{DC}} \quad (3.4)$$

dove:

1. I_{DC} , corrente in uscita dal convertitore
2. I_0 , corrente assorbita dal carico (DC/DC)

Risulta a questo punto utile mettere in correlazione la I_{DC} con la i_a ; per fare ciò si effettua un bilancio energetico supponendo, idealmente, che la struttura di conversione abbia rendimento e fattore di potenza unitari.

$$P = \frac{3}{2} v_{pk} i_{pk} \cos \phi = \frac{3}{2} v_{pk} i_d = v_{DC} i_{DC} \quad (3.5)$$

da cui,

$$I_{DC} = \frac{3}{2} \frac{v_{pk}}{v_{DC}} i_d \quad (3.6)$$

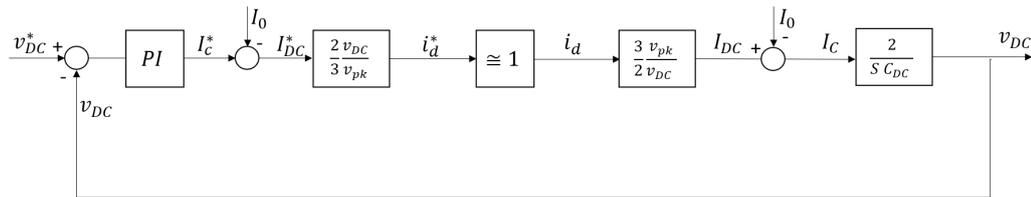


Figura 3.7: Anello controllo di tensione del DC-link

Capitolo 4

MCU

Il microcontrollore è il cervello dell'intero sistema in quanto è l'elemento che si occupa di controllare che tutto funzioni correttamente. In particolare esso si occupa di comandare gli switch di potenza, i relè di prearica, gestire i fault, gestire le acquisizioni e comunicare con l'esterno. Il successo dei microcontrollori negli ultimi anni è dovuto al fatto che essi integrano in uno stesso chip la CPU e molte altre periferiche che lo rendono versatile per vari tipi di applicazioni, pur non rinunciando ad una sufficiente potenza di calcolo.

4.1 Caratteristiche μC per ultra-fast charger

Dopo essersi occupati in una prima fase dell'analisi teorica di un convertitore T-Type a tre livelli e degli aspetti riguardanti il controllo di corrente e di tensione della struttura di conversione in esame, si analizzano le specifiche principali richieste ad un microcontrollore utilizzabile in applicazioni di questo tipo; successivamente, si passa alla configurazione e sulla validazione delle periferiche necessarie a svolgere tutti i tasks richiesti a garantire il corretto funzionamento dell'AFE, sia nei confronti della rete, sia nei confronti del convertitore DC/DC. A tal proposito si può osservare in Fig. 4.1 uno schema a blocchi dove viene presentato un overview del sistema in questione. Si distinguono le due unità di AFE T-type accoppiate, il convertitore DC/DC, la rete trifase ed un blocco, tra l'AFE e la rete stessa, contenente i relè di prearica ed il relè responsabile del soft start.

Si possono osservare inoltre le grandezze minime che bisogna acquisire: correnti di fase (i_a, i_b, i_c), le tensioni di rete (v_{ab}, v_{bc}, v_{ca}) e le tensioni del DC-link (V_+ e V_-). I segnali indicati come q_x rappresentano i segnali PWM destinati ai gate drivers dei MOSFET che costituiscono l'AFE. Entrando di più nel dettaglio di quello che è il microcontrollore, in Fig. 4.2, si sottolinea-

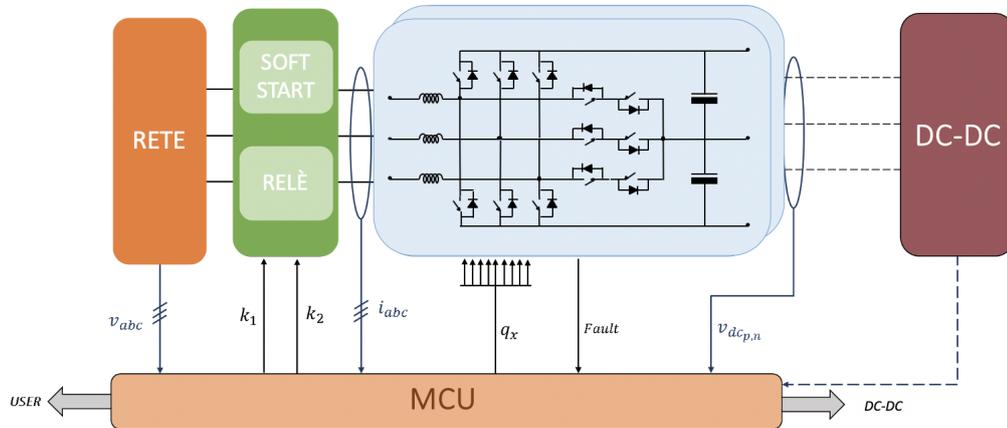


Figura 4.1: Schema a blocchi del sistema completo in esame

no le periferiche principali richieste come: timer (generazione segnali PWM e sincronizzazione), ADC (per acquisire grandezze analogiche come temperature etc), GPIO (per comandare relè o accendere led di stato), EXTI (gestione dei fault), DMA (per trasferire dati senza passare per la CPU) e periferiche di comunicazione come: SPI, I2C, USART, FDCAN etc indispensabili per interfacciarsi con il convertitore DC/DC e l'utente, il quale nel caso più semplice può essere un computer. Vengono riportati anche componenti fondamentali come: core e memorie (RAM, FLASH, CACHE) le cui specifiche possono variare in base all'applicazione.

E' inoltre opportuno sottolineare il fatto che il microcontrollore utilizzato all'interno del progetto di tesi non è stato pensato per soddisfare i requisiti minimi di safety per applicazioni automotive. Infatti, l'obiettivo principale del progetto è quello di realizzare un prototipo di un caricabatteria in modo da validarne il funzionamento, l'algoritmo di controllo e la procedura di design.

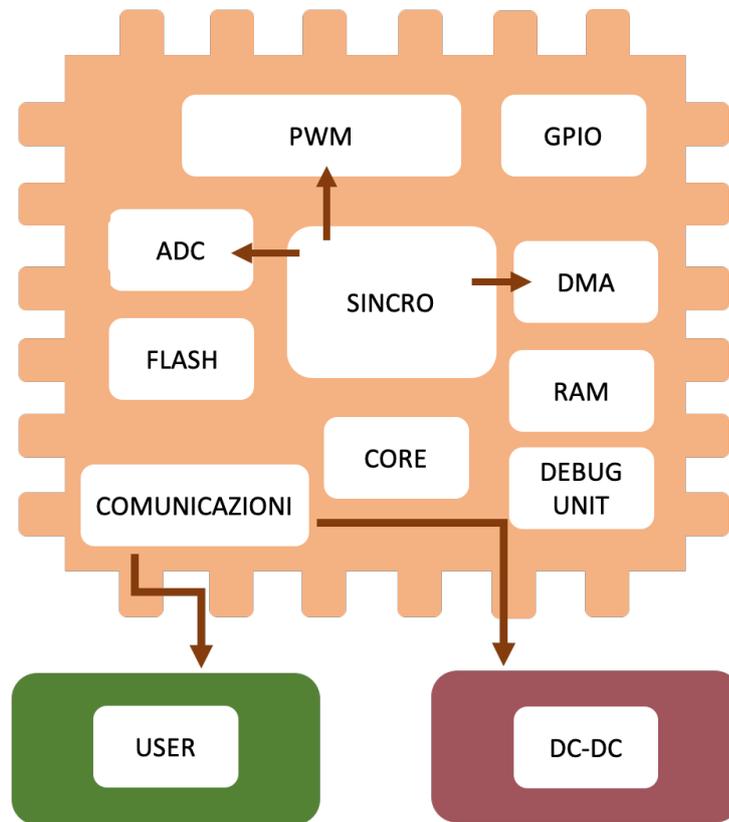


Figura 4.2: Blocchi principali di cui deve essere composto un microcontrollore per un'applicazione come quella in esame

4.2 MCU STM32H745xI/G

In questo capitolo viene presentata la famiglia di microcontrollori STM32H7-45xI/G. A questo livello della trattazione, è sufficiente introdurre le caratteristiche principali della famiglia di μC STM32H745xI/G prodotti da STMicroelectronics, in quanto le differenze principali consistono in diversa quantità di memoria e diverso package.

La Fig. 4.3 mostra le caratteristiche principali di tale μC , il quale si distingue per la presenza di due core: un core Cortex-M7 con f_{CLOCK} fino a 480 MHz ed un core Cortex-M4 con f_{CLOCK} fino a 240 MHz, entrambi con architettura ARM a 32-bite RISC con FPU (Floating Point Unit).



STM32H745xI/G

Dual 32-bit Arm® Cortex®-M7 up to 480MHz and -M4 MCUs, up to 2MB Flash, 1MB RAM, 46 com. and analog interfaces, SMPS

Datasheet - production data

Features

Dual core

- 32-bit Arm® Cortex®-M7 core with double-precision FPU and L1 cache: 16 Kbytes of data and 16 Kbytes of instruction cache; frequency up to 480 MHz, MPU, 1027 DMIPS/2.14 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- 32-bit Arm® Cortex®-M4 core with FPU, Adaptive real-time accelerator (ART Accelerator™) for internal Flash memory and external memories, frequency up to 240 MHz, MPU, 300 DMIPS/1.25 DMIPS /MHz (Dhrystone 2.1), and DSP instructions

Memories

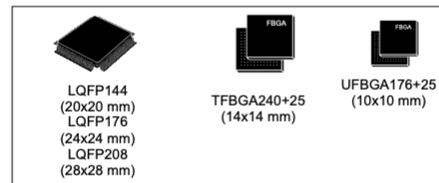
- Up to 2 Mbytes of Flash memory with read-while-write support
- 1 Mbyte of RAM: 192 Kbytes of TCM RAM (inc. 64 Kbytes of ITCM RAM + 128 Kbytes of DTCM RAM for time critical routines), 864 Kbytes of user SRAM, and 4 Kbytes of SRAM in Backup domain
- Dual mode Quad-SPI memory interface running up to 133 MHz
- Flexible external memory controller with up to 32-bit data bus: SRAM, PSRAM, SDRAM/LPSDR SDRAM, NOR/NAND Flash memory clocked up to 125 MHz in Synchronous mode
- CRC calculation unit

Security

- ROP, PC-ROP, active tamper

General-purpose input/outputs

- Up to 168 I/O ports with interrupt capability



Reset and power management

- 3 separate power domains which can be independently clock-gated or switched off:
 - D1: high-performance capabilities
 - D2: communication peripherals and timers
 - D3: reset/clock control/power management
- 1.62 to 3.6 V application supply and I/Os
- POR, PDR, PVD and BOR
- Dedicated USB power embedding a 3.3 V internal regulator to supply the internal PHYs
- Embedded regulator (LDO) to supply the digital circuitry
- High power-efficiency SMPS step-down converter regulator to directly supply V_{CORE} and/or external circuitry
- Voltage scaling in Run and Stop mode (6 configurable ranges)
- Backup regulator (~ 0.9 V)
- Voltage reference for analog peripheral/ V_{REF+}
- 1.2 to 3.6 V V_{BAT} supply
- Low-power modes: Sleep, Stop, Standby and V_{BAT} supporting battery charging

Low-power consumption

- V_{BAT} battery operating mode with charging capability
- CPU and domain power state monitoring pins
- 2.95 μ A in Standby mode (Backup SRAM OFF, RTC/LSE ON)

Figura 4.3: Datasheet STM32H745xI/G [10]

In Fig. 4.4 viene mostrato lo schema a blocchi dei principali elementi presenti all'interno del μ C STM32H745xI/G. Si distinguono nella parte alta in verde i due core, sulla destra e sulla sinistra tutte le periferiche e, inoltre, si osservano i bus APB o AHB che vengono utilizzati per trasportare i dati da un core all'altro, tra core e periferiche e tra periferiche e l'esterno, rappresentato, in questo caso specifico dal convertitore DC-DC e dall'utente, rendendo disponibile, o prelevando, l'informazione su pin GPIO.

Le principali periferiche configurate in questo progetto sono state:

- **TIMER**: si distinguono in advanced control timers, i general purpose timers e i basic timers, messi in ordine di complessità e funzionalità decrescenti. Utilizzata nel caso in esame per generare i segnali PWM e garantire la sincronizzazione tra le varie periferiche
- **HRTIM**: sono dei timer ad alta risoluzione, caratterizzati da una maggiore versatilità rispetto alla periferica TIMER standard. Utilizzata per generare segnali PWM
- **SPI**: periferica utilizzata per gestire le comunicazioni tramite il protocollo omonimo 'Serial Peripheral Interface' con ADC esterno per acquisizioni di corrente
- **DFSDM**: è l'acronico di 'Digital Filter for Sigma Delta Modulators' utilizzato, nel nostro caso, per acquisire il bit stream in uscita da un ADC Sigma-Delta esterno al microcontrollore. U
- **DMA**: utile per gestire operazioni senza gravare sulla CPU, in particolare per spostare dati da memoria a memoria o da memoria a periferica e viceversa
- **GPIO**: 'General Purpose Input Output' utilizzabili come input o output digitale
- **DAC**: è in grado di produrre sul suo terminale di uscita una grandezza di tipo analogico, in funzione del valore, digitale, contenuto nei propri registri. Utilizzata come debug per valutare forme d'onda analogiche
- **I2C**: utile nel caso si intende comunicare con un device esterno, come nel caso di DAC o un sensore di temperatura, tramite protocollo "Inter Integrated Circuit"

- ADC: è in grado di campionare sui suoi terminali di ingresso una grandezza di tipo analogico, con risoluzione variabile. L'ADC integrato all'interno del presente μC è un AD ad approssimazioni successive

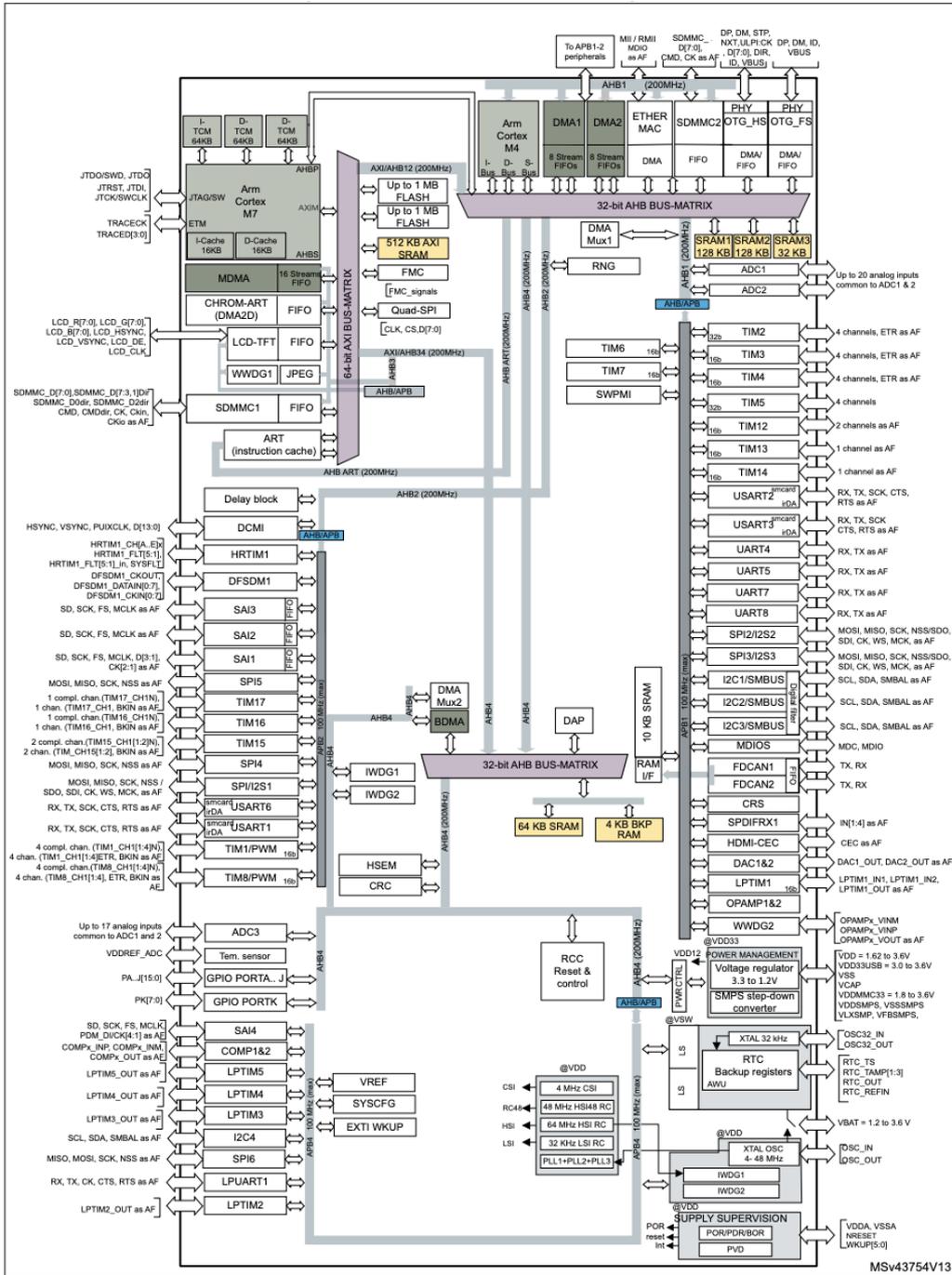


Figura 4.4: Schema a blocchi periferiche interne STM32H745xI/G [10]

Capitolo 5

Toolchain utilizzata

Il presente capitolo è incentrato sulla toolchain utilizzata per l'inizializzazione, configurazione del codice di controllo oggetto del progetto di tesi, costituita dai software STM32CubeMX e Keil μ Vision. STM32CubeMX è un tool grafico che consente una semplice configurazione dei microcontrollori STM32 e fornisce il codice di inizializzazione scritto in linguaggio C, ad esempio i file `.c/.h` delle periferiche che si utilizzano all'interno del progetto, ad esempio nel caso dei timer si avranno due file: `tim.c/tim.h` con il codice di configurazione di tutti i timer utilizzati. Per quanto riguarda Keil μ Vision, è un IDE (integrated development environment) che combina la gestione del progetto, l'ambiente di runtime, le strutture di compilazione, la modifica del codice sorgente e il debug del programma in un unico potente ambiente di sviluppo [11].

In particolare verranno presentati i due software sopracitati con un focus dedicato al caso in esame, ossia di un microcontrollore dual core della famiglia STM32H745xI/G.

5.1 STM32CubeMX dual core programming

Nel caso di programmazione di un microcontrollore che integra due core, STM32CubeMX consente di scegliere a quale core riservare il pin che si vuole configurare; per fare ciò nella pinout view si può scegliere inizializzare il pin sul Cortex-M7, sul Cortex-M4 o tenerlo libero. E' possibile inoltre selezionare la voce "Pin Stacking", utile nel caso si vuole sottolineare che lo stesso pin può essere associato a periferiche diverse, come mostrato in Fig. 5.1. E' importante precisare che tale funzione non ha alcun effetto sul codice generato da Cube, ma è utile nel caso si voglia ricordare a quale altra periferica e con che funzione può essere configurato lo stesso pin. Nel caso invece si voglia configurare una qualsiasi periferica, nel lato destro della schermata principale (Fig. 5.1), è presente un elenco con tutte le periferiche integrato nel microcontrollore suddivise per categorie. Scelta la periferica, bisogna assegnarla ad uno dei due core, oppure ad entrambi. Infatti, se la periferica viene assegnata ad un core, l'altro core non avrà nulla a che fare con tale periferica.

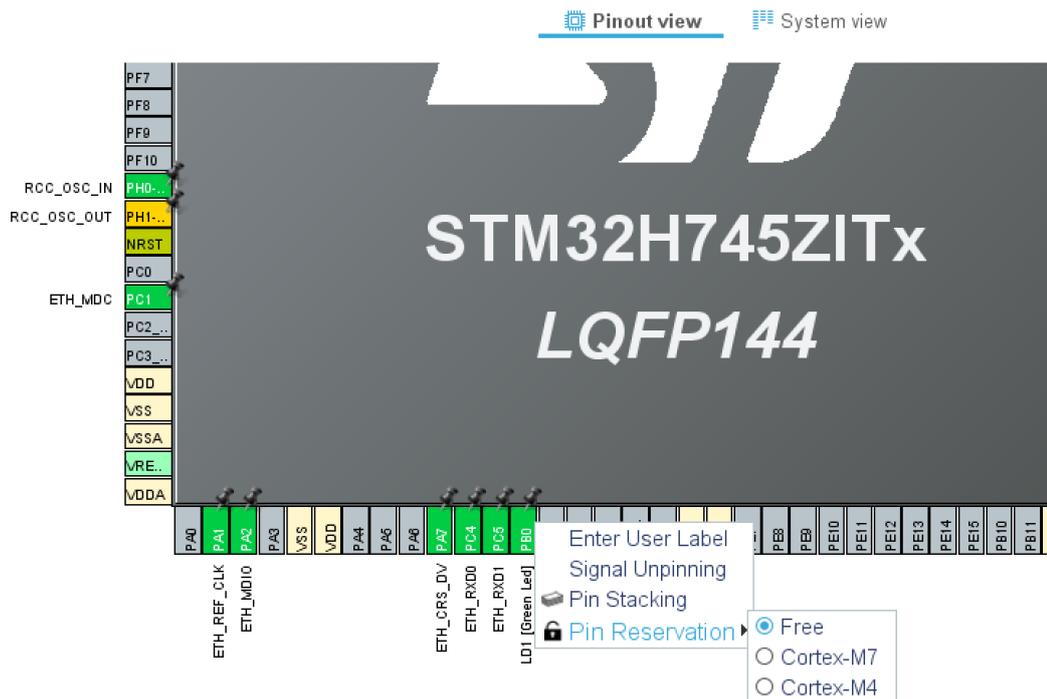
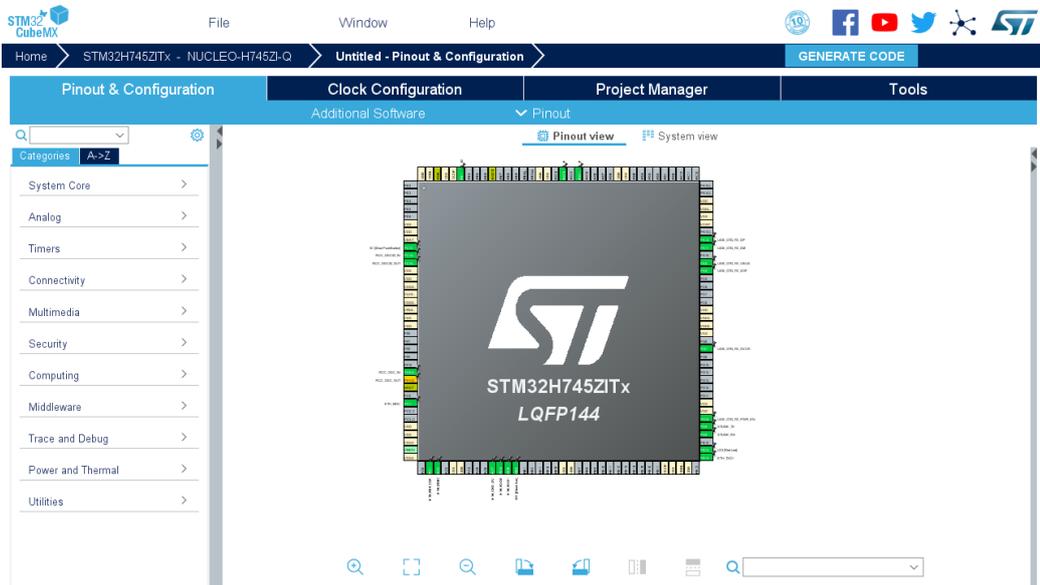


Figura 5.1: Schermata principale STM32CubeMX (in alto). Pin reservation (in basso)

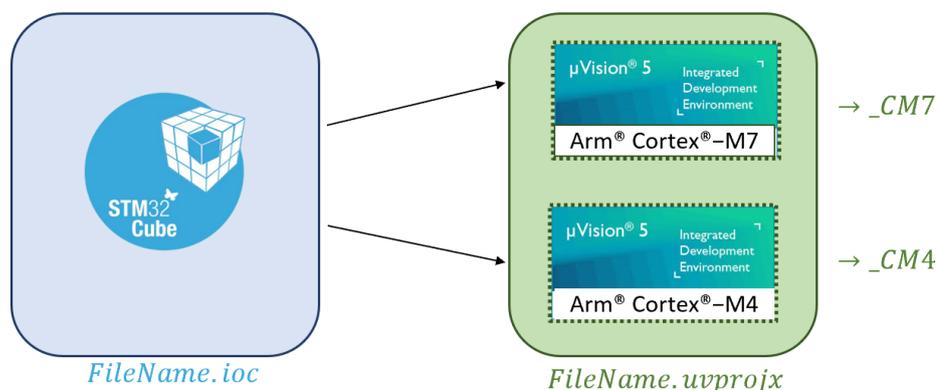


Figura 5.2: Passaggio da STM32CubeMX a Keil μ Vision

| Name | Date modified | Type | Size |
|--------------|------------------|----------------|-------|
| CM4 | 01/04/2020 17:36 | File folder | |
| CM7 | 01/04/2020 17:36 | File folder | |
| Common | 01/04/2020 17:06 | File folder | |
| Drivers | 01/04/2020 17:06 | File folder | |
| MDK-ARM | 02/04/2020 17:07 | File folder | |
| .mxproject | 01/04/2020 17:37 | MXPROJECT File | 34 KB |
| TestH745.ioc | 02/04/2020 11:59 | STM32CubeMX | 27 KB |

Figura 5.3: Organizzazione delle cartelle output STM32CubeMX

Dopo aver configurato tutte le periferiche necessarie, nel tab "Project Manager" si possono settare i principali parametri del codice da generare, tra cui è possibile impostare l'IDE. STM32CubeMX genera due sotto progetti, uno per ogni core, come mostrato in Fig. 5.3, dove:

- CM4: file .c/.h del Cortex-M4
- CM7: file .c/.h del Cortex-M7
- COMMON: file .c che fornisce le funzioni di inizializzazione di sistema per applicazioni dual core
- DRIVERS: HAL (Hardware abstraction layers) drivers e CMSIS (Cortex Microcontroller Software Interface Standard) per entrambi i core
- MDK-ARM: progetto μ Vision
- FileName.ioc: progetto STM32CubeMX

5.2 Keil μ Vision dual core programming

In Fig. 5.4 sono stati messi in evidenza le voci principali per lavorare su un progetto dual core in Keil μ Vision; in particolare si distinguono:

- Project tree, in verde
- Finestra per selezionare su quale core lavorare, in giallo
- μ Vision editor, in rosso
- tasto per entrare in debug, in nero

Si osserva inoltre che nel project tree, i file .c che corrispondono al core non selezionato sono identificati con un segnale di divieto di ingresso.

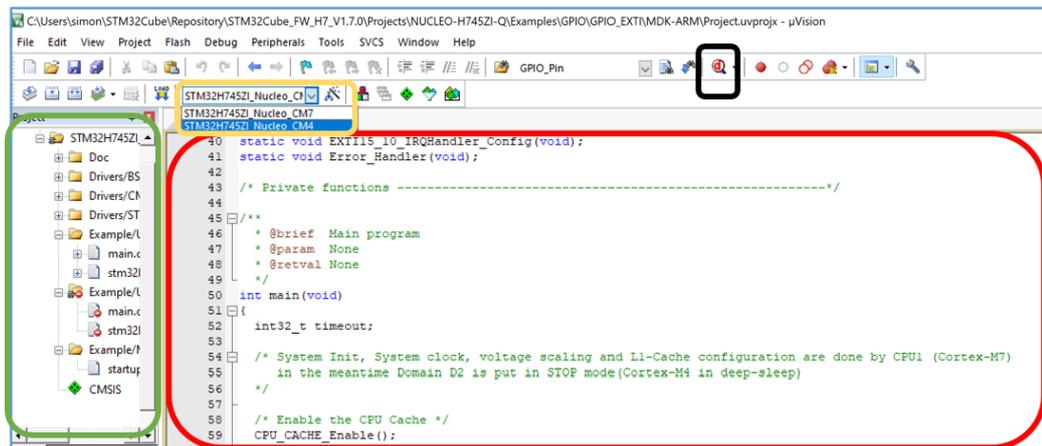


Figura 5.4: Schermata principale Keil μ Vision

5.2.1 Options setting

Ogni volta che un nuovo progetto viene generato da STM32CubeMX, bisogna apportare delle modifiche nella sezione "Project options" (in rosso in Fig. 5.5). Si riportano, per entrambi i core, le opzioni da settare per il corretto funzionamento nel caso specifico del microcontrollore STM32H745ZI [12].

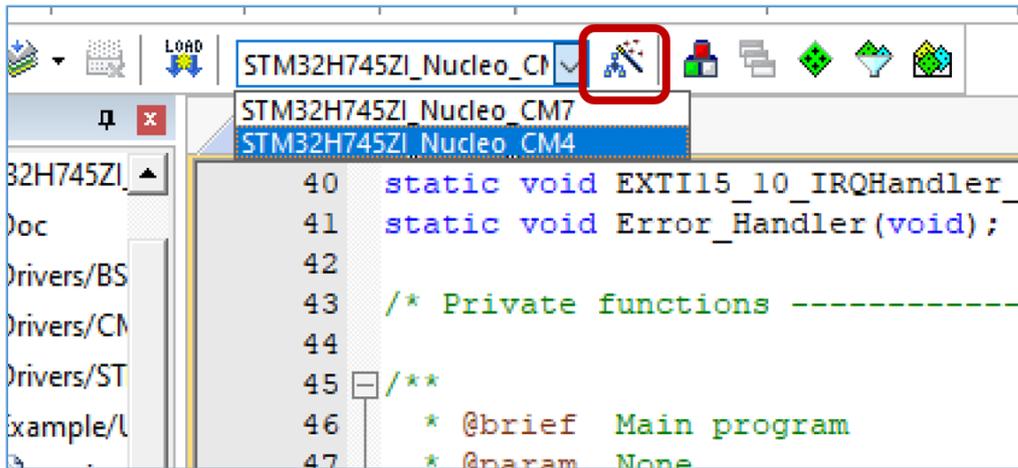


Figura 5.5: Tab per settare le opzioni del progetto in Keil

Options setting CM7

1. Assicurarsi che le giuste area di memoria siano selezionate in Project Options for Target → Target (Fig. 5.6)
2. Assicurarsi che sia selezionato Project Options for Target → Linker → Use Memory Layout from Target dialog (Fig. 5.7)
3. Selezionare ST-Link come debugger da: Project → Options for Target → Debug
4. Per entrare correttamente in debug, nella finestra Debug (Fig. 5.8):
 - Selezionare connect options → Under Reset
 - Selezionare reset options → HW RESET
 - Selezionare Performance profile → Balanced Perf
5. Dalla finestra Flash Download (Fig. 5.9) selezionare:
 - RAM for algorithm = start: 0x20000000 size: 0x00001000
 - Program algorithm = start: 0x08000000 size: 0x00100000

Options setting CM4

1. Assicurarsi che le giuste area di memoria siano selezionate in Project Options for Target → Target (Fig. 5.10)
2. Assicurarsi che sia selezionato Project Options for Target → Linker → Use Memory Layout from Target dialog (Fig. 5.11)
3. Selezionare ST-Link come debugger da: Project → Options for Target → Debug
4. Per entrare correttamente in debug, nella finestra Debug (Fig. 5.12):
 - Selezionare connect options → Under Reset
 - Selezionare reset options → HW RESET
 - Selezionare Performance profile → Balanced Perf
5. Dalla finestra Flash Download (Fig. 5.13) selezionare:
 - RAM for algorithm = start: 0x10000000 size: 0x00001000
 - Program algorithm = start: 0x08000000 size: 0x00200000

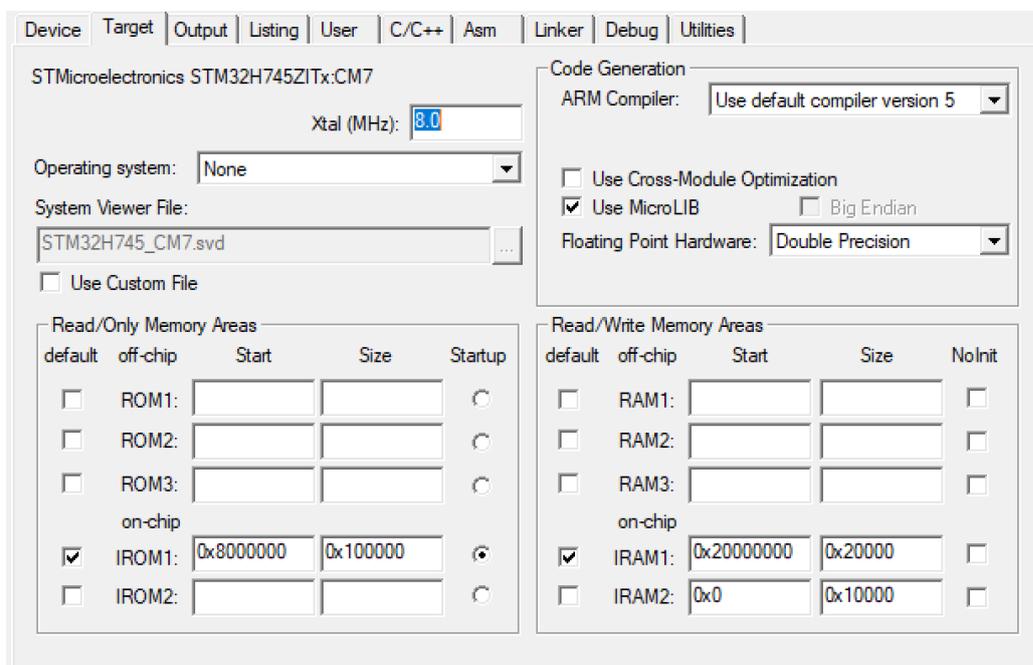


Figura 5.6: Opzioni da settare Cortex-M7, tab target

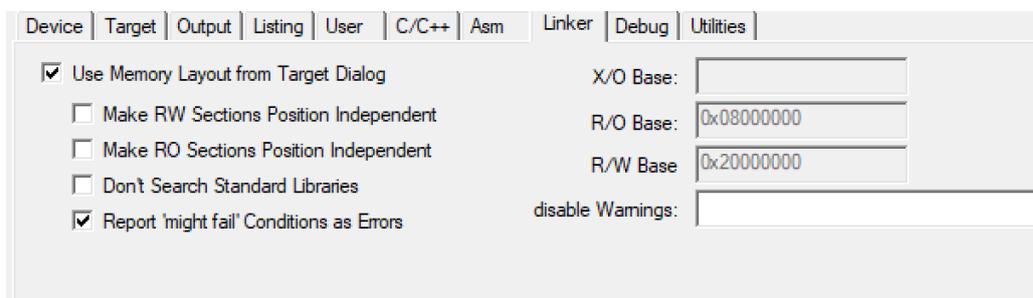


Figura 5.7: Opzioni da settare Cortex-M7, tab linker

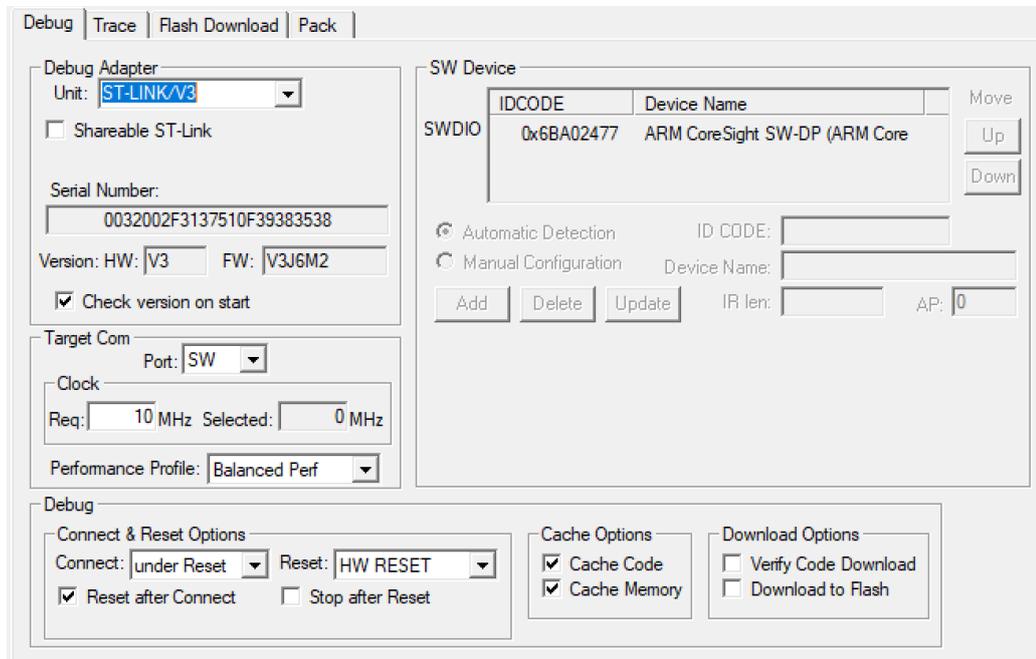


Figura 5.8: Opzioni da settare Cortex-M7, tab debug

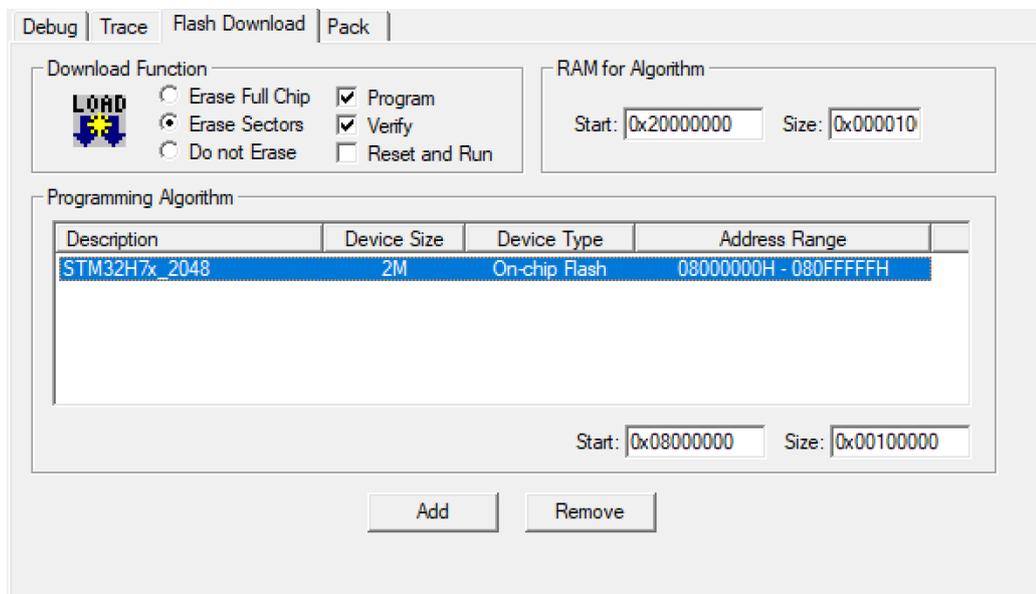


Figura 5.9: Opzioni da settare Cortex-M7, tab flash download

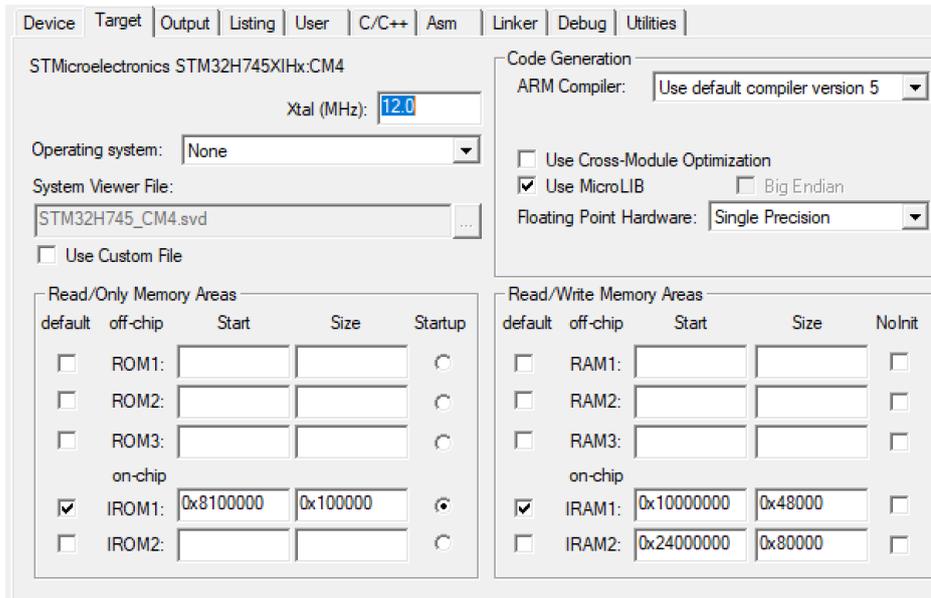


Figura 5.10: Opzioni da settare Cortex-M4, tab target

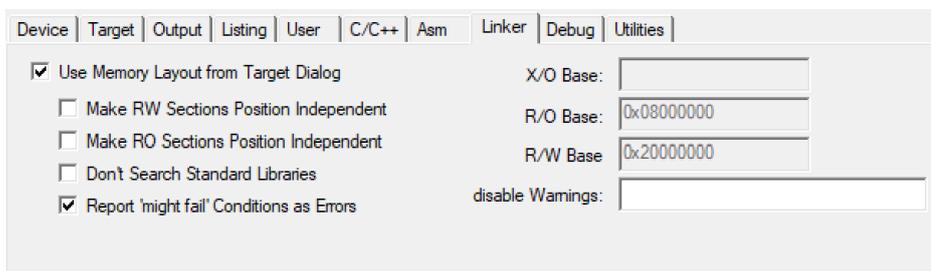


Figura 5.11: Opzioni da settare Cortex-M4, tab linker

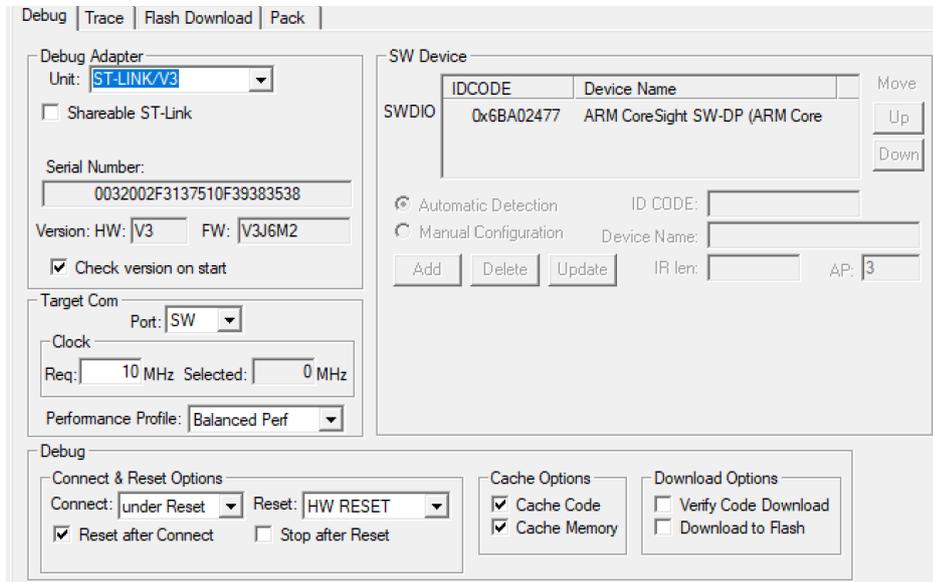


Figura 5.12: Opzioni da settare Cortex-M4, tab debug

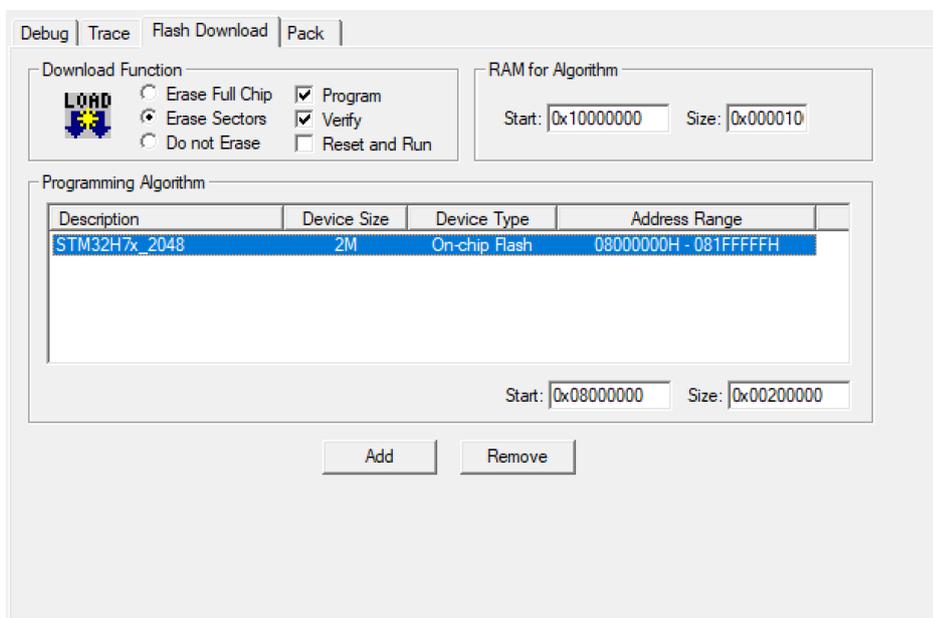


Figura 5.13: Opzioni da settare Cortex-M4, tab flash download

5.2.2 Dual core debug Keil μ Vision

Per avviare una sessione di debug bisogna cliccare sui pulsanti download e debug. Essa può essere eseguita in diverse modalità:

- Entrambi i core vengono eseguiti contemporaneamente
- I core funzionano in modo indipendente

Per accedere al debug con entrambi i core contemporaneamente, bisogna assicurarsi che l'opzione Shareable ST-Link sia selezionato per entrambi. Le azioni da seguire sono [13]:

1. Avviare il debug del Cortex-M7 per scaricare sia le immagini Cortex-M4 che Cortex-M7
2. Impostare il core Cortex-M7 in esecuzione in modo che il semaforo HSEM venga rilasciato e Cortex-M4 si riattivi dalla modalità Stop
3. Avviare il debug del Cortex-M4 utilizzando la freccia accanto all'icona di debug. In questo modo è in modalità di esecuzione e l'utente può interromperlo

La Fig. 5.14 riporta un riassunto del funzionamento di una sessione di debug nel caso del μ C STM32H745ZI. Si possono osservare le porte di accesso di debug differenti per i due core, la porta AP0 per il Cortex-M7 e la porta AP3 per il Cortex-M4.

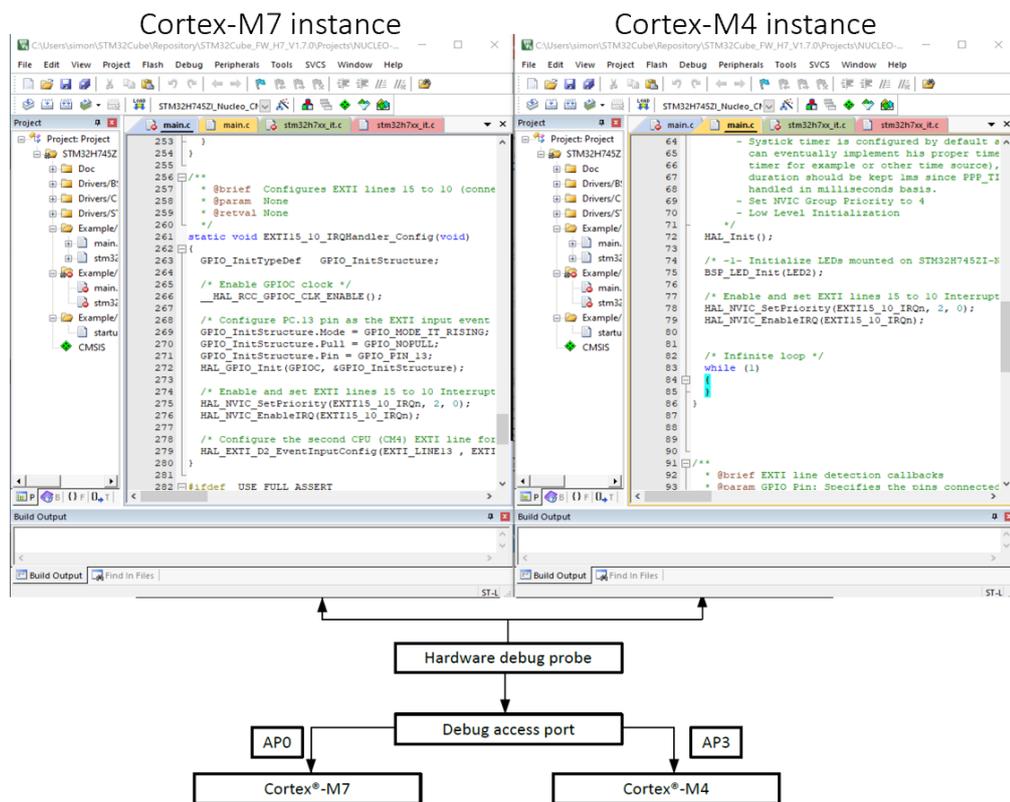


Figura 5.14: Debug dual core in Keil μ Vision [13]

Capitolo 6

Convertitori Digitale-Analogico

Il presente capitolo si concentra sulla necessità di implementare una soluzione per visualizzare delle forme d'onda analogiche in modo da testare il corretto funzionamento del firmware di controllo. Questo è stato effettuato poiché risulta più immediato poter visualizzare direttamente dei segnali analogici; inoltre, può essere utile voler visualizzare gli andamenti di grandezze "non fisiche", come il duty-cycle o di grandezze a valle delle trasformazioni di Clarke e di Park. Nella prima parte del capitolo si fa riferimento alla periferica DAC interna ai microcontrollori della famiglia STM-H745xI/G, mentre nella seconda parte si è utilizzato un DAC esterno. Si è deciso di optare per questa soluzione, in quanto la periferica interna al microcontrollore possiede solo 2 canali e può essere di interesse pratico, in fase di debug, poter vedere almeno 3 forme d'onda, trattandosi di un sistema trifase.

6.1 Periferica DAC μC

La periferica DAC integra un convertitore digitale-analogico a due canali con uscita in tensione a 12 bit. Il DAC può essere configurato in modalità a 8 o 12 bit. Nella modalità a 12 bit, i dati possono essere allineati a sinistra (Left shifting) o a destra (Right shifting) del Data Hold register. Il DAC dispone di due canali di uscita, ciascuno con il proprio convertitore. Nella modalità a doppio canale, le conversioni possono essere eseguite indipendentemente o simultaneamente [14]. La Fig. 6.1 mostra lo schema a blocchi della periferica DAC.

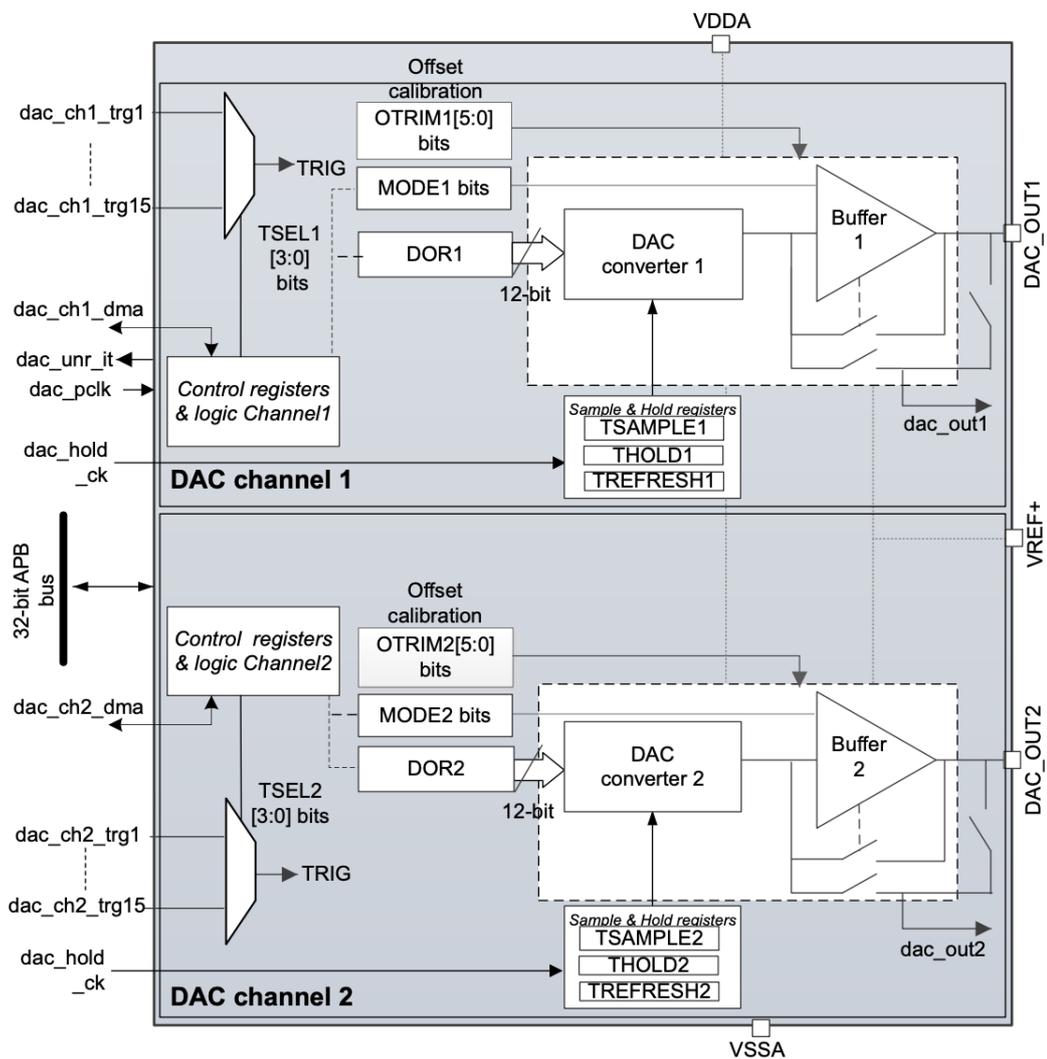


Figura 6.1: Schema a blocchi periferica DAC [14]

Le conversioni possono iniziare o scrivendo all'interno del Data Hold Register oppure possono essere triggerate da:

- da un timer interno
- da un trigger esterno al μC
- settando il bit trigger software

Una funzionalità molto utile è quella di utilizzare la periferica DMA per alleggerire il carico della CPU. Una volta rilevato una trigger request, il valore del Data Hold register viene trasferito al Data Output register.

Nel presente lavoro di tesi, la periferica interna è stata utilizzata per vedere delle forme d'onda analogiche campionate dagli ADC esterni per le acquisizioni, come approfondito nel capitolo "Gestione acquisizioni". Si riportano le funzioni utilizzati per far partire la periferica DAC e per settare il valore in uscita; le sopracitate funzioni sono contenute all'interno di librerie presenti di default dopo aver generato il codice con il software STM32CubeMX.

Algoritmo 1 Funzioni per impostare lo start ed il valore nel registro del DAC interno

```
HAL_DAC_START(DAC_HandleTypeDef *hdac, uint32_t Channel)
```

```
HAL_DAC_SetValue(DAC_HandleTypeDef *hdac, uint32_t  
                 Channel, uint32_t Alignment, uint32_t Data)
```

dove:

- *hdac: puntatore alla struttura DAC_HandleTypeDef che contiene le informazioni di configurazione per lo specifico DAC
- Channel: canale DAC selezionato
- Alignment: allineamento del dato; settabile tra: Right Alignment e Left Alignment
- Data: dato da caricare nel Data Hold register

6.2 DAC esterno

Dato il numero esiguo di canali della periferica DAC interna al μC , si è optato per l'uso di un DAC esterno. In particolare, per le prove sperimentali è stato utilizzato il DAC MCP4725 ad un canale, ma nella scheda di controllo finale verrà implementato il DAC MCP4728 a quattro canali con package MSOP, riportato in Fig: 6.3. Si è utilizzato il DAC MCP4725 in quanto si aveva già a disposizione in laboratorio ed inoltre è della stessa famiglia di convertitore analogico digitali del DAC MCP4728. D'ora in poi si farà esclusivamente riferimento al DAC MCP4725.

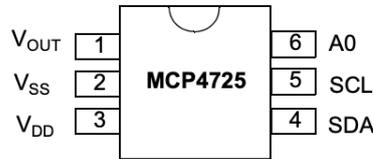


Figura 6.2: Package DAC MCP4725 [15]

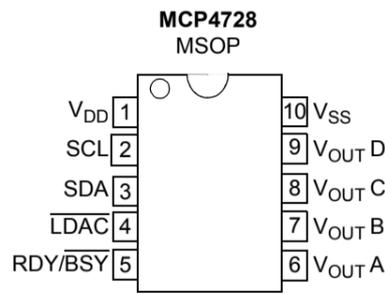
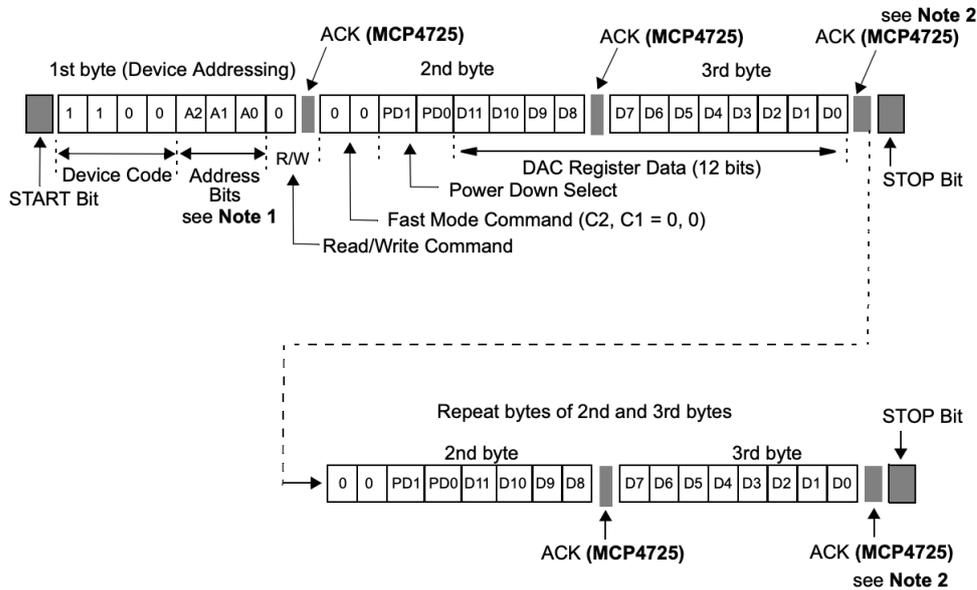


Figura 6.3: Package DAC MCP4728 [16]

L'MCP4725 è un convertitore da digitale ad analogico a singolo canale a 12 bit con memoria non volatile EEPROM. L'ingresso del DAC e i dati di configurazione possono essere programmati nella memoria non volatile (EEPROM) dall'utente tramite interfaccia I2C. La funzione di memoria non volatile consente al dispositivo di contenere i dati anche durante il tempo di spegnimento e quindi con l'uscita disponibile immediatamente dopo l'accensione. L'MCP4725 dispone di una selezione bit di indirizzo A0 esterna pin. Questo pin A0 può essere collegato a VDD o a GND dall'utente; questo permette di poter utilizzare due MCP4725 sullo stesso bus. L'MCP4725 dispone di un bus I2C a due fili e supporta le modalità standard (100 kHz), Fast (400 kHz) o High Speed Mode (3,4 MHz). I test sono stati effettuati con $f_{clock}=1000$ kHz poichè è la massima frequenza supposta dalla periferica I2C del microcontrollore. Trovandosi nel caso $f_{clock} > 100kHz$, si riporta (Fig. 6.4) il modo di funzionamento nel caso si voglia scrivere all'interno del registro del DAC in Fast Mode o High Speed Mode.



- Note 1:** A2 and A1 bits are programmed at the factory by hard-wired, and A0 bit is determined by the logic state of A0 pin.
- 2:** The device updates V_{OUT} at the falling edge of the ACK pulse of the 3rd byte.

Figura 6.4: Write commands per DAC input register e EEPROM[15]

dove:

- Il primo byte identifica l'indirizzo del device (primi 7 bit) e l'ottavo bit può assumere 0 (write command) o 1 (read command). Per il dispositivo in esame l'indirizzo è 0x62 in esadecimale
- Il secondo byte è composto come segue: primi due bit sono posti a 0 (fast mode command); terzo e quarto bit identificano il power down mode (nel caso in esame sono posti a 0 in quanto si è scelto il normal mode); i restanti 4 bit corrispondono ai primi 4 bit della parola (12 bit totali) da convertire in analogico
- l'ultimo byte corrisponde ai restanti 8 bit della parola digitale da convertire
- si identificano i segnali di Acknowledgment (ACK) che indicano che il byte è stato trasferito correttamente
- dopo aver trasferito la prima parola per intero, si possono trasferire altre word oppure inviare un segnale di stop e terminare la conversione

Validazione sperimentale DAC I2C

Il setup sperimentale è riportato in 6.5 ed è costituito, procedendo in senso orario, da: notebook, oscilloscopio, generatore di funzioni, DAC MCP4725 ed infine dalla NUCLEO-H745ZIQ.

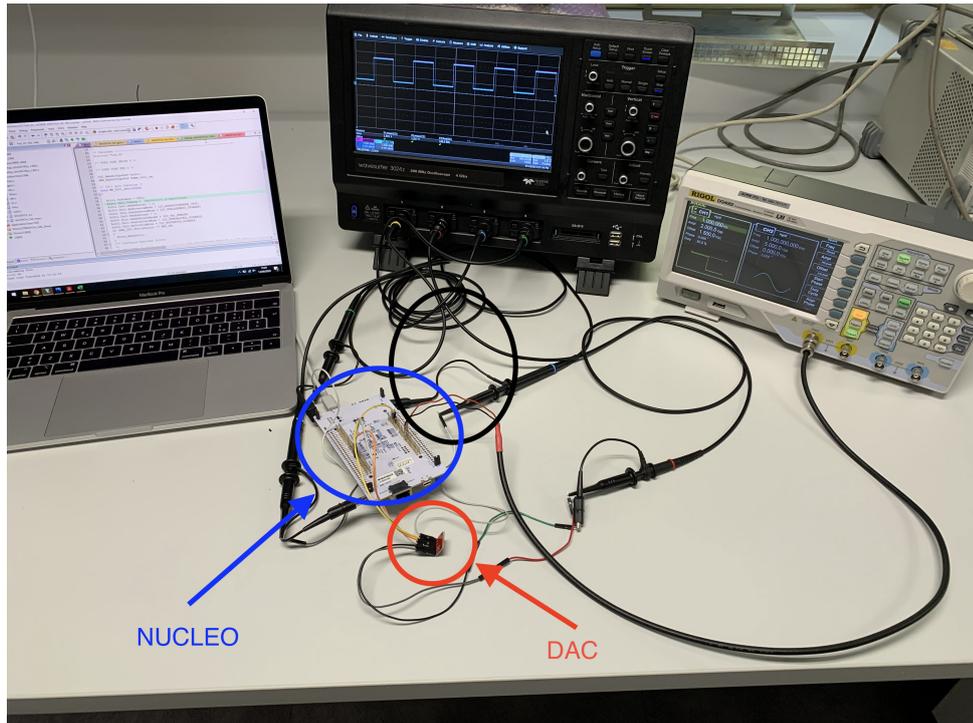


Figura 6.5: Setup sperimentale validazione DAC esterno

I test sono stati effettuati nel seguente modo:

1. Si è utilizzato il generatore di funzioni per generare un segnale analogico, con caratteristiche variabili in termini di forma d'onda e frequenza
2. Tale segnale viene campionato da un canale della periferica ADC interna al microcontrollore
3. Una volta campionata, la parola digitale viene suddivisa in due byte separati
4. I due byte vengono mandati tramite il bus I2C al DAC MCP4725, insieme all'indirizzo che identifica il device
5. Si visualizza la forma d'onda disponibile sui terminali di uscita dal DAC esterno

In Fig. 6.6 si può osservare una singola word inviata al DAC. Le forme d'onda sono state ottenute per mezzo del software Saleae Logic. Si distinguono lo start (in verde) e lo stop (in rosso) della comunicazione tra master (μC) e slave (DAC esterno) ed i segnali di clock SCL e del dato SDA.



Figura 6.6: Andamenti di SDA e SCL della comunicazione I2C

Le successive figure sono state invece ottenute con l'oscilloscopio per mezzo della logica sopra riportata.

Si riportano in Fig. 6.7 e Fig. 6.8, i casi corrispondenti ad un segnale analogico in ingresso, rispettivamente, pari a 50 Hz e 300 Hz. La curva blu rappresenta il segnale analogico in ingresso proveniente dal generatore di funzioni, mentre la curva rossa (in secondo piano) identifica l'uscita del DAC.

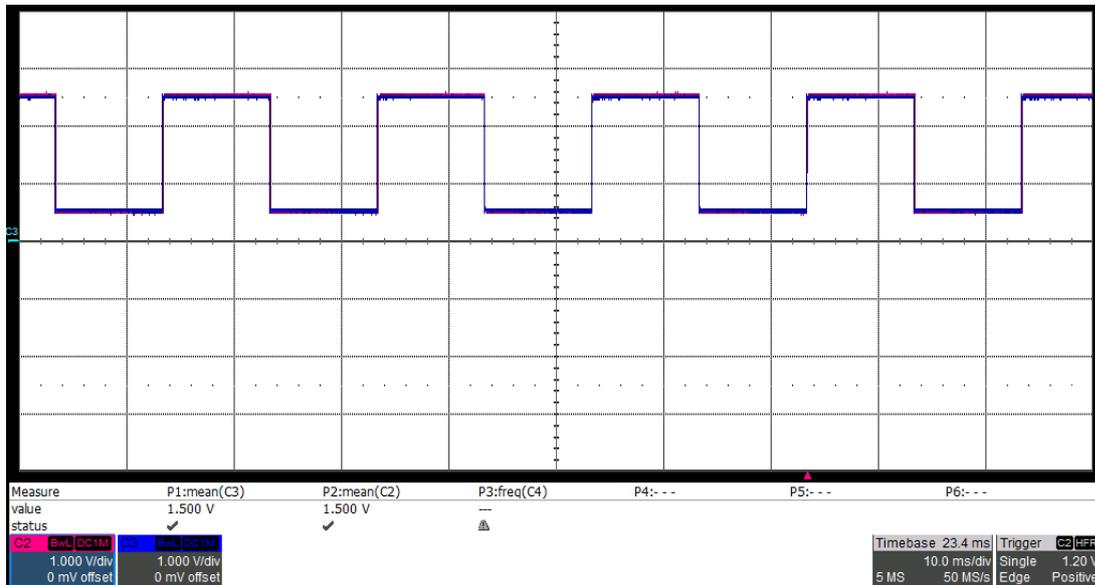


Figura 6.7: Andamenti di: segnale analogico in input (linea blu) e uscita DAC (linea rossa), onda quadra a 50 Hz, DAC MCP4725

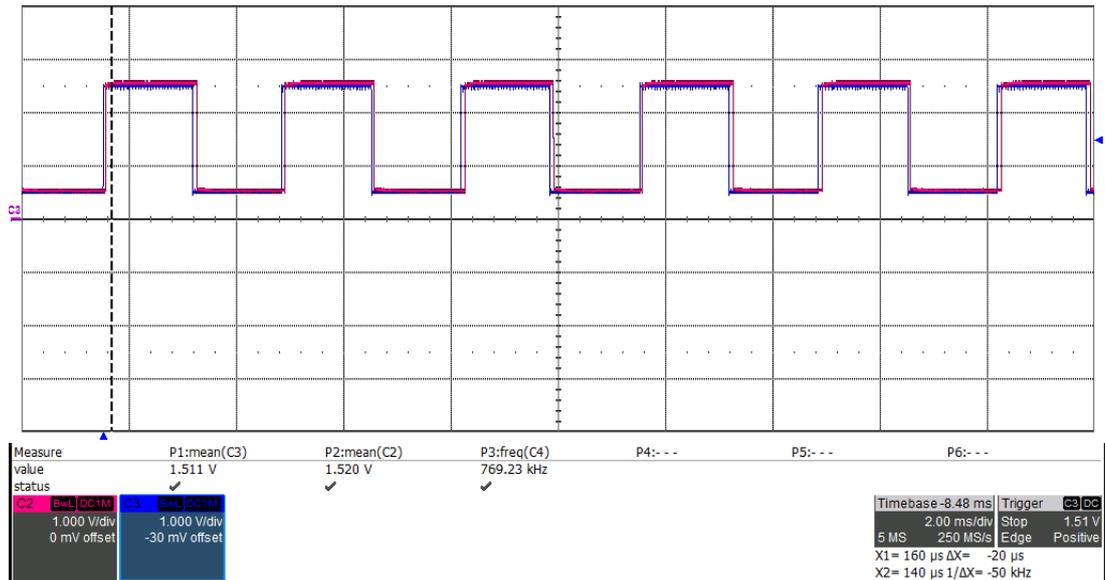


Figura 6.8: Andamenti di: segnale analogico in input (linea blu) e uscita DAC (linea rossa), onda quadra a 300 Hz, DAC MCP4725

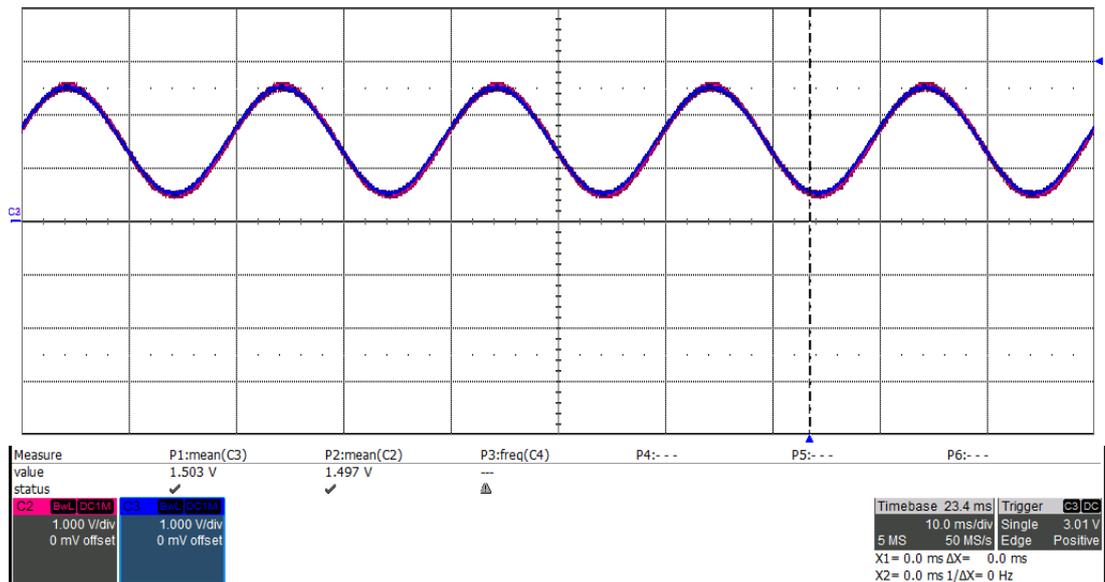


Figura 6.9: Andamenti di: segnale analogico in input (linea blu) e uscita DAC (linea rossa), onda sinusoidale a 50 Hz, DAC MCP4725

Nelle Fig. 6.9 e 6.10 si può osservare, con due forme d'onda differenti alla frequenza di 50 Hz, la corretta configurazione delle periferiche del microcon-

trollore relative alle acquisizioni da parte dell' ADC e delle comunicazioni con il DAC esterno I2C; questo è possibile dedurlo dal fatto che le forme d'onda ottenute in questa fase di validazione sono del tutto uguali. Inoltre, si è utilizzata la periferica DMA (Direct Memory Access) per alleggerire il lavoro alla CPU spostando il dato dalla memoria alla periferica I2C in Normal Mode.

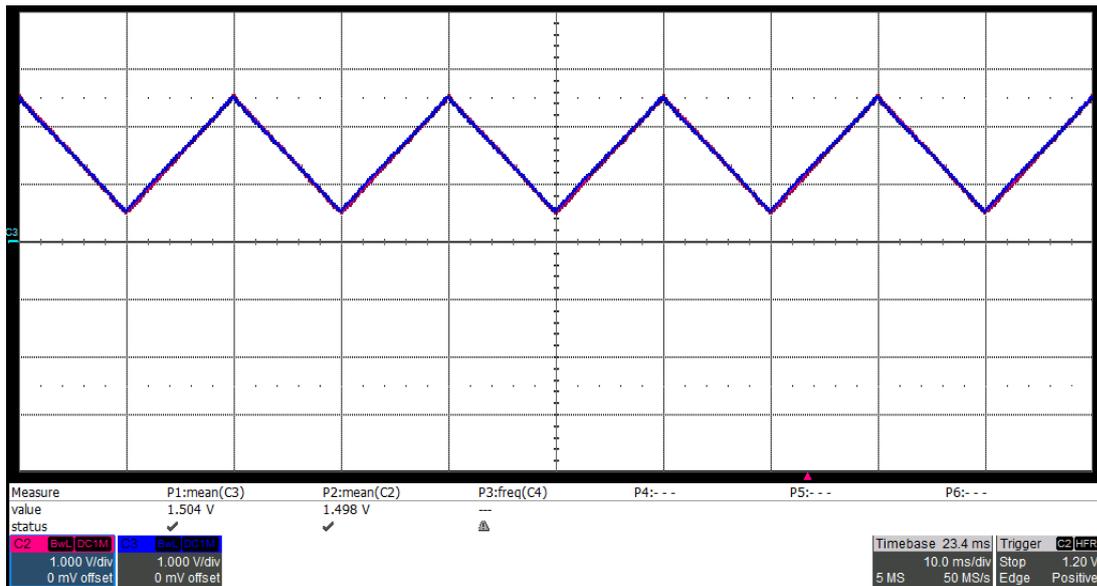


Figura 6.10: Andamenti di: segnale analogico in input (linea blu) e uscita DAC (linea rossa), onda triangolare a 50 Hz, DAC MCP4725

Capitolo 7

Gestione PWM

7.1 PWM Active Front-End 3LTT

Nella primissima parte della tesi ci si è soffermati sul funzionamento della tecnica di modulazione Pulse Width Modulation, PWM, applicata al caso di un convertitore in grado di realizzare tre diversi livelli di tensione sull'utenza. Inoltre, si è preso come riferimento una sola fase, in quanto la trattazione è identica per tutte le tre fasi.

La Fig. 7.1 mostra il caso di un convertitore T-Type 3 livelli monofase. Tra tutte le possibili combinazioni date dalla chiusura o dall'apertura dei quattro interruttori solo tre di queste hanno un interesse pratico, ossia quelle che permettono di ottenere i tre diversi livelli di tensione sul carico. Questi sono identificati con lo stato P, stato N e stato M, messi in evidenza nella Fig. 7.2.

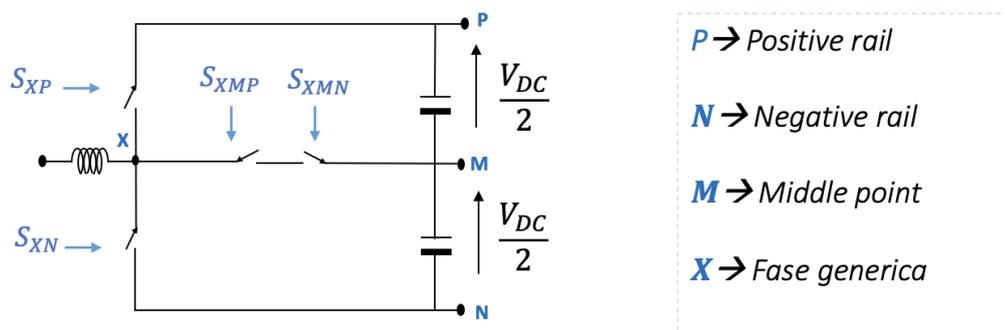


Figura 7.1: Convertitore 3 livelli T-Type monofase

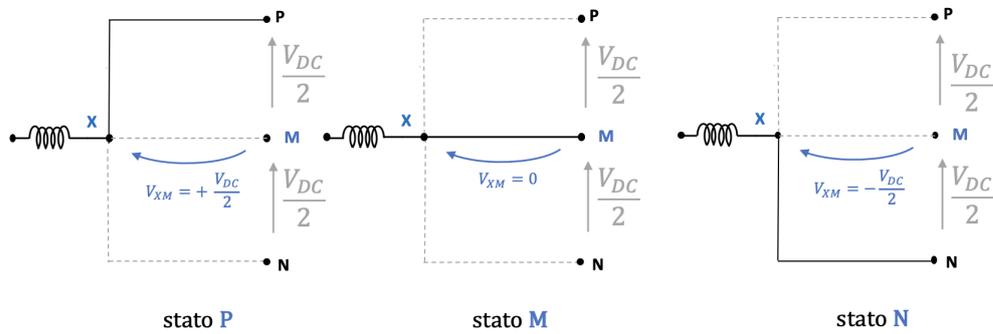


Figura 7.2: Stato P, stato M e stato N

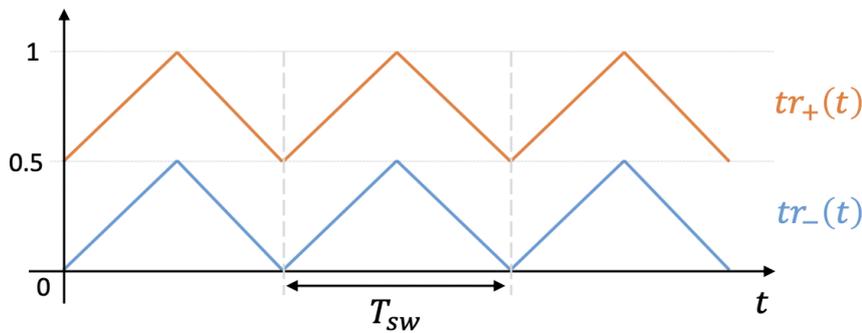


Figura 7.3: Portanti triangolari

S_{XP} , S_{XN} , S_{XMP} e S_{XMN} rappresentano lo stato logico degli switch di potenza e possono assumere solo due valori, 0 o 1. Essi sono generati confrontando una modulante $d(t)$ con due portanti triangolari $tr_+(t)$ e $tr_-(t)$. Le portanti sono rappresentate da due forme d'onda triangolari isoscele, di cui $tr_-(t)$ varia da 0 a 0.5 e $tr_+(t)$ varia da 0.5 ad 1, come mostrato in Fig. 7.3. L'importanza di tali andamenti, in fase tra loro, è rivestita dal fatto che contengono l'informazione della frequenza con cui commutano gli switch.

Vengono riportati gli andamenti nel tempo delle portanti, del duty-cycle e dei segnali logici degli switch nel caso di duty-cycle costantemente minore di 0.5 (Fig. 7.4), costantemente maggiore di 0.5 (Fig. 7.5) e variabile (Fig. 7.6). Come si può osservare, il primo caso corrisponde nel realizzare in uscita una tensione fase-punto medio $-\frac{V_{DC}}{2} < V_{XM} < 0$, mentre nel secondo caso ci troviamo nell'esatto opposto, ovvero $0 < V_{XM} < +\frac{V_{DC}}{2}$.

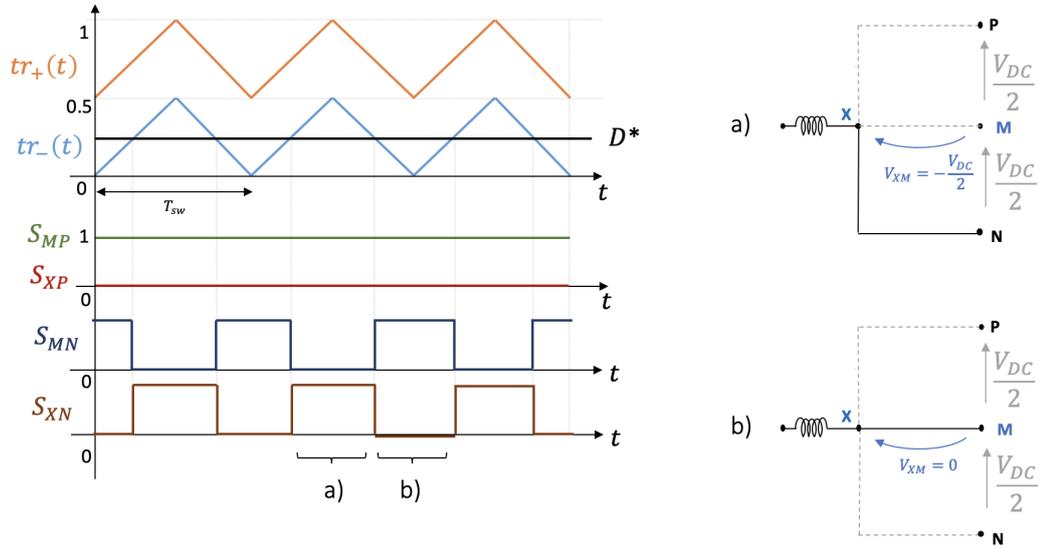


Figura 7.4: $-\frac{V_{DC}}{2} < V_{XM} < 0$

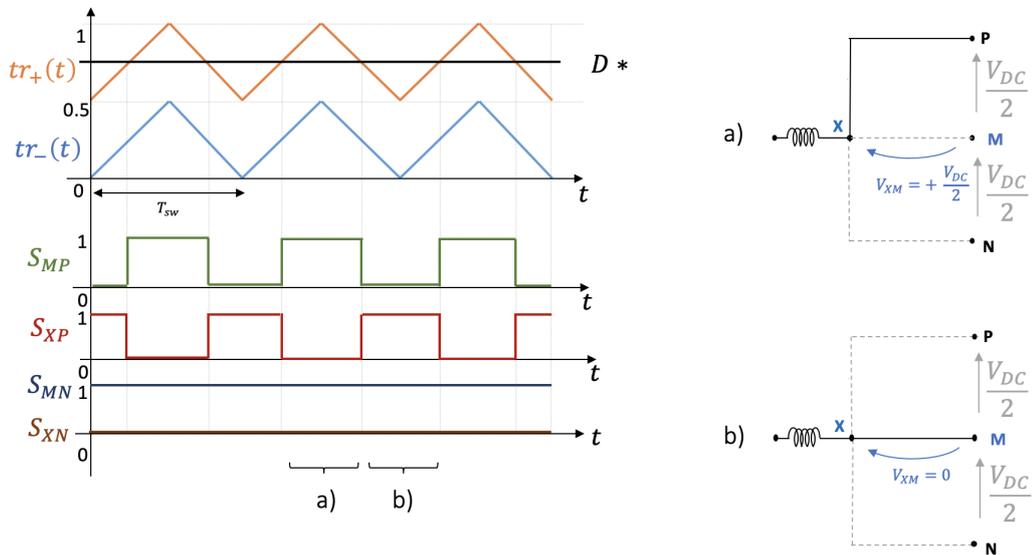


Figura 7.5: $0 < V_{XM} < +\frac{V_{DC}}{2}$

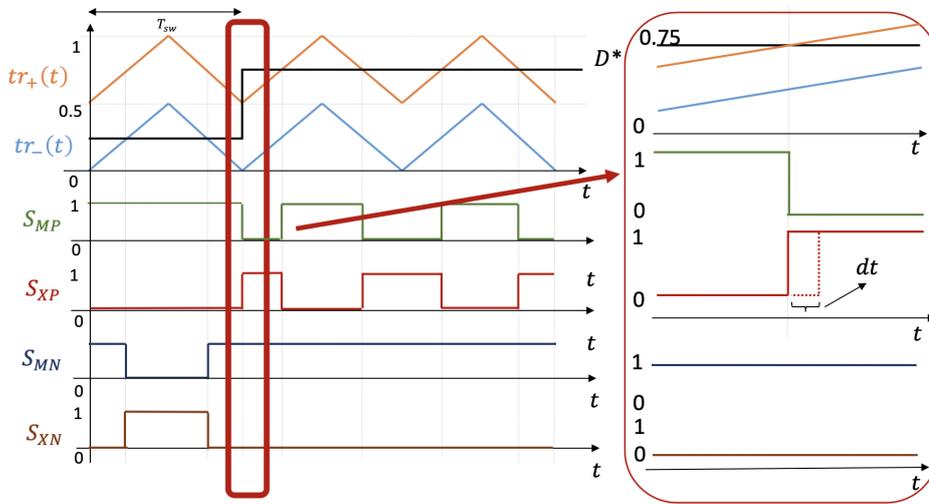


Figura 7.6: $-\frac{V_{DC}}{2} < V_{XM} < 0 \rightarrow 0 < V_{XM} < +\frac{V_{DC}}{2}$

7.1.1 Simulazione PLECS PWM 3LTT

Dopo aver effettuato un'analisi teorica sulla generazione dei segnali di gamba, si è proceduto con una simulazione al computer tramite software PLECS. Vengono riportati gli stessi andamenti di cui sopra, questa volta ottenuti in simulazione, il cui schematico è mostrato in Fig. 7.7.

La simulazione prevede l'applicazione di una modulante a $f = 50Hz$ ad onda quadra, ossia con duty-cycle del 50 %, che varia tra 0.25 e 0.75. In Fig. 7.8 viene mostrato uno zoom nell'intervallo in cui $d(t)$ è costantemente minore di 0.5 mentre in Fig. 7.9 viene mostrato uno zoom nell'intervallo in cui $d(t)$ è costantemente maggiore di 0.5.

La Fig. 7.10 mette invece in evidenza l'intervallo di tempo in cui avviene la transizione, ideale, del segnale $d(t)$ da 0.25 a 0.75. Come si può notare, le forme d'onda ottenute in simulazione sono del tutto equivalenti a quelle riportate nell'analisi teorica.

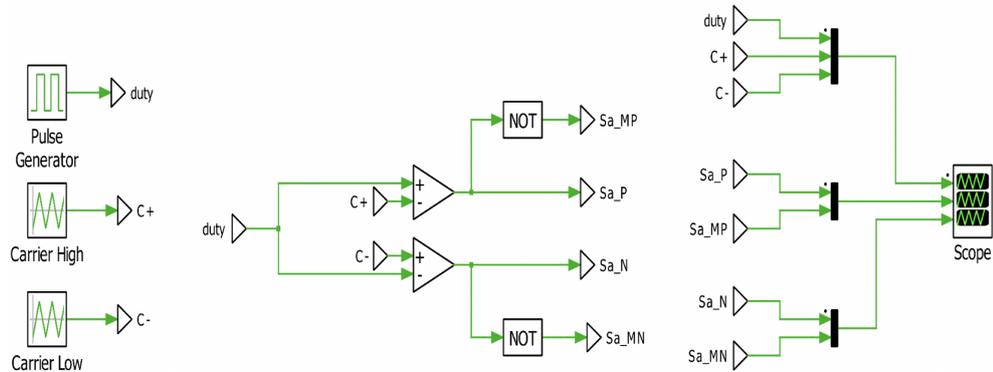


Figura 7.7: Schematico PLECS utilizzato per la simulazione

7.2 Implementazione digitale PWM 3LTT

Il primo problema affrontato in questo progetto di tesi è stato quello di dover adattare la tecnica di modulazione PWM presentata nel paragrafo precedente alle esigenze del microcontrollore. Il microcontrollore non è in grado di generare delle portanti triangolari complementari, per questo motivo si è deciso di aggirare il problema lavorando non più con due portanti ed una modulante, bensì con due modulanti ed una portante, la quale rappresenta un vincolo imposto dal μC .

La periferica timer interna al μC è in grado di generare una forma d'onda triangolare, isoscele o rettangolare, che varia da 0 fino al valore contenuto nell' ARR (Auto Reload Register). I parametri caratteristici della periferica, come evidenziato in Fig. 7.11, che bisogna impostare per definire la portante, nel caso di timer configurato in center aligned mode, sono i seguenti:

$$f_{sw} = \frac{f_{CNT}}{2ARR} \quad (7.1)$$

$$duty = \frac{CCR}{ARR} \quad (7.2)$$

dove f_{CNT} è la frequenza del contatore e CRR è il valore contenuto nel registro di Capture Compare.

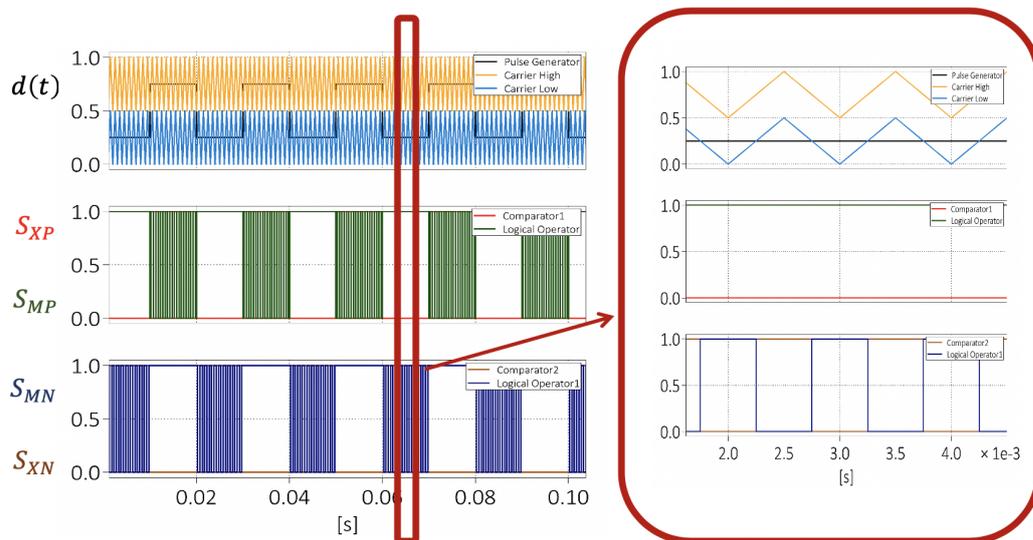


Figura 7.8: Simulazione PLECS $-\frac{V_{DC}}{2} < V_{XM} < 0$

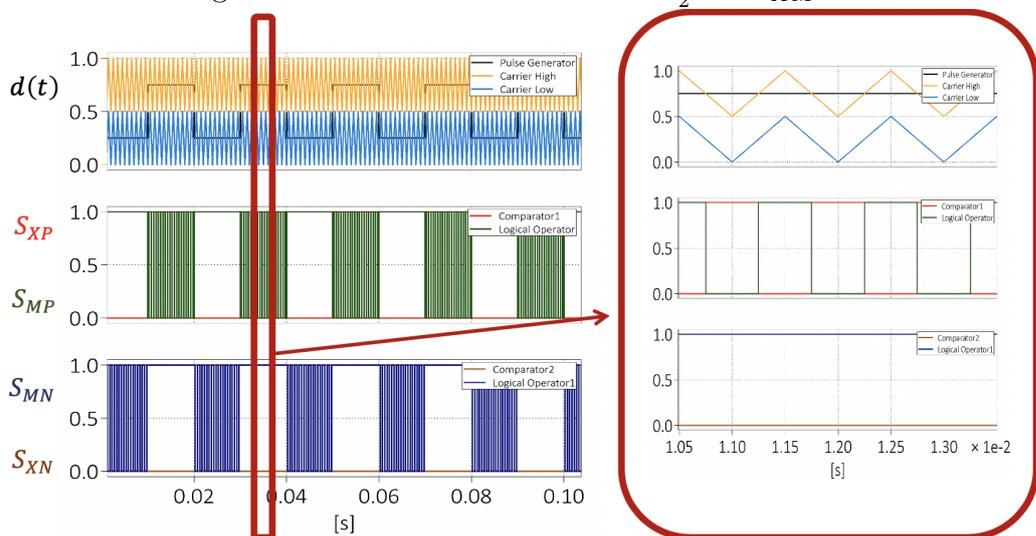


Figura 7.9: Simulazione PLECS $0 < V_{XM} < +\frac{V_{DC}}{2}$

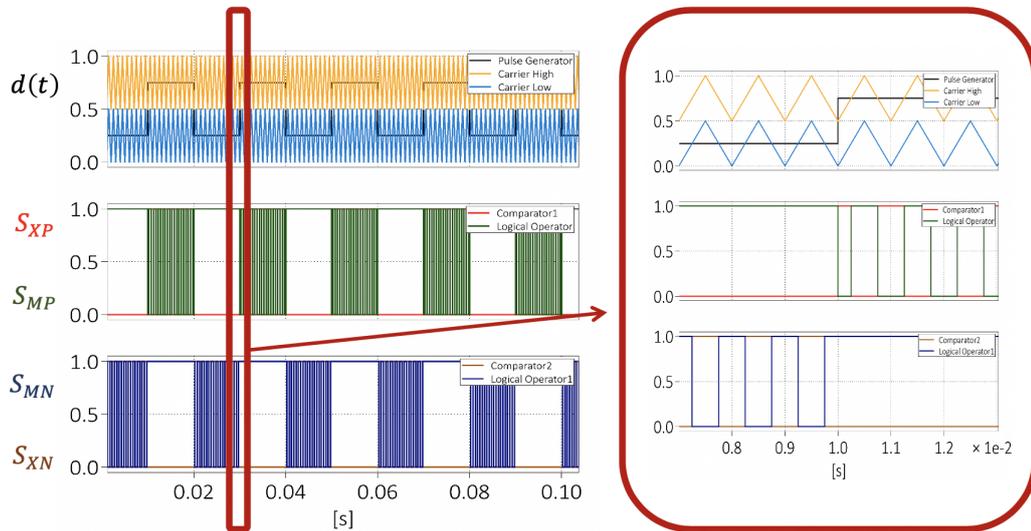


Figura 7.10: Simulazione PLECS $-\frac{V_{DC}}{2} < V_{XM} < 0 \rightarrow 0 < V_{XM} < +\frac{V_{DC}}{2}$

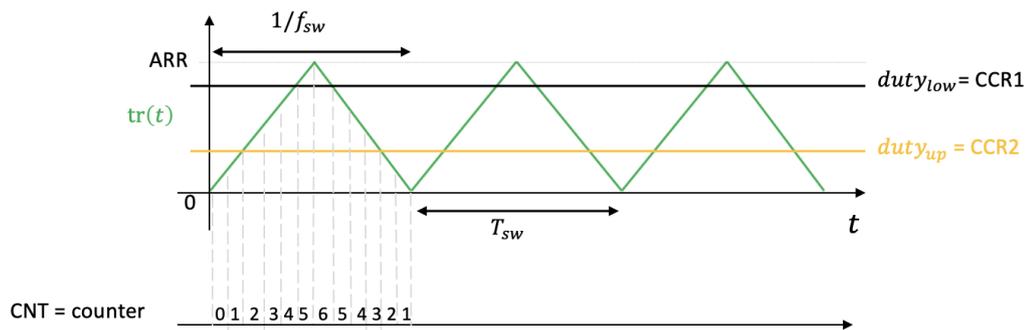


Figura 7.11: Overview counter timer

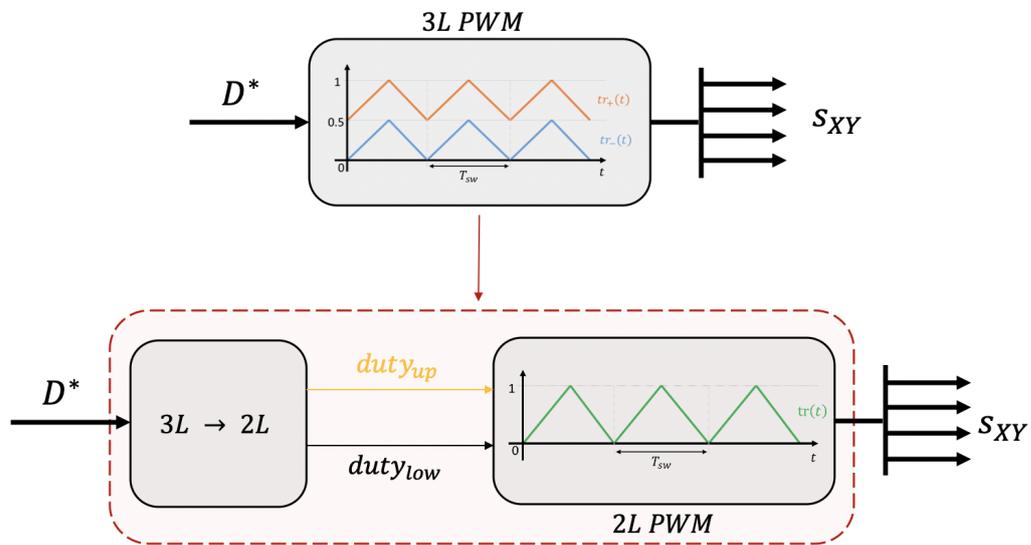


Figura 7.12: Schema a blocchi passaggio da PWM 3 livelli a PWM 2 livelli

Per poter utilizzare i timer del microcontrollore bisogna adattare il duty-cycle generato dagli anelli di controllo del convertitore. La soluzione adottata, riportata in Fig. 7.12, 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 é importante che i segnali PWM siano identici a quelli della PWM a 3 livelli.

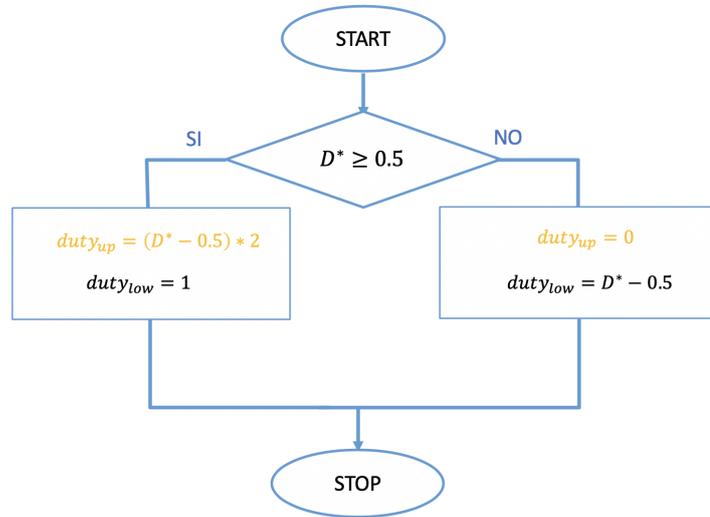


Figura 7.13: Diagramma di flusso passaggio da PWM 3 livelli a PWM 2 livelli

La Fig. 7.13 mostra l'algoritmo, sotto forma di diagramma di flusso, utilizzato per convertire il segnale di riferimento desiderato D^* in due duty-cycle equivalenti $duty_{up}$ e $duty_{low}$ necessari al suddetto scopo. Le Fig. 7.14 e 7.15 riportano gli andamenti che si ottengono a valle dell'applicazione del suddetto algoritmo.

7.2.1 Simulazione PLECS implementazione digitale

Come svolto nel paragrafo precedente, le forme d'onda ricavate durante l'analisi teorica sono confrontate con una simulazione al computer tramite PLECS.

In Fig. 7.16 è mostrato lo schematico utilizzato per la simulazione. In particolare, il blocco denominato '3L → 2L' contiene al suo interno lo script in linguaggio C mostrato nell'algoritmo 2. Tale script non è altro che l'implementazione in codice dell'algoritmo di Fig. 7.13. Esso ha come input il duty-cycle a 3 livelli di riferimento, con $f = 50Hz$ e andamento ad onda quadra con valori 0.2 e 0.8 (come esempio), e restituisce i valori di $duty_{up}$ e $duty_{low}$. Le portanti, sia nel caso del 3 livelli che del 2 livelli sono alla frequenza di switching pari a 20 kHz. Il blocco '3L → 2L', inoltre, è triggerato da una ISR anch'essa alla frequenza $f_{sw} = 20kHz$.

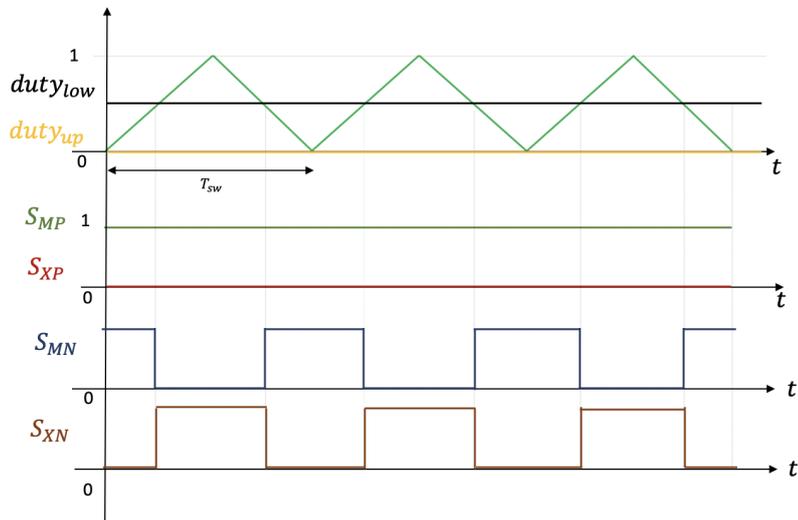


Figura 7.14: $-\frac{V_{DC}}{2} < V_{XM} < 0$

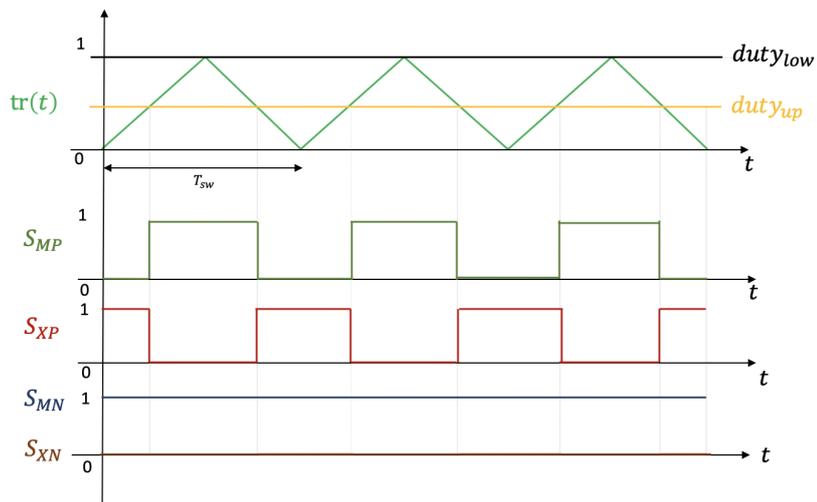


Figura 7.15: $0 < V_{XM} < +\frac{V_{DC}}{2}$

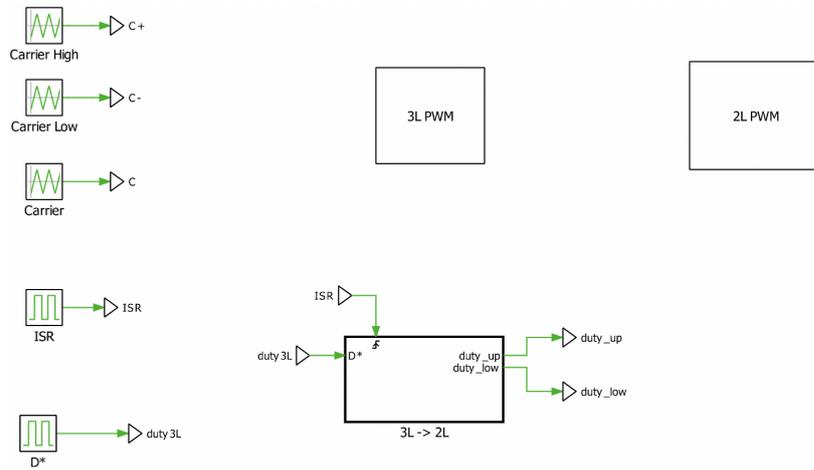
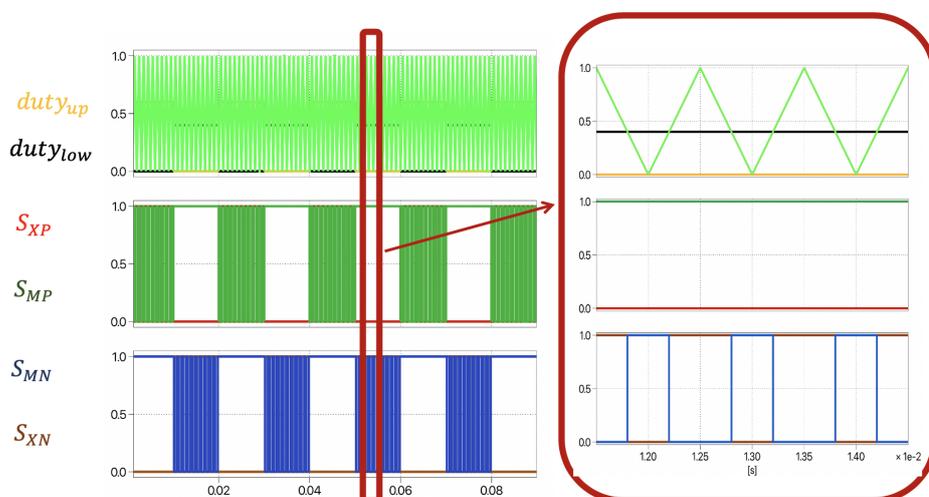


Figura 7.16: Schematico PLECS utilizzato per la simulazione duty-cycle equivalenti

Algoritmo 2 Script C blocco 3L → 2L

```

duty=InputSignal(0,0)
if (duty >= 0.5)
    {
        duty_up=(duty - 0.5) * 2;
        duty_low=1;
    }
else
    {
        duty_low=duty * 2;
        duty_up=0;
    }
OutputSignal(0,0)=duty_up
OutputSignal(0,1)=duty_low
    
```


 Figura 7.17: Simulazione PLECS $d(t) < 0.5$

In Fig. 7.17 si può notare come, quando si intende realizzare una tensione sul carico $-\frac{V_{DC}}{2} < V_{XM} < 0$, gli unici switch a commutare sono S_{XN} e S_{MN} mentre S_{MP} e S_{XP} sono tenuti rispettivamente chiuso ed aperto per tutto la durata dell'intervallo di tempo in cui $d(t) < 0.5$. In Fig. 7.18 ci si trova esattamente nel caso complementare. Infine, la Fig. 7.19 mostra il caso in cui avviene il passaggio da un duty-cycle maggiore di 0.5 al caso opposto. Inoltre, è opportuno precisare che durante tutta la trattazione, si è posto per semplicità dead time nullo tra gli switch complementari.

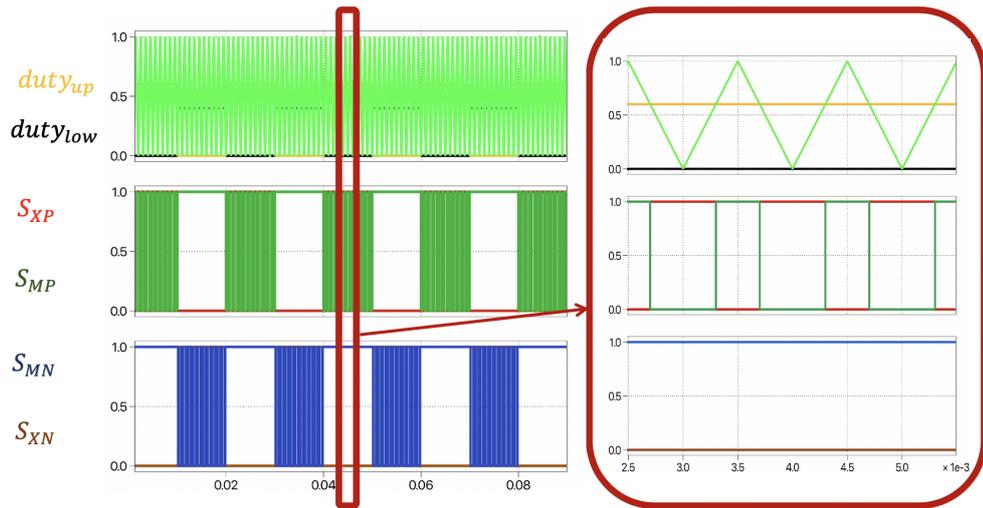


Figura 7.18: Simulazione PLECS $d(t) > 0.5$

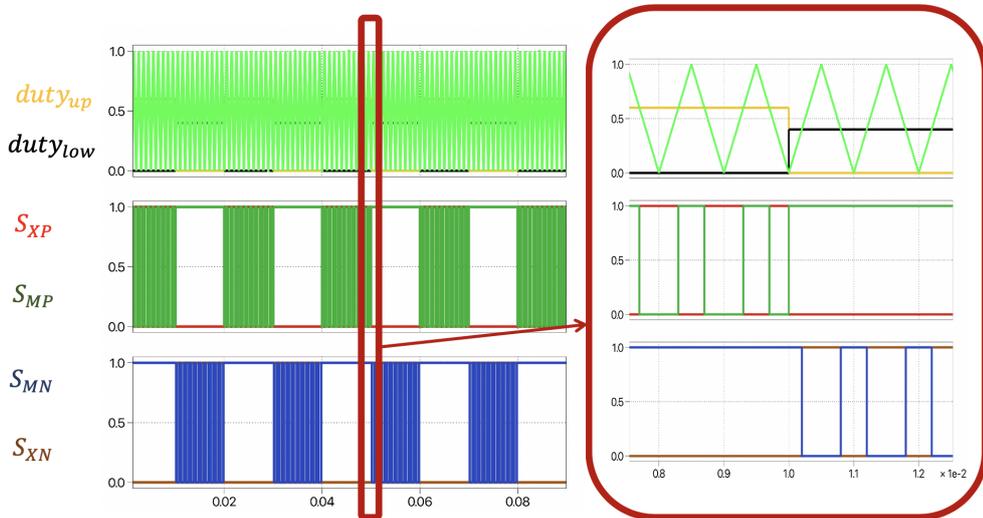


Figura 7.19: Simulazione PLECS con duty-cycle che passa da 0.2 a 0.8

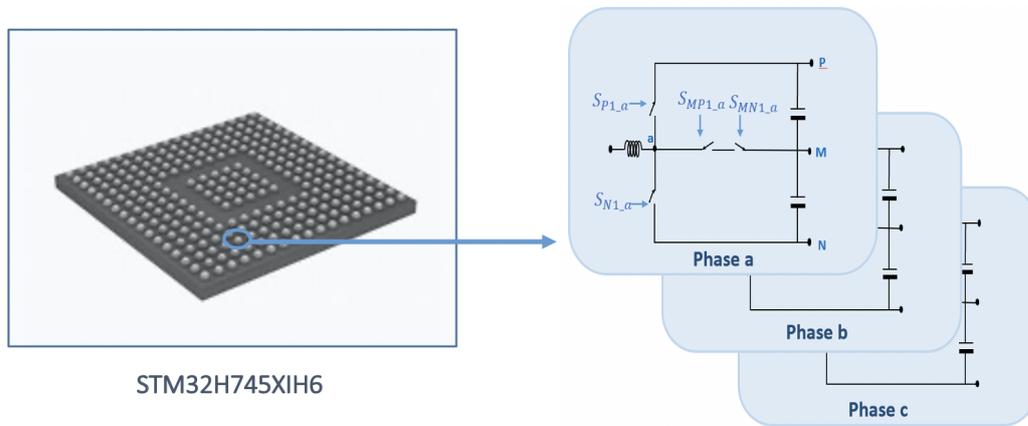


Figura 7.20: Overview gestione PWM del convertitore 1

7.3 Configurazione periferica TIMER

Il primo aspetto della configurazione del microcontrollore della famiglia STM-32H7 che è stato affrontato in questo lavoro di tesi riguarda la generazione dei segnali che identificano lo stato logico degli switch di potenza, nel caso in esame rappresentati da dei MOSFET (Metal Oxide Silicon Field Effect Transistor). Tali segnali logici vengono generati dal microcontrollore e rappresentano l'input dei gate drivers, che costituiscono il vero e proprio stadio di comando degli switch. Essendo il convertitore da controllare costituito da due sotto convertitori, ciascuno dei quali è un Active front-end T-Type a 3 livelli composto da 12 transistor da comandare, è necessario che il microcontrollore generi 24 segnali. L'alto numero di segnali da generare contemporaneamente ha vincolato la scelta del microcontrollore da utilizzare. Inizialmente la scelta è ricaduta sul microcontrollore STM32H745ZI, con package LQFP144 dove 144 è il numero di pin. Successivamente, col procedere del lavoro ci si è resi conto che le periferiche minime per generare i 24 segnali PWM non erano contemporaneamente utilizzabili sul suddetto μC e quindi la scelta è ricaduta sul microcontrollore STM32H745XI con package TFBGA240, la quale avendo 240 pin disponibili ha permesso di risolvere il problema. I test per la validazione della configurazione delle varie periferiche sono stati effettuati sulla NUCLEO-H745ZIQ, nonostante ciò, essendo i core e le periferiche identiche per entrambi i μC sopra citati, è del tutto indifferente in questo caso dove viene implementato il firmware. In Fig. 7.20 viene mostrata una overview della gestione PWM, prendendo in considerazione solamente il convertitore AFE 1 e sono nel caso monofase.

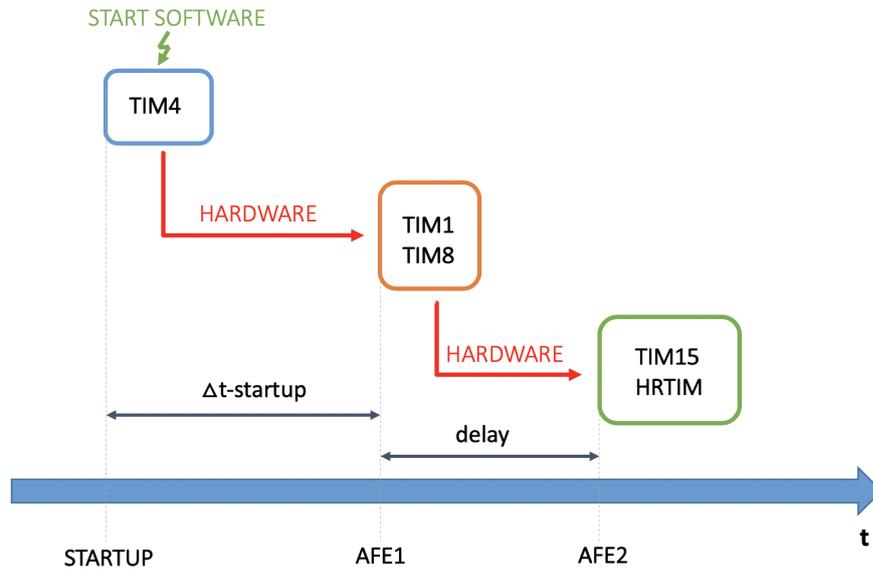


Figura 7.21: Schema a blocchi implementazione PWM

Un aspetto estremamente importante è rappresentato dalla sincronizzazione dei timer. È molto importante che i timer siano sincronizzati tra di loro e che partano in modo controllato dall'utente. In Fig. 7.21 si riporta uno schermo a blocchi in cui viene esplicitata la logica che si è seguita per la configurazione dei timer coinvolti nella gestione dei comandi PWM

La logica che si è seguita per la configurazione dei timer è la seguente:

1. Si è utilizzata la periferica TIM4, con start software, in modo da garantire che vengano effettuate determinate operazioni, come l'inizializzazione di tutte periferiche, prima di iniziare la modulazione PWM.
2. TIM4, con trigger hardware, fa partire il TIM1 e il TIM8 che costituiscono i timer responsabili della generazione dei segnali PWM del convertitore AFE 1.
3. Il TIM1, sempre con trigger hardware, fa partire il TIM15 e i timer HRTIM che costituiscono i timer responsabili della generazione dei segnali PWM del convertitore AFE 2. Si utilizza questa la soluzione in quanto si vuole che il convertitore AFE 1 ed il convertitore AFE 2 lavorino con un ritardo di fase programmabile. In maniera più dettagliata, si vedrà in seguito, i due sottoconvertitori lavorano con un ritardo reciproco di 90° .

Viene riportata la Fig. 7.25 in cui è mostrato uno schema a blocchi che riassume la logica seguita per la configurazione dei timer PWM, dove

per $\Delta t - startup$ si intende l'intervallo di tempo che intercorre tra lo start software di TIM4 e la partenza dei counter di TIM1/TIM8 e per *delay* il ritardo tra l'AFE 1 e l'AFE 2.

7.3.1 TIM4

TIM4 è un general-purpose timer utilizzato per abilitare il conteggio del TIM1 e TIM8 quando avviene l'evento di compare. In Fig. 7.22 si riporta lo schema a blocchi funzionale di un general-purpose timer, di cui TIM4 fa parte. Tale tipo di periferica è caratterizzata da un contatore a 16 bit, comune a tutti i canali, la cui 'velocità' di conteggio dipende dalla f_{clock} e dal valore del Prescaler, programmabile da 0 a $2^{16} - 1$. Si può notare nella parte alta della figura il circuito di controllo destinato alla sincronizzazione del timer con eventi esterni e con altri timer, mentre nella parte bassa si distinguono i 4 canali della periferica che possono essere impostati come 'Input capture', ossia prelevando un segnale esterno disponibile su un pin dedicato, oppure come 'Capture compare'. Quest'ultimo caso viene utilizzato per generare dei segnali in output, sul pin TIMxCH corrispondente a ciascun canale, utile come nel caso in esame per generare dei segnali PWM.

In questa applicazione TIM4 è impostato come up-counter ed abilita il TIM1 e TIM8, lanciando un trigger hardware in corrispondenza dell'evento di compare. Come si può notare dalla Fig. 7.23 esso è configurato in 'One pulse mode', ossia il counter va da 0 ad ARR-1 e si ferma. CCR1 è settato in modo da avere l'intervallo di startup desiderato:

$$startup = \frac{CCR1 - 1}{f_{clock}} \quad (7.3)$$

con la f_{clock} settata a 240 MHz per tutti i timer.

Inoltre, poiché il trigger hardware destinato ai timer del AFE 1 avviene in corrispondenza del fronte di salita del segnale OC1REF, il canale 1 di TIM4 è stato impostato in PWM mode 2. Per PWM mode 2 si intende che il segnale in uscita, in questo caso OC1REF, è alto quando il counter è maggiore di CCR1 ed è minore nel caso opposto. Essendo tale timer utilizzato solo per settare il tempo di startup, non è impostato alcun pin come output.

Si noti inoltre, dalla Fig. 7.23, che la freccia verde sta ad indicare uno start software, a differenza della freccia rossa usata per indicare un trigger hardware.

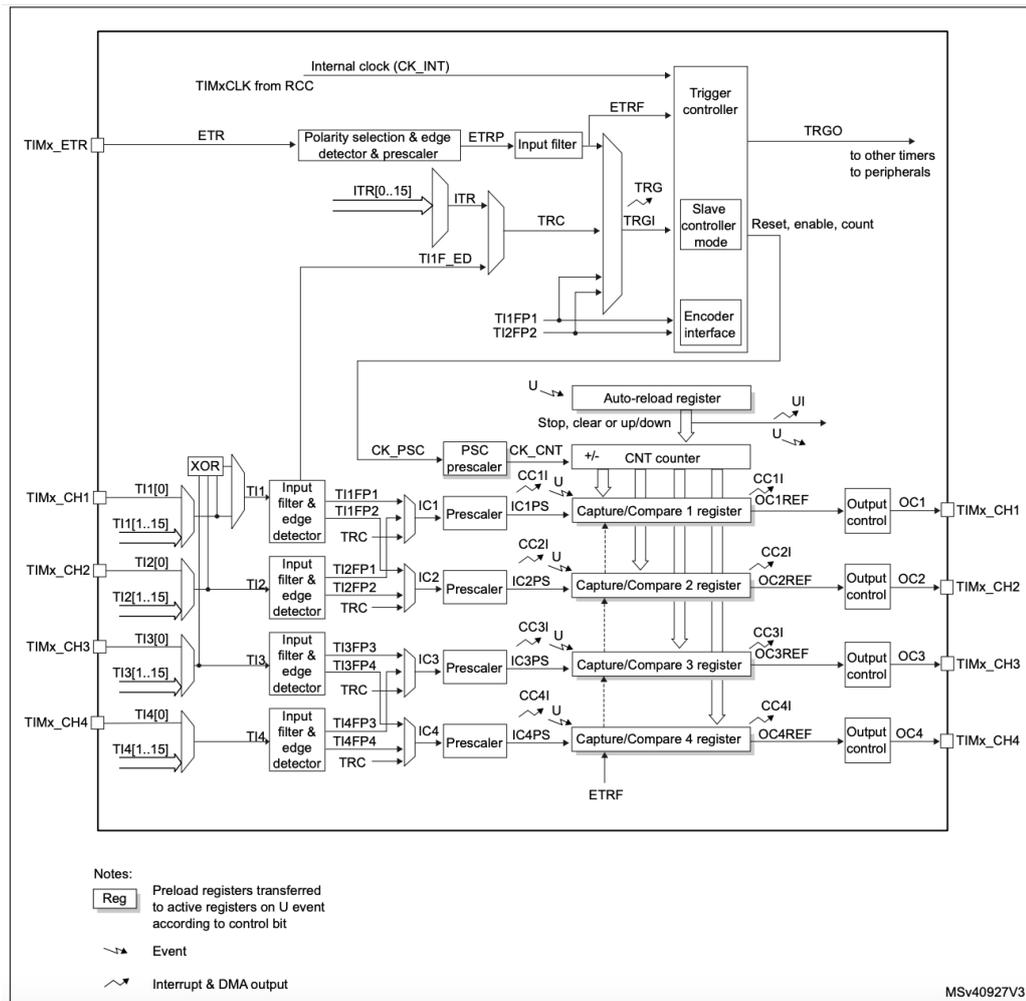


Figura 7.22: Schema a blocchi general-purpose timers [14]

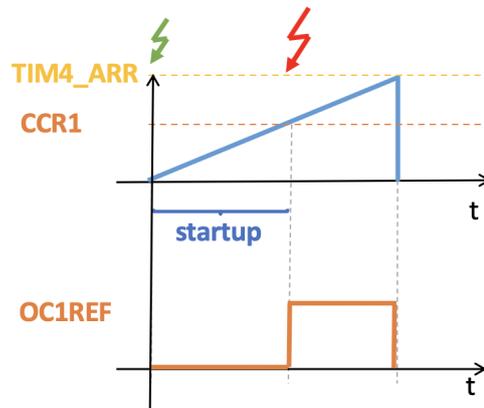


Figura 7.23: Counter di TIM4 e generazione startup time

7.3.2 TIM1-TIM8

TIM1, come anche TIM8, fa parte degli advanced-control timers. Essi sono dei timer con molti gradi di libertà in quanto, oltre alle funzioni base condivise con i general purpose timers ed i basic timers, implementano numerose funzionalità aggiuntive.

La Fig. 7.24 mostra lo schema a blocchi degli advanced-control timers di cui fanno parte TIM1 e TIM8. Sono messe in evidenza le principali differenze con le altre tipologie di timer. Si può notare la presenza di 6 canali, di cui solo i primi 4 possono realizzare in uscita dei segnali complementari, e di un Repetition counter, che conta dal valore presente nel REP register fino a 0. Si osserva inoltre la possibilità di generare dei comandi PWM trifase complementari con dead-time programmabile, blocco DTG e DTG register. Caratteristica di questa periferica è la presenza di 2 break input per forzare gli output del timer in una configurazione sicura in caso di fault.

I canali di TIM1 e TIM8 sono configurati in maniera identica per quanto riguarda la generazione dei comandi PWM, ma sia per l'uno per uno che per l'altro, viene utilizzato un quarto canale con fini differenti. Il canale 4 di TIM1 viene utilizzato per garantire la sincronizzazione reciproca tra i timer che costituiscono il sottoconvertitore AFE 2 (TIM15 e HRTIM) con quelli che costituiscono l'AFE 1 (TIM1 e TIM8). Si è optato per questa soluzione poichè la periferica HRTIM, che verrà approfondita sempre all'interno di questo capitolo, può essere comandata solo dal TIM1. Per quanto concerne invece il canale 4 di TIM8, esso è stato utilizzato per garantire la sincronizzazione delle acquisizioni delle correnti di fase generando il segnale di CS (Chip Select) destinato ad un ADC SPI esterno, come specificato in maniera più dettagliata nel capitolo 8.1.

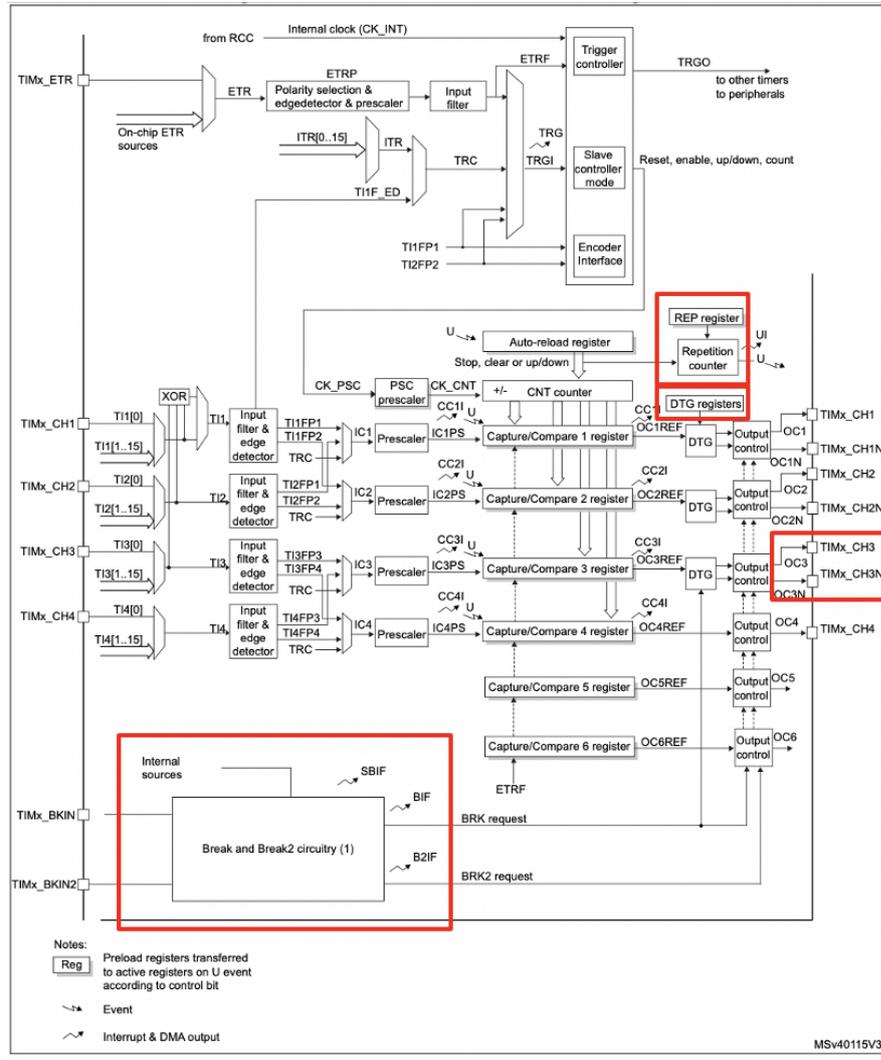


Figura 7.24: Schema a blocchi advanced-control timers [14]

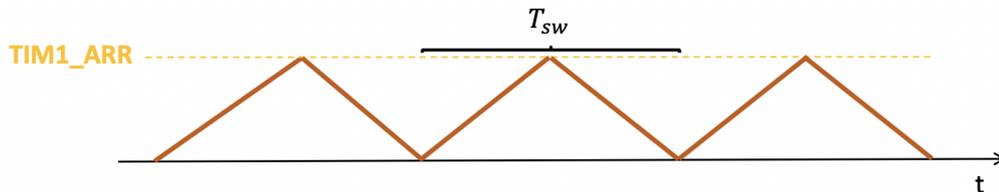


Figura 7.25: Counter di TIM1

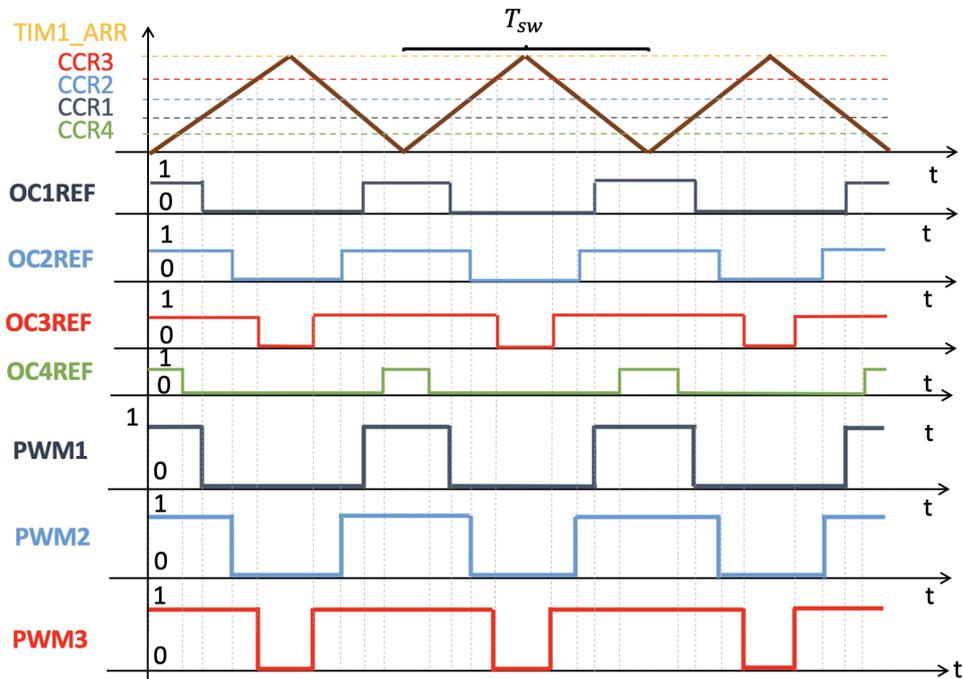


Figura 7.26: Esempio di generazione segnali PWM TIM1

TIM1 è configurato in center-aligned mode, ossia il contatore viene incrementato da zero fino ad ARR ad ogni colpo di clock e decrementato da ARR fino a 0. Tale modalità è utilizzata per simulare una portante triangolare isoscele. Nella fase di validazione per ottenere una $f_{sw} = 20kHz$ si è posto $TIM1_ARR = 6000$. Tre canali sono utilizzati per generare sei segnali PWM. Altri parametri importanti per garantire il corretto funzionamento della gestione PWM sono stati: il repetition counter posto uguale ad 1, per avere una ISR ogni periodo di switching e $PSC=0$ (Prescaler), per non rallentare il conteggio e sfruttare la massima frequenza in ingresso a tale periferica. Tutte queste considerazioni sono valide anche per il TIM8.

La Fig. 7.26 riporta un esempio di generazione dei segnali PWM per il TIM1. Come specificato in precedenza, CCR1 CCR2 e CCR3 sono utilizzati per generare segnali sul pin $TIMxCHy$, mentre CCR4 è utilizzato per settare il delay tra AFE1 ed AFE2.

7.3.3 TIM15

Definiti i 12 segnali PWM generati da TIM1 e TIM8 si è poi proceduto con la generazione degli ingressi destinati ai gate drivers del AFE 2. Dei 12 totali è stata scelta la soluzione che prevede la generazione di 2 segnali da parte di TIM15 ed i restanti 10 da parte della periferica HRTIM. La differenza sostanziale del TIM15 rispetto ai timer presentati in precedenza, come si può notare in Fig. 7.27, oltre alla presenza di 2 soli canali, risiede nel fatto che il TIM15 può essere configurato solo come up-counter.

Questo aspetto, inizialmente, ha rappresentato un ostacolo poichè i ragionamenti effettuati nell'analisi teorica e nell'implementazione dei timer del AFE 1 prevedevano l'uso di una portante isoscele, ossia configurati in center-aligned mode. Il problema è stato risolto utilizzando una modalità alternativa, ovvero configurando il canale 1 di TIM15 in modalità 'Combined PWM mode'. Si è impostato $TIM15_{ARR} = 11999$, per avere una $f_{sw} = 20kHz$, e si sono utilizzati due canali poichè il segnale 'reale' in output, come suggerisce il nome, è dato dalla combinazione di OC1REF, canale 1, e OC2REF, per il canale 2.

Come si può notare dalla Fig. 7.28 il canale 1 è configurato in 'Combined PWM mode 1', ciò significa che OC1REF è alto quando il counter è minore di CCR1 e che OC1REFC viene ottenuto effettuando una somma logica 'OR' tra OC1REF e OC2REF. Il canale 2 è invece configurato in PWM mode 2, ossia OC2REF è ottenuto seguendo la logica opposta rispetto al caso di OC1REF. I segnali realmente utilizzati per comandare i due switch di potenza desiderati, sono quelli disponibili sui pin associati al canale 1, ovvero $TIM15_{CH1}$ e $TIM15_{CH1N}$. Il canale 2 non è configurato come output.

Per rendere ciò più chiaro, viene riportato il meccanismo di funzionamento della modalità 'Combined PWM'. In particolare bisogna sottolineare che sono possibili due modalità differenti: 'Combined PWM mode 1' e 'Combined PWM mode 2'. L'unica differenza risiede nel fatto che nel primo caso:

$$OCxREFC = OCxREF \parallel OCyREF \quad (7.4)$$

mentre nel secondo caso:

$$OCxREFC = OCxREF \& OCyREF \quad (7.5)$$

Inoltre, per ottenere l'output desiderato, i due canali accoppiati devono essere configurati con differenti PWM modes: PWM mode 1 e PWM mode 2.

La Fig. 7.30 mostra in che modo sono stati ricavati i valori trascritti nel registro di capture compare, sia per il canale 1 che per il canale 2, a partire dal duty-cycle di riferimento e dal valore $TIM15_{ARR}$.

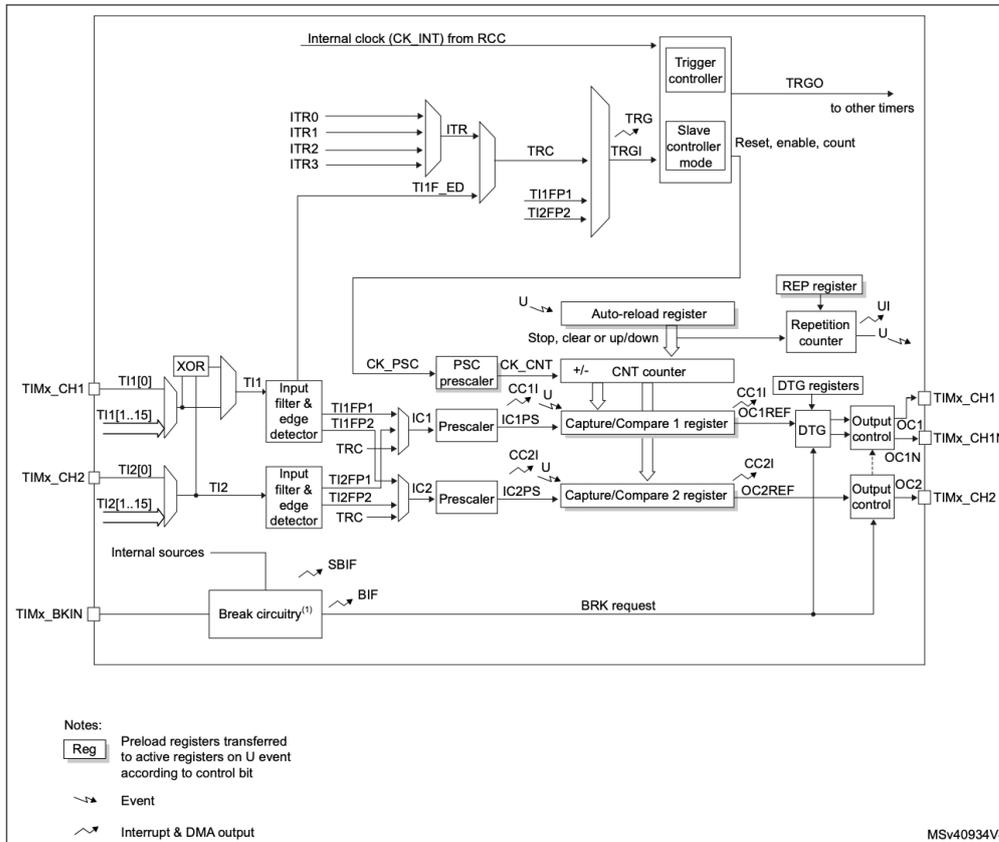


Figura 7.27: Schema a blocchi TIM15 [14]

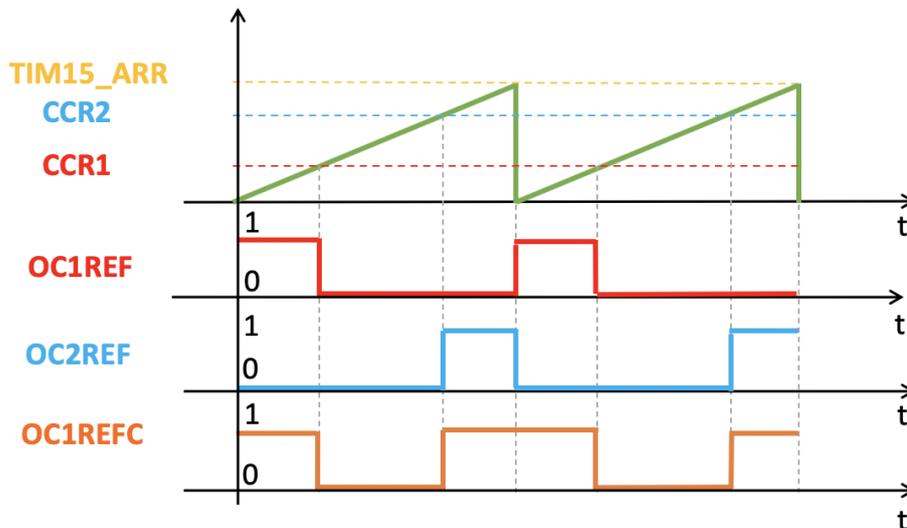


Figura 7.28: Esempio di generazione segnali PWM TIM15

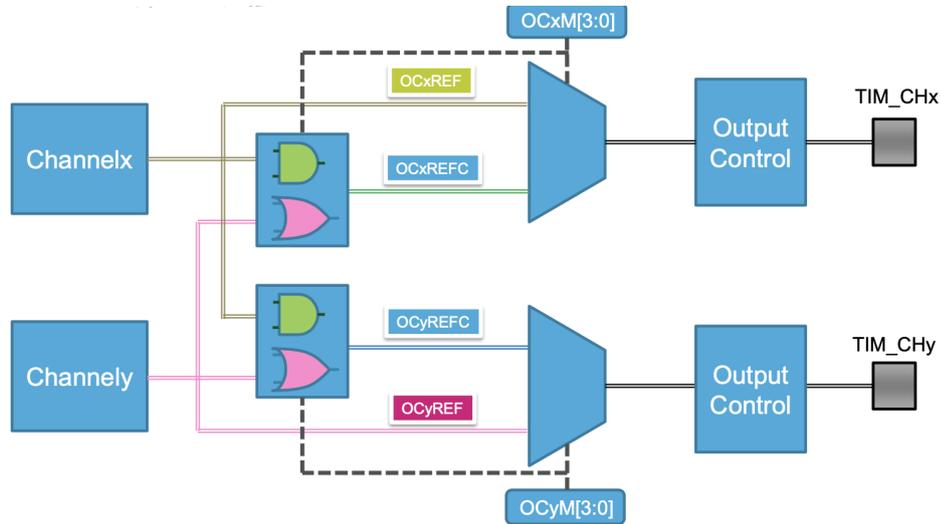


Figura 7.29: Meccanismo di funzionamento Combined PWM mode [17]

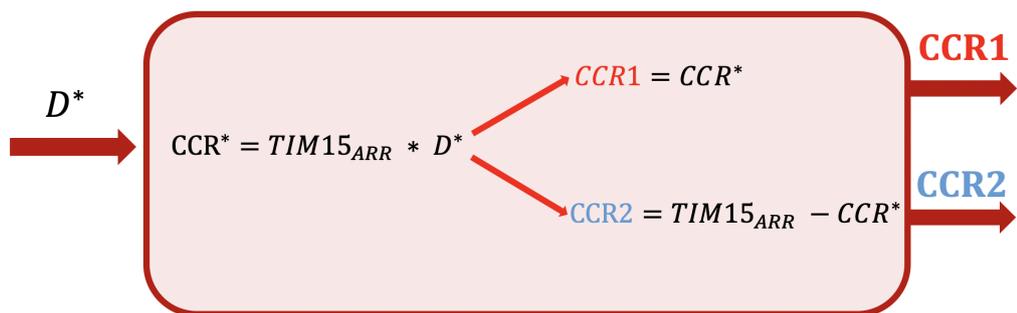


Figura 7.30: Implementazione digitale PWM TIM15

7.4 Configurazione periferica HRTIM

HRTIM, o High Resolution Timer, è una periferica molto versatile la quale è in grado di generare fino a 10 segnali digitali con tempistiche estremamente precise. È principalmente destinata a pilotare sistemi di conversione di potenza come alimentatori a commutazione o sistemi di illuminazione, ma può essere di uso generale, ogni volta che si prevede una risoluzione di temporizzazione molto fine. La sua architettura modulare consente di generare forme d'onda indipendenti o accoppiate. La forma d'onda è definita da temporizzazioni autonome (utilizzando contatori e comparatori) e un'ampia gamma di eventi esterni, come feedback e segnali di sincronizzazione. Ciò consente di produrre una grande varietà di segnali di controllo adatte alla maggior parte delle strutture di conversione.

Vengono riportate le caratteristiche principali di tale periferica:

- Risoluzione completa disponibile su tutte le uscite, possibilità di regolare il duty-cycle, la frequenza e la durata dell'impulso in one pulse mode
- 6 unità di temporizzazione a 16 bit (ognuna con un contatore indipendente e 4 comparatori)
- 10 uscite che possono essere controllate da qualsiasi unità di temporizzazione, fino a 32 sorgenti di set/reset per canale
- Più canali possono essere sincronizzati con input/output esterni

In Fig. 7.31 si possono notare i componenti funzionali più importanti di cui la periferica HRTIM è costituita. Dall'alto verso il basso sono presenti il Master Timer, le unità di temporizzazione (dal timer A al timer E), lo stadio di uscita e una logica di condizionamento del segnale di guasto e di evento esterno condivisa da tutti i timer.

Gli High Resolution Timer essendo abbastanza diversi dai timer 'standard' presentano anche una nomenclatura a sè stante. Il valore di ARR prende il nome di Period, mentre il valore di CCR prende il nome di CMP (Compare). Anche il meccanismo in cui si ottengono i segnali PWM sono completamente differenti. Ogni unità di timer (da Timer A a Timer E), può essere usata indipendentemente l'una dall'altra ed ognuna può realizzare due segnali in uscita $HRTIM_{CHx1}$ ed il suo negato $HRTIM_{CHx2}$, dove x varia da A ad E. Ogni timer, compreso il Master Timer, può essere configurato per avere fino a 4 eventi di compare, ciascuno dei quali può essere utilizzato per

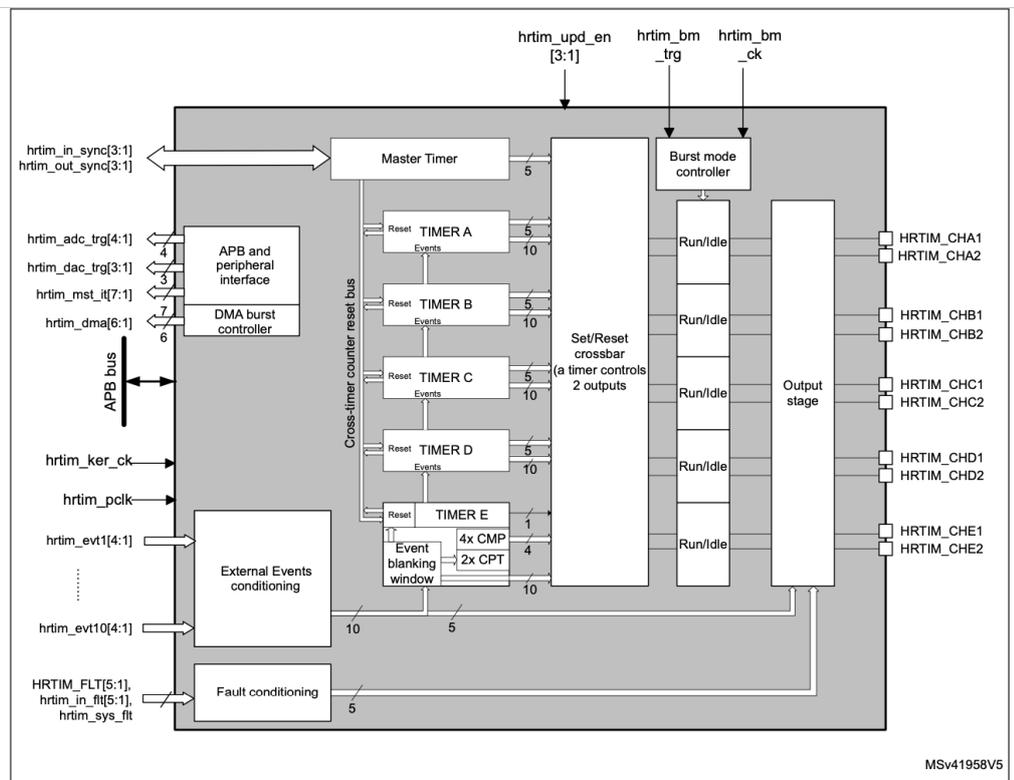


Figura 7.31: Schema a blocchi High-resolution timer [14]

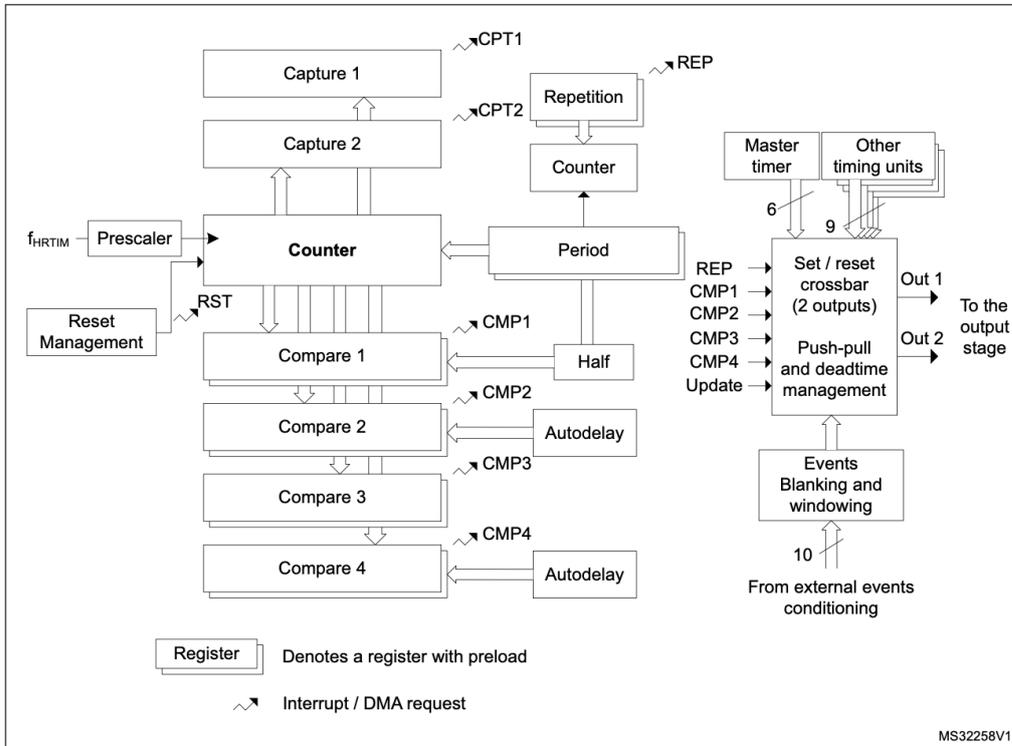


Figura 7.32: Schema a blocchi unità di timing A..E [14]

forzare l'uscita a uno (SET) o forzare l'uscita a zero (RESET). Un approfondimento del modo in cui funziona ogni singola unità di temporizzazione è riportato in Fig. 7.32.

Per quanto riguarda il presente lavoro di tesi, la periferica HRTIM è stata utilizzata per generare i restanti 10 segnali logici mancanti per comandare i gate drivers del AFE 2. Pertanto, sono stati utilizzati tutte e 5 le unità di timing, dalla A alla E, e ciascuna delle quali comprende il segnale 'positivo' ed il suo negato. Volendo una $f_{sw} = 20kHz$, ed essendo dei timer up-counter, il Period è stato settato per tutte le unità pari a 12000.

Le Fig. 7.33 mostra in maniera più esaustiva la logica utilizzata per realizzare i segnali PWM desiderati; in particolare l'evento di RESET (in rosso) avviene quando il counter è uguale a CMPy e l'evento di SET (in grigio) avviene quando il counter è uguale a CMPx; mentre la Fig. 7.34 riporta un esempio di 5 segnali PWM generati fissando, in maniera del tutto generale, degli eventi di compare. Si osserva che non sono presenti i corrispettivi segnali negati.

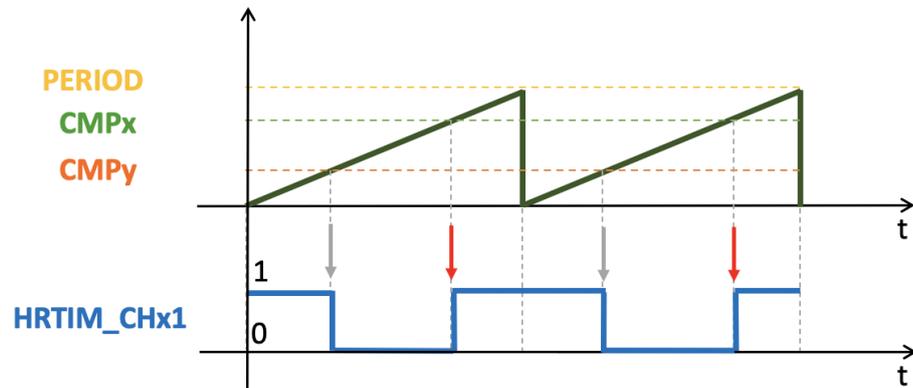


Figura 7.33: Logica di generazione segnali PWM HRTIM

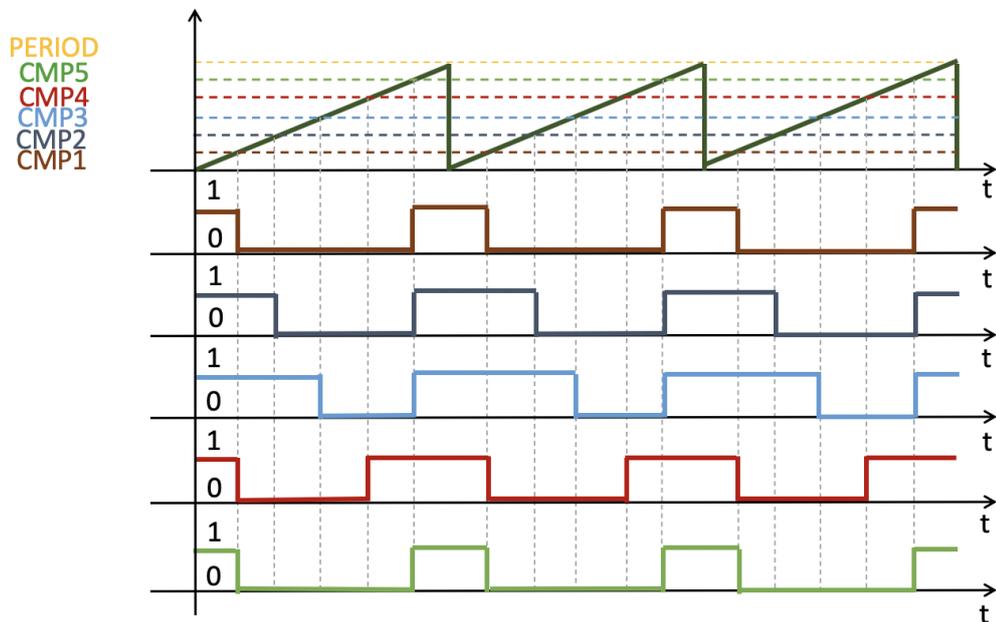


Figura 7.34: Esempio di generazione segnali PWM HRTIM

7.5 Validazione segnali PWM

A valle della configurazione delle periferiche TIM4, TIM1, TIM8, TIM15 ed HRTIM e dell'implementazione del codice tramite l'IDE Keil μ Vision, si è proceduto con la fase di testing delle forme d'onda. In particolare, ci si è concentrati principalmente sul controllare che tutti i timer partissero con il conteggio nei momenti prestabiliti. Questo aspetto è molto importante, perchè nel caso in esame il numero di switch da comandare è elevato ed è necessario che tutti i ritardi siano corretti e gestibili tramite software in maniera agevole. Si è predisposto a questo scopo un file denominato 'UserSetup.h' in cui l'utente può modificare i seguenti parametri:

- Frequenza di clock dei timer, posta a 240MHz.
- Ritardo tra il convertitore AFE 1 ed il convertitore AFE 2, definito in μ s.

Si riportano, nell'algoritmo 3 le linee di codice contenute nel file 'UserSetup.h' relative alla sincronizzazione tra i due convertitori. Nella stessa figura si può notare anche il calcolo di $DELAY_{COMPARE}$; esso rappresenta il numero di conteggi che il counter di TIM1 effettua prima di lanciare un trigger hardware che fa partire TIM15 ed HRTIM.

Algoritmo 3 Sincronizzazione tra i due convertitori UserSetup.h

```
#define STEP_TIME_SYNC_MHZ    240.0 f
#define DELAY_AFE_US         5.0 f
#define DELAY_COMPARE (uint16_t)(DELAY_AFE_US *
STEP_TIME_SYNC_MHZ)
```

Per la validazione dei segnali destinati ai gate driver si è utilizzato un analizzatore logico di Saleae, caratterizzato da una frequenza di campionamento massima di 24 MHz ed 8 canali. Per visualizzare le forme d'onda presenti sui pin di uscita dei timer si è utilizzato il software 'Saleae Logic'. Inoltre, bisogna sottolineare che nelle figure seguenti è assente il segnale PWM relativo al TIM8, poichè con la NUCLEO-STM32H745ZI non erano contemporaneamente disponibili i pin di output del TIM8 e della periferica HRTIM.

La Fig. 7.35 mette in evidenza i vertici dei timer coinvolti. Si possono notare i vertici superiore e inferiore del TIM1, essendo un timer che può contare sia up che down, ed i vertici dei timer TIM15 ed HRTIM essendo configurati come edge-aligned mode.

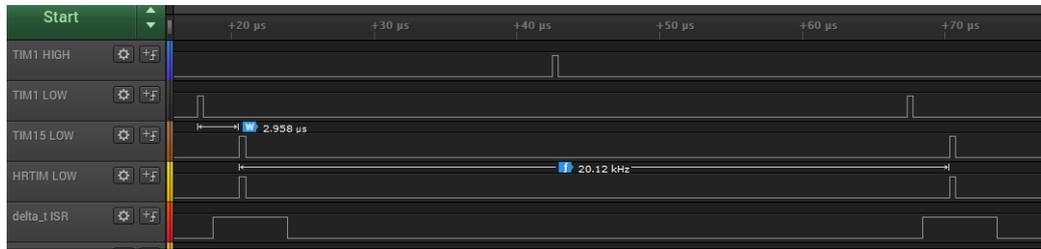


Figura 7.35: Validazione ritardo tra AFE 1 e AFE 2

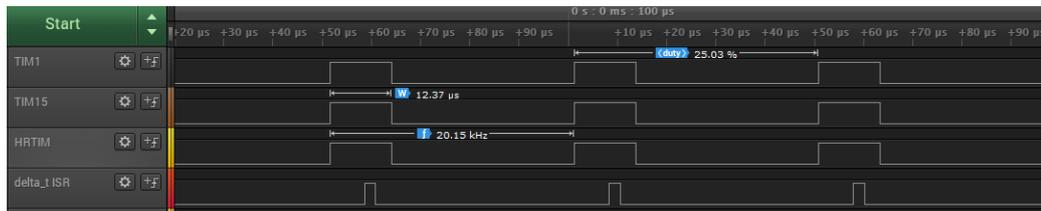


Figura 7.36: Validazione PWM con duty costante e ritardo nullo

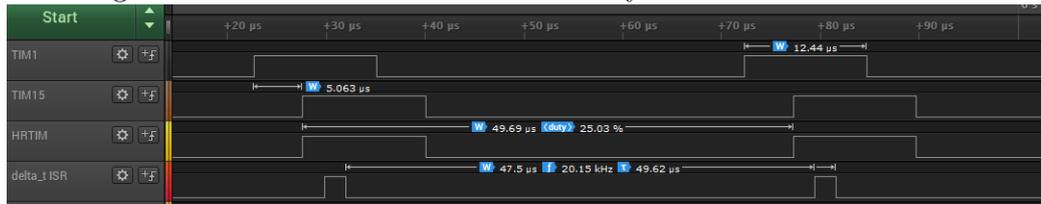


Figura 7.37: Validazione PWM con duty costante e ritardo pari a 5 μs

Viene riportato inoltre, sul quinto canale, l'andamento del pin PB0 utilizzato per valutare la durata della ISR in secondi. Tale durata corrisponde con l'intervallo di tempo in cui il pin PB0 è alto. È stato impostato un ritardo di 3 μs tra AFE 1 ed AFE 2 ed la frequenza di switching pari a 20 kHz.

Le Fig. 7.36 e Fig. 7.37 riportano i segnali generati sui pin di uscita del TIM1, TIM15 ed HRTIM nel caso di $f_{sw} = 20kHz$ e nel caso in cui essi realizzano un duty-cycle costante pari al 25%. Dai valori di f_{sw} e duty-cycle imposti si può riscontrare che l'intervallo in cui i pin sono alti è pari a circa 12.5 μs. Il fatto che non sia esattamente uguale ad $\frac{T_{sw}}{4}$ è dovuto alle varie non idealità presenti nel sistema, in particolar modo al tempo di campionamento dell'analizzatore logico non infinitesimo. Inoltre, la differenza tra le due figure sopra riportate consiste nel fatto che nella Fig. 7.36 le uscite dei timer sono sincrone tra loro, mentre nella Fig. 7.37 è stato impostato, e realizzato, un ritardo tra i due sotto convertitori pari a 5 μs.

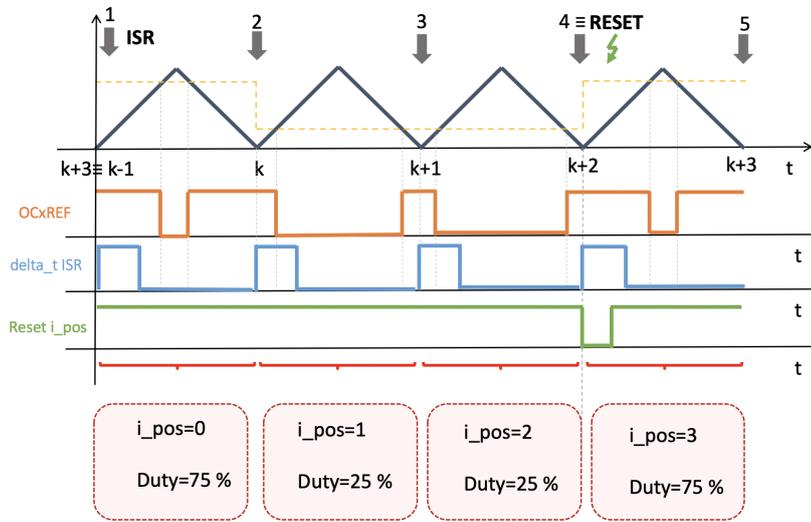


Figura 7.38: Analisi teorica PWM con duty-cycle variabile e reset i_{pos}

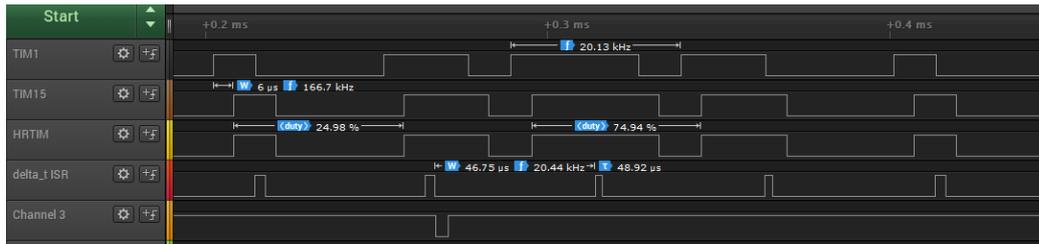


Figura 7.39: Validazione PWM con duty-cycle variabile e reset i_{pos}

Le Fig. 7.38 e Fig. 7.39 mostrano, rispettivamente, l'analisi teorica e l'acquisizione con l'analizzatore logico del caso in cui si intende realizzare un duty-cycle variabile; in particolare si è considerato un vettore di quattro posizioni in cui il duty-cycle assume per le prime due posizioni un valore del 25 % e le restanti due posizioni un valore del 75 %, $duty = 0.25, 0.25, 0.75, 0.75$. La Fig. 7.39 è stata acquisita per controllare che il reset software del contatore i_{pos} avvenisse in corrispondenza del vertice corretto. Infatti, si può osservare come la realizzazione del duty al passo di campionamento successivo rispetto alla posizione che il duty occupa all'interno del vettore. Anche in questo caso la f_{sw} è stata fissata a 20 kHz.

Si può concludere il presente capitolo dicendo che, nella prima parte ci si è soffermati sullo studio del funzionamento della tecnica di modulazione PWM nel caso di un convertitore a 3 livelli. Successivamente ci si è posti l'obiettivo di riuscire, per mezzo di una serie di operazioni aritmetiche, ad ottenere due duty-cycle equivalenti $duty_{up}$ e $duty_{low}$. Questi devono essere in grado di permettere la realizzazione degli stessi segnali, che si ottengono nel caso di una modulazione PWM a 3 livelli e che identificano lo stato logico degli switch di potenza. A conferma di ciò, si può notare che, confrontando la 7.10 con la 7.19, le forme d'onda a valle della modulazione sono identiche.

Capitolo 8

Gestione acquisizioni

8.1 Acquisizioni di corrente

Dopo aver affrontato la problematica della gestione PWM, ci si è soffermati sulla parte relativa alla gestione delle acquisizioni delle grandezze fondamentali per il controllo del sistema in esame. In particolar modo, il presente capitolo è suddiviso in due parti poichè vengono utilizzati due diversi convertitori analogico-digitale, esterni al microcontrollore, in base alla tipologia di segnale da acquisire:

- Un ADC esterno ad approssimazioni successive (SAR) per le acquisizioni di corrente
- Un ADC esterno Sigma-Delta per le acquisizioni di tensione

Nel corso del capitolo, verrà spiegato in maniera più dettagliata quali correnti e tensioni è stato necessario acquisire, in che modo si è provveduto alla configurazione delle periferiche relative a soddisfare tale scopo ed infine viene riportata la validazione delle forme d'onda.

La Fig. 8.1 riporta uno schema semplificato del sistema relativo alle acquisizioni di correnti. Avendo a che fare con due strutture di conversione trifase, il numero di correnti da acquisire è pari a 6; tali correnti, come specificato in Fig. 8.2 rappresentano l'input del convertitore analogico-digitale, responsabile della conversione del segnale analogico in una grandezza digitale, più facilmente interpretabile dal μC . Il μC , a sua volta, permette il corretto funzionamento del processo di acquisizione in quanto genera due segnali, SCLK e CS, che scandiscono il timing della conversione analogico-digitale.

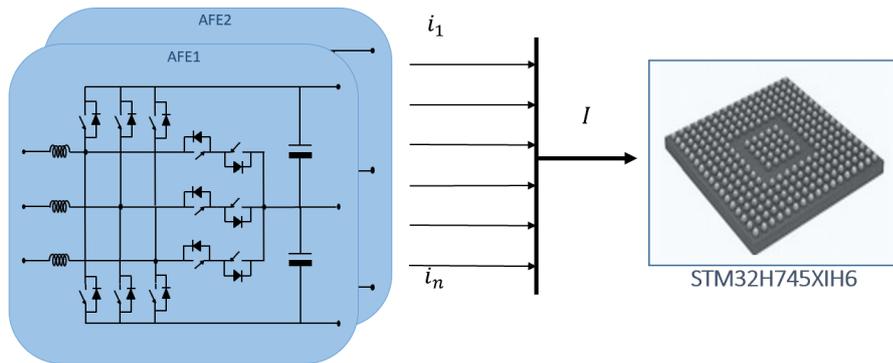


Figura 8.1: Overview gestione acquisizioni di corrente

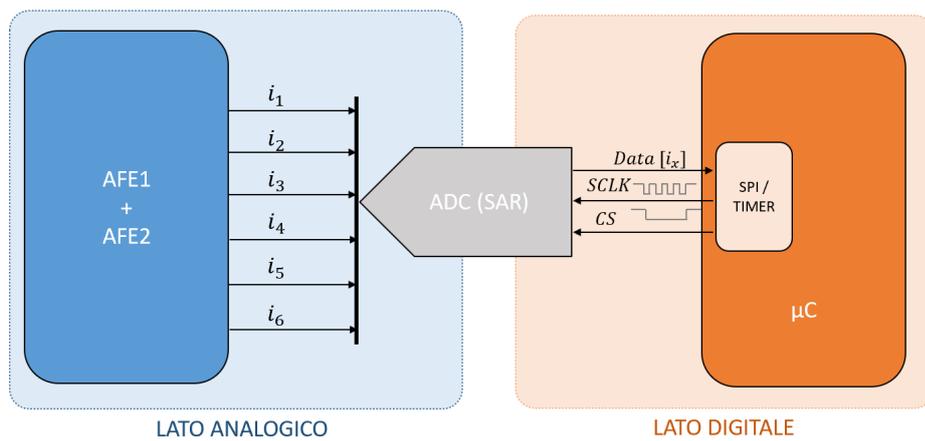


Figura 8.2: Schema a blocchi acquisizioni di corrente

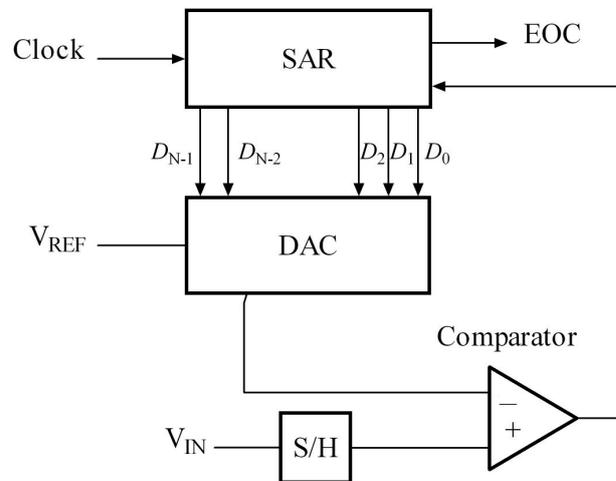


Figura 8.3: Schema a blocchi ADC SAR[18]

L'AD esterno utilizzato in questo progetto di tesi è stato l'AD7276 di Analog Devices il quale, come si può osservare dalla Fig. 8.4 è caratterizzato da un package da 6 pin; si identificano i pin per il segnale in ingresso, per la tensione di alimentazione e per quella di ground, mentre sulla sinistra i pin per il clock, il dato e per il segnale di CS poichè il dato viene trasferito con protocollo SPI.

Tale AD7276 è caratterizzato, inoltre, da una risoluzione a 12 bit ed esegue la conversione basandosi su un meccanismo di ricerca dicotomica attraverso tutti i possibili livelli di quantizzazione fino alla determinazione del valore di conversione finale. Il convertitore analogico-digitale SAR, come visibile in Fig. 8.3, è composto tipicamente da quattro sottocircuiti principali: un circuito sample&hold per l'acquisizione della tensione d'ingresso (V_{in}); un comparatore di tensione che confronta la tensione V_{in} con l'uscita del DAC; un registro ad approssimazioni successive progettato per emettere un codice digitale approssimato della V_{in} al DAC interno ed infine un DAC interno la cui uscita torna al comparatore con un segnale analogico equivalente al codice digitale d'uscita del SAR da comparare con la V_{in} .

Il registro SAR viene inizializzato in modo che il MSB sia uguale ad 1. Questo codice viene mandato al DAC, il quale manda un segnale analogico equivalente al valore digitale $(V_{ref}/2)^1$ all'interno del comparatore per confrontarlo con il campione di tensione all'ingresso. Se questo segnale analogico è maggiore della V_{in} allora al bit verrà assegnato il valore 0, altrimenti il bit verrà lasciato ad 1. Il procedimento viene quindi ripetuto per tutti i bit suc-

¹Nel caso del AD7276 esso è pari a $V_{DD}/2$

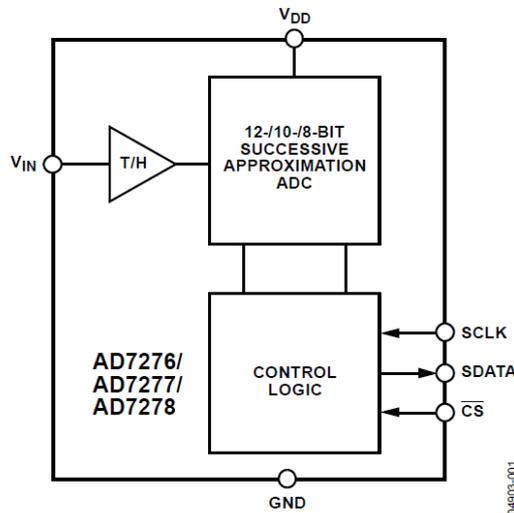


Figura 8.4: Schema a blocchi AD7276

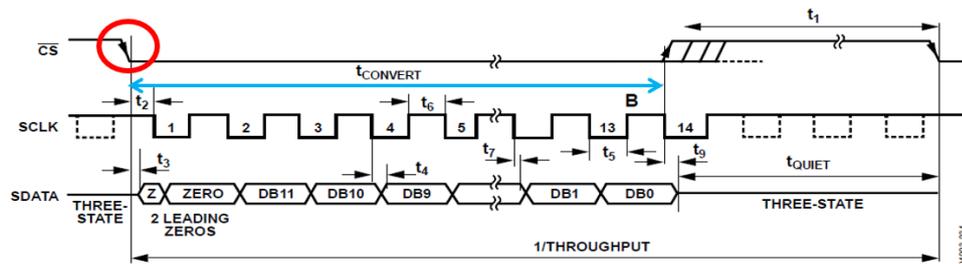


Figura 8.5: Diagramma temporale convertitore AD7276

cessivi usati per la conversione, con una frequenza determinata da un clock esterno. Il codice risultante è un'approssimazione digitale del campione analogico d'ingresso e viene infine emesso dal DAC alla fine della conversione (EOC²).

La Fig. 8.5, presa dal datasheet del convertitore analogico-digitale AD7276, mette in evidenza quali sono i segnali coinvolti nel processo di acquisizione; in particolare si individuano:

- **SDATA**: segnale digitale in output della conversione AD, caratterizzato da una risoluzione di N bit
- **SCLK**: segnale di clock che scandisce la conversione analogico-digitale

²End Of Conversion

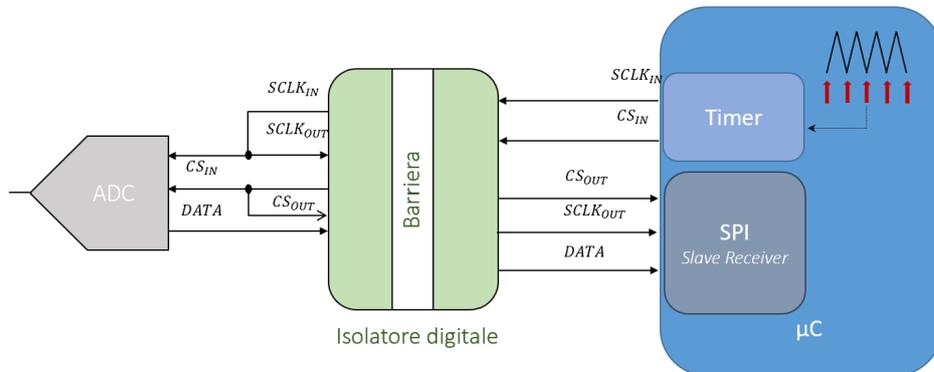


Figura 8.6: Logica di generazione segnali PWM HRTIM

- **CS**: segnale di Chip Select che determina l'inizio del processo di conversione

Si può osservare, in rosso, che l'acquisizione inizia quando c'è il fronte di discesa del segnale CS, dopo un ritardo t_2 ; inoltre, in blu, viene sottolineato che la durata dell'acquisizione dipende dalla risoluzione desiderata del segnale di output.

Per ottenere la corretta informazione relativa al dato campionato dal convertitore AD esterno, è importante che venga garantito l'esatto sincronismo tra il segnale di clock ed il dato; per questo motivo, è stata implementata la soluzione descritta in Fig. 8.6. Questa configurazione è stata scelta per compensare il ritardo introdotto dalla barriera dell'isolatore digitale e per assicurare il corretto sincronismo tra SCLK e DATA, aggiungendo un ritardo nel segnale di clock. Per fare ciò, un timer interno al μC genera i segnali CS e SCLK, facendo in modo che il processo di acquisizione avvenga con un certo ritardo dall'inizio della Interrupt Service Routine (ISR). Inoltre, si è utilizzata la periferica SPI, interna al microcontrollore, configurandola esclusivamente come 'Slave Receiver', la quale riceve in input i segnali CS, SCLK e DATA in uscita dall'AD esterno. La Fig. 8.7 riporta la logica appena descritta.

8.1.1 Generazione CLOCK e CS SPI

Per quanto riguarda l'implementazione della logica presentata nel paragrafo precedente, e quindi la generazione dei segnali di CS e SCLK, si sono utilizzati 3 timer:

- **TIM2** responsabile della generazione del segnale di Chip Select

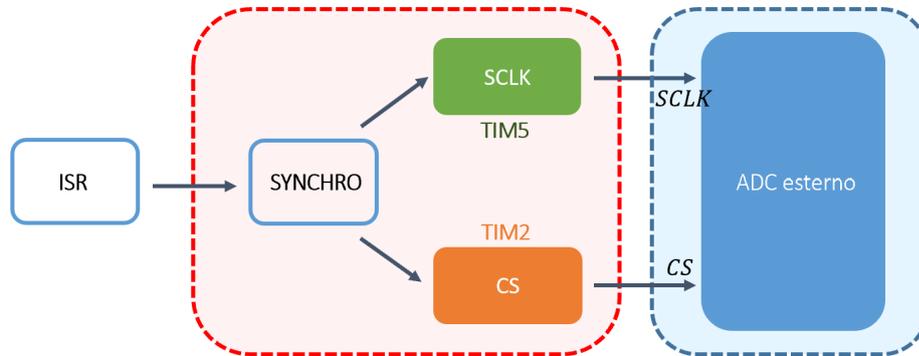


Figura 8.7: Sincronizzazione segnali di CS e SCLK

- **TIM5** responsabile della generazione del segnale di clock
- **TIM3** utilizzato per settare il ritardo tra CS ed SCLK

dove, in Fig. 8.8 $DELAY_{CS}$ rappresenta l'intervallo di tempo che intercorre tra l'inizio della ISR, nel nostro caso corrispondente con uno dei vertici della triangola di TIM8, e lo start di TIM2; $DELAY_{SCLK}$ individua, invece, il ritardo tra il segnale di CS, generato da TIM2, ed il segnale di clock, generato da TIM5. Essendo il TIM3 non configurato per generare alcun output, la Fig. 8.7 riporta esclusivamente i timer responsabili della generazione dei segnali di CS ed SCLK.

TIM2

TIM2 è un up-counter general purpose timer usato per generate il segnale di Chip Select CS, triggerato (hardware) da TIM8. Per quanto riguarda i principali paramentri di configurazione è stato posto $TIM2_{ARR}=5999$ per avere una frequenza doppia rispetto alle ISR; CCR1 è stato settato in funzione del ritardo desiderayo tra TIM8 e il fronte di discesa di CS:

$$CCR1 = DELAY_{CS} * f_{clock} \quad (8.1)$$

con $f_{clock}=240$ MHz, e CCR2 settato in funzione del tempo di acquisizione:

$$CCR2 = T_{acqui,\mu S} * f_{clock} + CCR1 \quad (8.2)$$

dove $T_{acqui,\mu S} = N_{BIT} * T_{SCLK}$, definito in μS , e T_{SCLK} è il periodo del segnale del clock SPI.

In Fig. 8.9 viene riportato un esempio di generazione del segnale di Chip Select, facendo un'operazione logica 'OR' tra il segnale CCR1 ed il

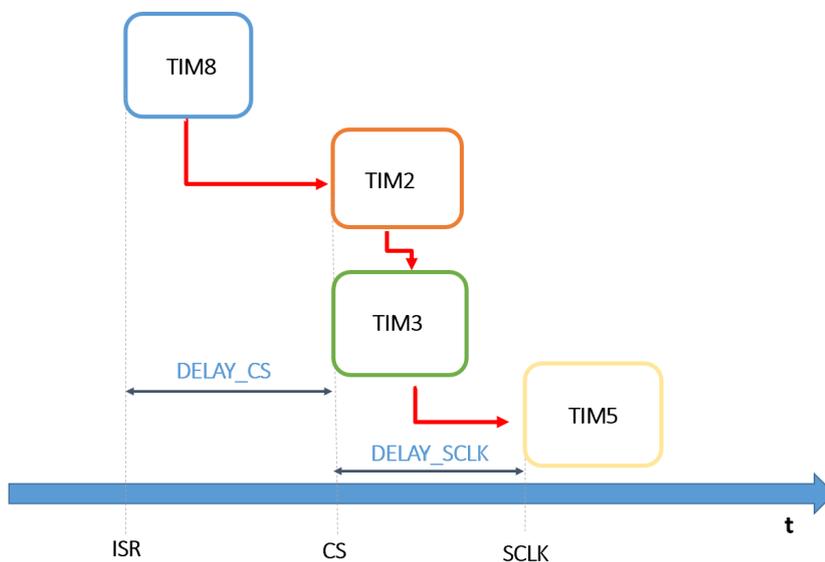


Figura 8.8: Sincronizzazione segnali di CS e SCLK

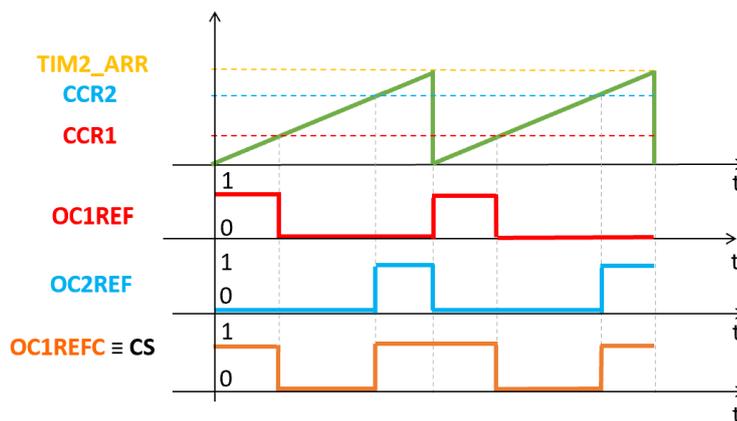


Figura 8.9: TIM2

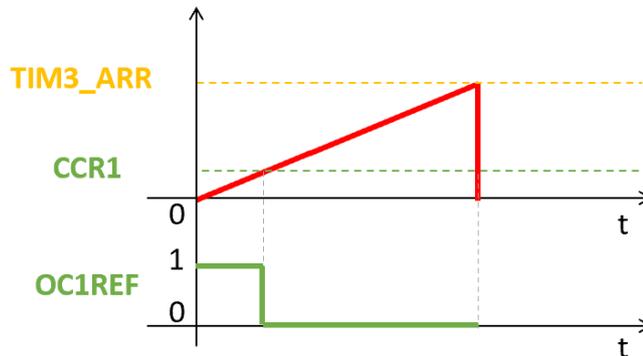


Figura 8.10: TIM3

segnale CCR2. Vengono quindi utilizzati due canali, di cui Channel1 è stato configurato in 'Combined PWM mode 1' ed Channel2 in 'PWM mode 2'; si ricorda che la logica di funzionamento della modalità 'Combine PWM' è stata spiegata più nel dettaglio nel capitolo relativo alla Gestione PWM. Infine, si osserva che il segnale OC1REFC corrisponde proprio al Chip Select desiderato.

TIM3

TIM2 è un up-counter general purpose timer, configurato in 'One Pulse mode' e triggerato dall'update event di TIM2. È stato utilizzato un solo canale, non configurato come output digitale, in quanto il suo scopo principale è quello di impostare il ritardo tra il CS e il segnale del clock SPI SCLK. $TIM3_{ARR}$ è stato posto uguale a $CCR2-1$, dove CCR2 è il valore contenuto del registro di capture compare del canale 2 di TIM2. Inoltre:

$$CCR1 = DELAY_{SCLK} * f_{clock} + compare1_{CS} \quad (8.3)$$

con $compare1_{CS}$ uguale al valore contenuto del registro di capture compare del canale 1 di TIM2.

TIM5

TIM5 è il timer utilizzato per la generazione del segnale di clock; esso è un up-counter general-purpose timer che inizia il proprio conteggio in seguito al trigger hardware dovuto all'evento di compare di TIM3.

$$CCR1 = (TIM5_{ARR} + 1) * 0.5 \quad (8.4)$$

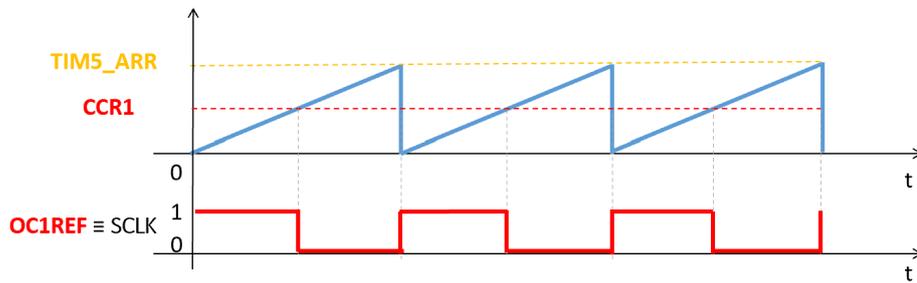


Figura 8.11: TIM5

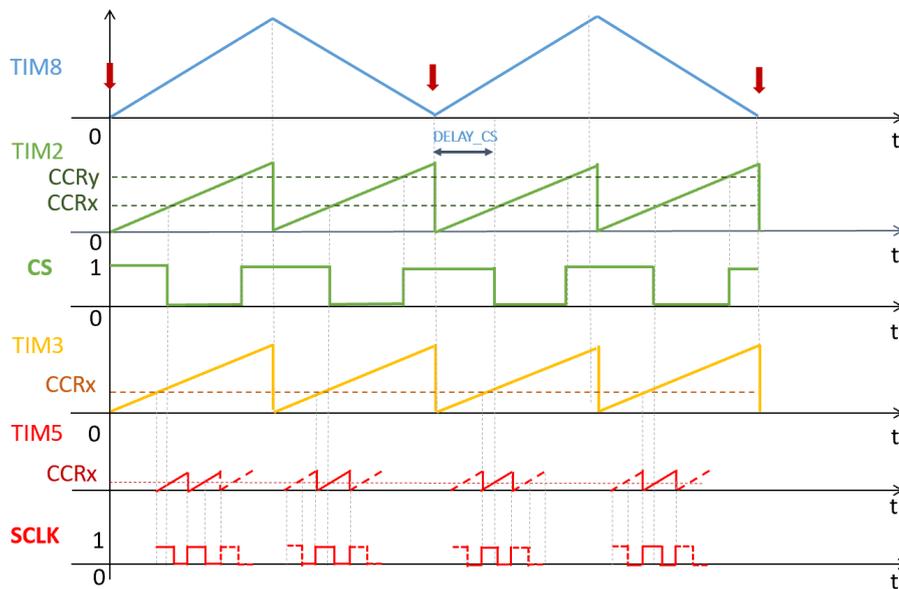


Figura 8.12: Overall generazione segnali di CS e SCLK

con $TIM5_{ARR} = T_{SCLK} * f_{clock}$ e per fare in modo da avere un segnale di SCLK con duty-cycle pari al 50 %

Si conclude la tematica della generazione dei segnali di Chip Select e di clock riportando, in Fig. 8.18, un grafico che riassume il modo in cui sono stati configurati i timer TIM2, TIM3 e TIM5, con relativi segnali PWM, per soddisfare lo scopo preposto.

8.2 Validazione acquisizioni di corrente

Dopo aver effettuato l'analisi teorica della generazione dei segnali di Chip Select e clock, da mandare in ingresso al AD esterno, si è proceduto con la fase di validazione delle forme d'onda, utilizzando la medesima toolchain del caso della validazione dei segnali PWM del capitolo precedente. Successivamente, in laboratorio si è testato il corretto funzionamento dell'intero processo di acquisizione delle correnti confrontando una forma d'onda in ingresso con una forma d'onda generata dal DAC interno al micro controllore, in funzione del segnale acquisito dal AD7276 e trasferito al μC stesso attraverso protocollo SPI.

Algoritmo 4 Implementazione sincronizzazione timer SPI UserSetup.h

```
#define N_BIT    6.0 f
#define DELAY_CS_US    2.0 f
#define SCALA    1.0 f
#define F_SCLK_MHZ (float)(1/SCALA)
#define T_ACQUI_US (float)(N_BIT*SCALA)
#define COMPARE1_CS (uint16_t)(DELAY_CS_US*
STEP_TIME_SYNC_MHZ )
#define COMPARE2_CS (uint16_t)(T_ACQUI_US*
STEP_TIME_SYNC_MHZ+COMPARE1_CS)
#define TIM5_ARR (uint16_t)(SCALA*STEP_TIME_SYNC_MHZ-1)
#define COMPARE_SCLK (uint16_t)(TIM5_ARR/2)
#define DELAY_SCLK_US (float)(SCALA/4)
#define COMPARE_AUX (uint16_t)(COMPARE1_CS-DELAY_SCLK_US*
STEP_TIME_SYNC_MHZ)
```

Come effettuato nel caso del capitolo sulla Gestione PWM, si sono aggiunte delle linee di codice nel file 'UserSetup.h' in cui vengono riportati i principali parametri, modificabili in maniera agevole dall'utente, per regolare le forme d'onda del chip select e del clock in base alle proprie esigenze. Oltre alle grandezze presentate nella sezione precedente, è stato aggiunto il parametro 'SCALA', il quale è stato pensato per aumentare o diminuire la frequenza del segnale di clock: ad un valore di SCALA pari ad 1 corrisponde una $f_{SCLK} = 1$ MHz. Per come è stato implementato il codice riportato nell'algoritmo 4, ad un valore di SCALA maggiore di 1 corrisponde una $f_{SCLK} < 1$ MHz, mentre ad un valore di SCALA minore di 1, ci si pone nel caso opposto. Si può inoltre osservare che il parametro $COMPARE_{AUX}$ coincide con $TIM3_{CCR1}$, mentre $COMPARE1_{CS}$ e $COMPARE2_{CS}$, coin-

cidono rispettivamente con i valori contenuti nei registri di capture compare del canale 1 e del canale 2 di TIM2, responsabile della generazione del chip select.

8.2.1 Validazione generazione CS e SCLK

Per quanto riguarda la validazione della generazione delle forme d'onda di CS e SCLK desiderate, sono stati effettuati tre test riportati nelle sottosezioni che seguono. Lo scopo di questa prima fase di testing è stato quello di controllare che tra i segnali generati venisse garantito il sincronismo voluto; in particolare si è voluto controllare che il segnale di CS venisse generato a partire da un ritardo rispetto all'inizio della ISR, stabilito dall'utente nel file 'UserSetup.h', e che il fronte di discesa del segnale di SCLK fosse ritardato rispetto al fronte di discesa del CS, istante che indica l'inizio del processo di acquisizione. Vengono riportati i tre test effettuati; essi differiscono in base ai parametri di input, specificati rispettivamente nelle tabelle 6.1, 6.2 e 6.3. Si osserva, inoltre, che in questo caso, è stato impostato un ritardo tra CS e SCLK pari a $T_{SCLK}/4$.

Test 1

| | |
|----------------|------------|
| $DELAY_{CS}$ | $3 \mu s$ |
| f_{sw} | 20 kHz |
| T_{sw} | $50 \mu s$ |
| f_{ISR} | 40 kHz |
| N_{BIT} | 14 |
| f_{SCLK} | 1 MHz |
| T_{SCLK} | $1 \mu s$ |
| $DELAY_{SCLK}$ | 250 ns |

Tabella 8.1: Input test 1

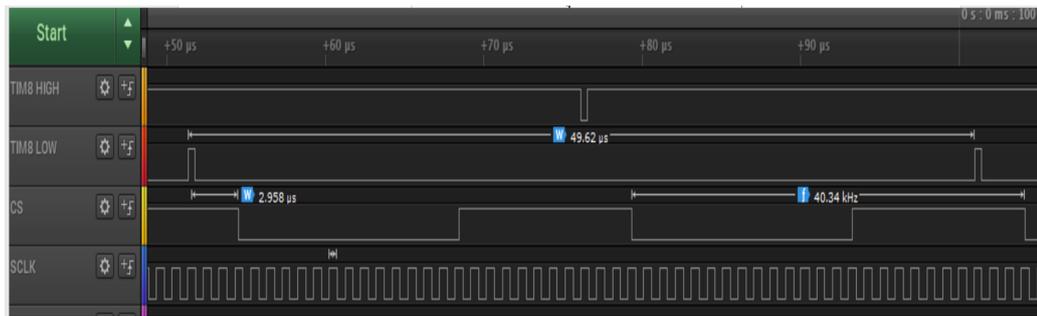


Figura 8.13: Validazione generazione CS e SCLK. Dall'alto verso il basso: Andamenti del vertice alto di TIM8, vertice basso di TIM8, CS e SCLK, test 1

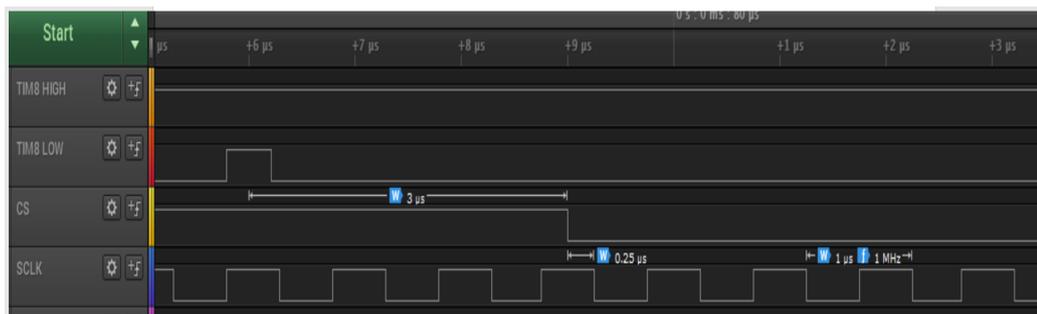


Figura 8.14: Verifica ritardo tra inizio ISR/CS e tra fronte di discesa del CS e fronte di discesa di SCLK, test 1

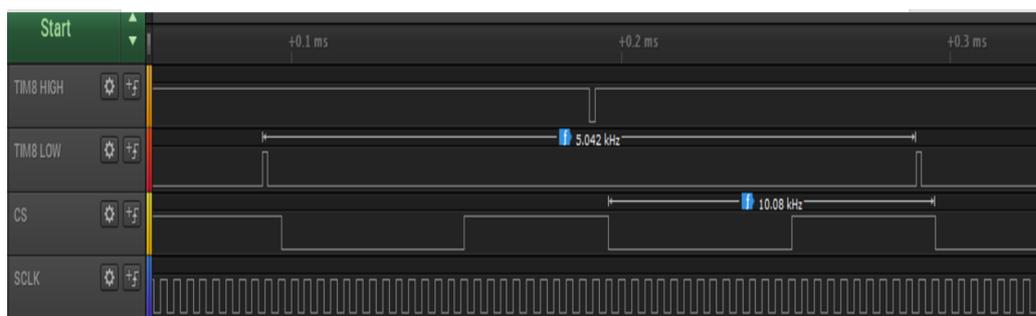


Figura 8.15: Validazione generazione CS e SCLK. Dall'alto verso il basso: Andamenti del vertice alto di TIM8, vertice basso di TIM8, CS e SCLK, test 2



Figura 8.16: Verifica ritardo tra inizio ISR/CS e tra fronte di discesa del CS e fronte di discesa di SCLK, test 2

Test 2

| | |
|----------------|-------------------|
| $DELAY_{CS}$ | $5 \mu s$ |
| f_{sw} | 5 kHz |
| T_{sw} | $200 \mu s$ |
| f_{ISR} | 10 kHz |
| N_{BIT} | 14 |
| f_{SCLK} | 250 kHz |
| T_{SCLK} | $4 \mu s$ |
| $DELAY_{SCLK}$ | $1 \mu s$ |

Tabella 8.2: Input test 2

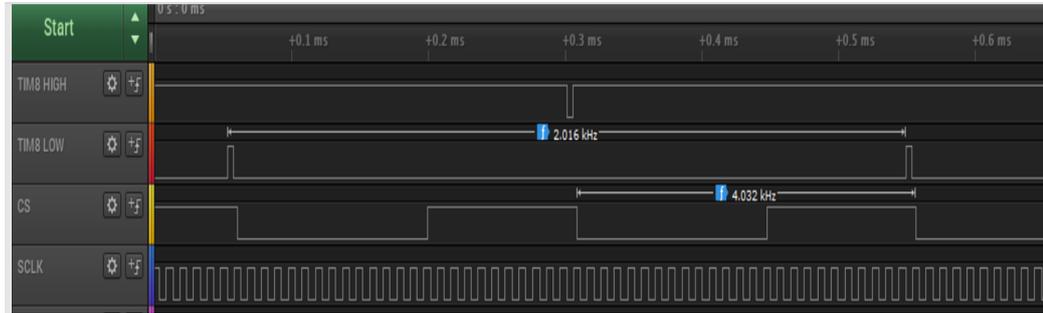


Figura 8.17: Validazione generazione CS e SCLK. Dall'alto verso il basso: Andamenti del vertice alto di TIM8, vertice basso di TIM8, CS e SCLK, test 3

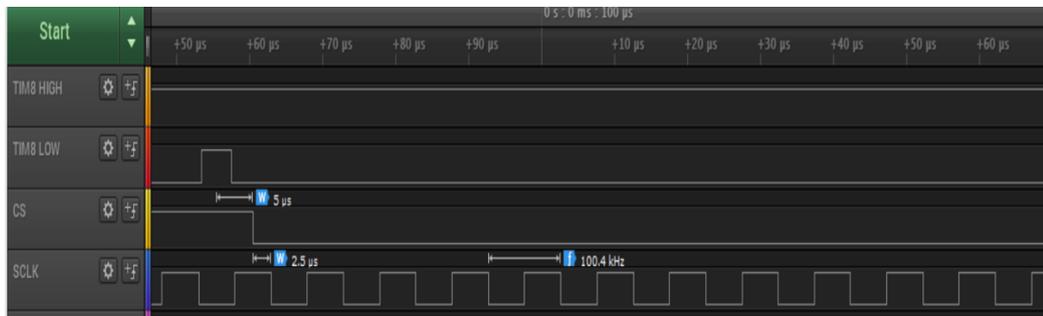


Figura 8.18: Verifica ritardo tra inizio ISR/CS e tra fronte di discesa del CS e fronte di discesa di SCLK, test 3

Test 3

| | |
|----------------|-------------------|
| $DELAY_{CS}$ | $5 \mu s$ |
| f_{sw} | 2 kHz |
| T_{sw} | $500 \mu s$ |
| f_{ISR} | 4 kHz |
| N_{BIT} | 14 |
| f_{SCLK} | 100 kHz |
| T_{SCLK} | $10 \mu s$ |
| $DELAY_{SCLK}$ | $2.5 \mu s$ |

Tabella 8.3: Input test 3

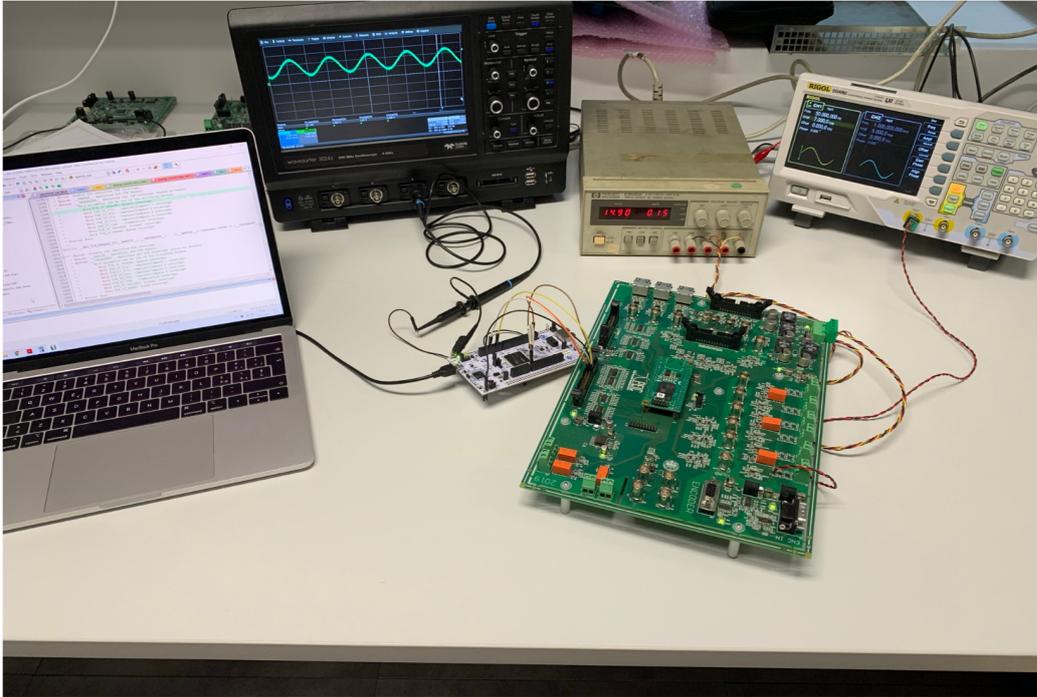


Figura 8.19: Setup sperimentale validazione acquisizioni di corrente

8.2.2 Test SPI in laboratorio

Il setup sperimentale è riportato in 8.19 ed è costituito, procedendo in senso orario, da: notebook, oscilloscopio, alimentatore bipolare, generatore di funzioni, scheda contenente l' AD7276 ed infine dalla NUCLEO-H745ZIQ.

Avendo a disposizione strumentazione più sofisticata rispetto all'analizzatore logico usato nella prima parte, si è adattato il codice per simulare una ISR alla frequenza di 72 kHz. Si può osservare in Fig. 8.20 che la ISR, nel caso in esame coincidente con uno dei vertici della triangola di TIM8, ha una frequenza di 72 kHz (in giallo), mentre il chip select (in rosso) ha frequenza doppia ossia 144 kHz, ossia che l'acquisizione di corrente viene fatta due volte ogni periodo di switching T_{sw} .

In Fig. 8.21 è stato effettuato uno zoom dell'immagine precedente perchè si è voluto mettere in evidenza l'intervallo di tempo che intercorre tra l'inizio della ISR e il fronte di discesa del segnale CS. Inoltre, si osserva che in questo caso il dato prelevato ha una risoluzione di 12 bit; questo si può influire dal fatto che ogni bit della parola digitale SDATA viene trasferito in corrispondenza di ogni fronte di salita del segnale SCLK (in blu) come specificato nel datasheet dell'AD7276.

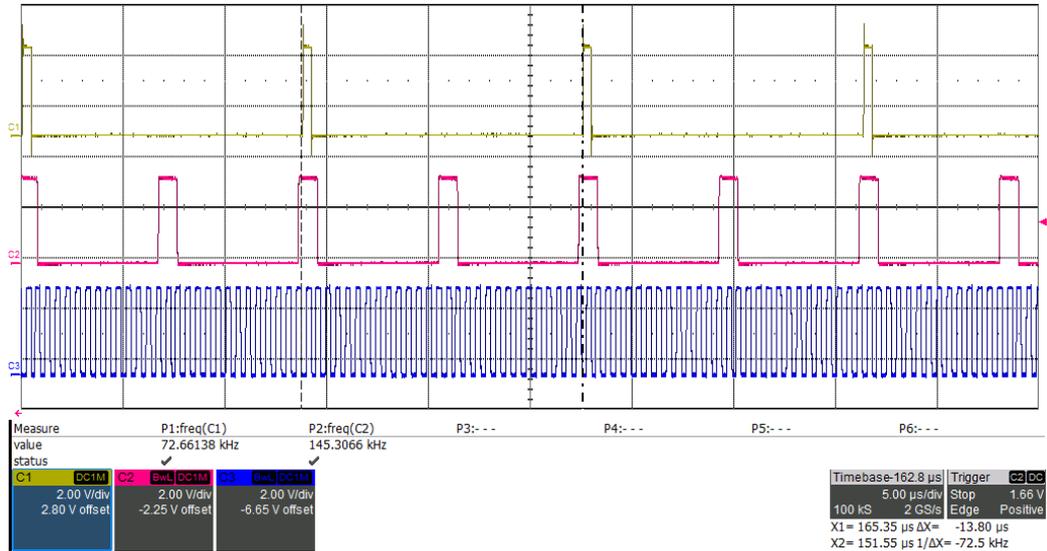


Figura 8.20: Verifica ISR a 72 kHz e acquisizione di corrente a 144 kHz

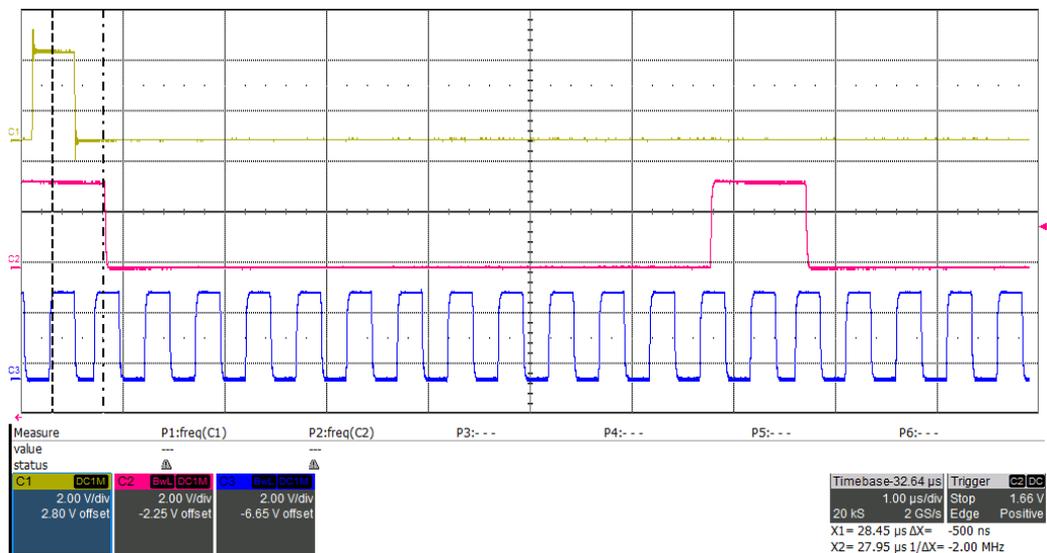


Figura 8.21: Andamenti di: vertice basso di TIM8 (linea gialla), CS (linea rossa) e SCLK (linea blu), con ISR a 72 kHz

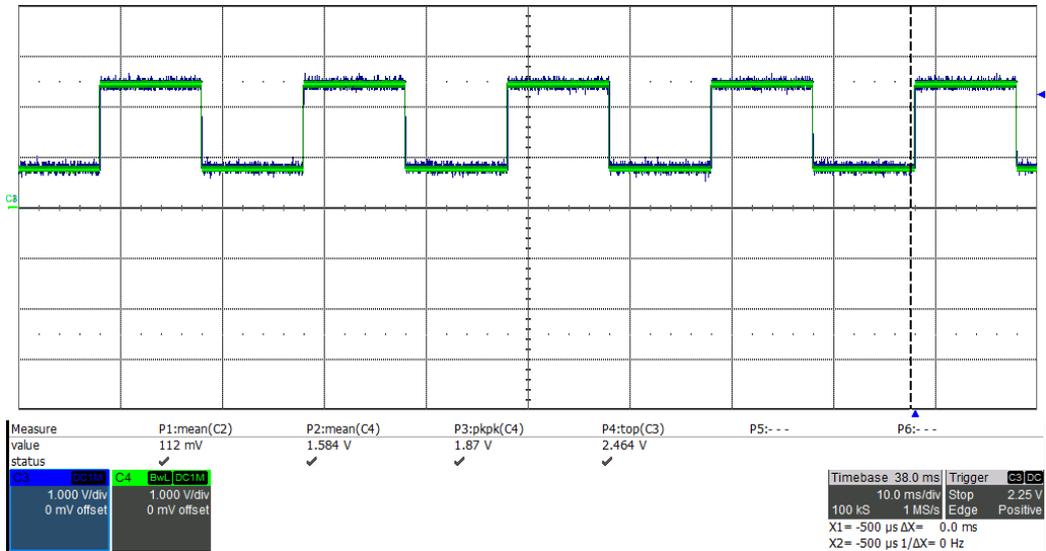


Figura 8.22: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 50 Hz

Successivamente, si è utilizzato il generatore di funzioni per generare un segnale analogico che viene convertito in una parola digitale dall'AD7276, attraverso la barriera digitale (presente sulla scheda verde della Fig. 8.19) e tramite delle connessioni viene mandato sul pin MOSI (Master Output Slave Input) presente sulla scheda NUCLEO-H745ZIQ. Internamente al μC tale segnale digitale viene convertito in analogico dalla periferica DAC e tramite oscilloscopio si può quindi visualizzare la forma d'onda analogica ricostruita dal DAC sul pin corrispondente al canale 1.

Si riportano in Fig. 8.22, Fig. 8.23 e Fig. 8.24, i casi corrispondenti ad un segnale analogico in ingresso, rispettivamente, pari a 50 Hz, 150 Hz ed 1 kHz. La curva verde rappresenta il segnale analogico in ingresso proveniente dal generatore di funzioni, mentre la curva blu (in secondo piano) identifica l'uscita del DAC. Si è provveduto ad acquisire forme d'onda alla frequenza fondamentale delle correnti di fase (50 Hz), alla frequenza delle componenti di terza armonica ed alla frequenza di 2 kHz.

Nelle Fig. 8.25 e 8.26 si può osservare, con due forme d'onda differenti alla frequenza di 50 Hz, la corretta configurazione delle periferiche del micro relative alla gestione delle acquisizioni di corrente; questo è possibile dedurlo dal fatto che le forme d'onda ottenute in questa fase di validazione sono uguali; infatti, se l'output digitale dell'AD7276 non fosse stato correttamente trasferito al μC e non fosse stato implementato correttamente il codice, l'uscita del DAC sarebbe stata differente dal segnale analogico in ingresso all'AD

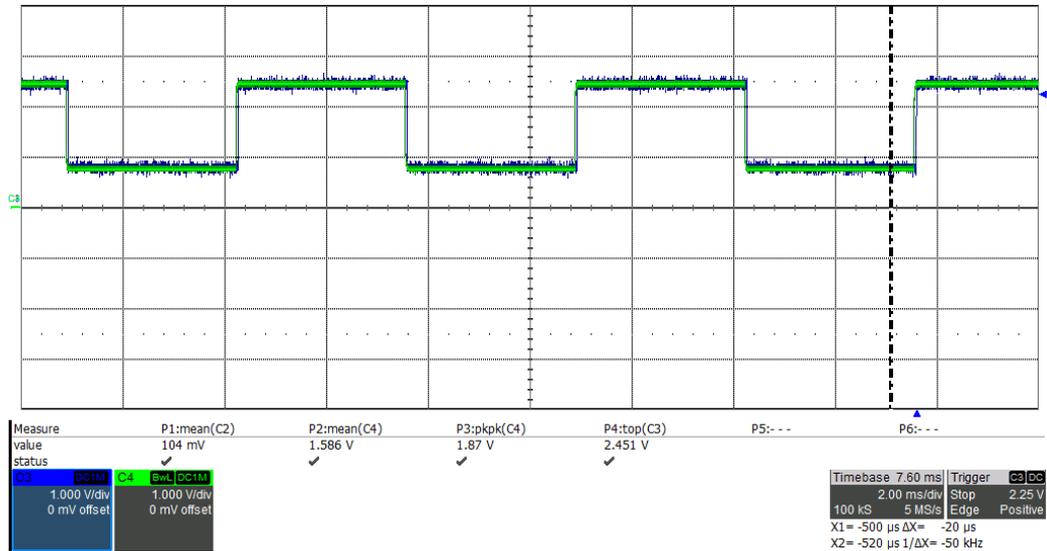


Figura 8.23: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 150 Hz

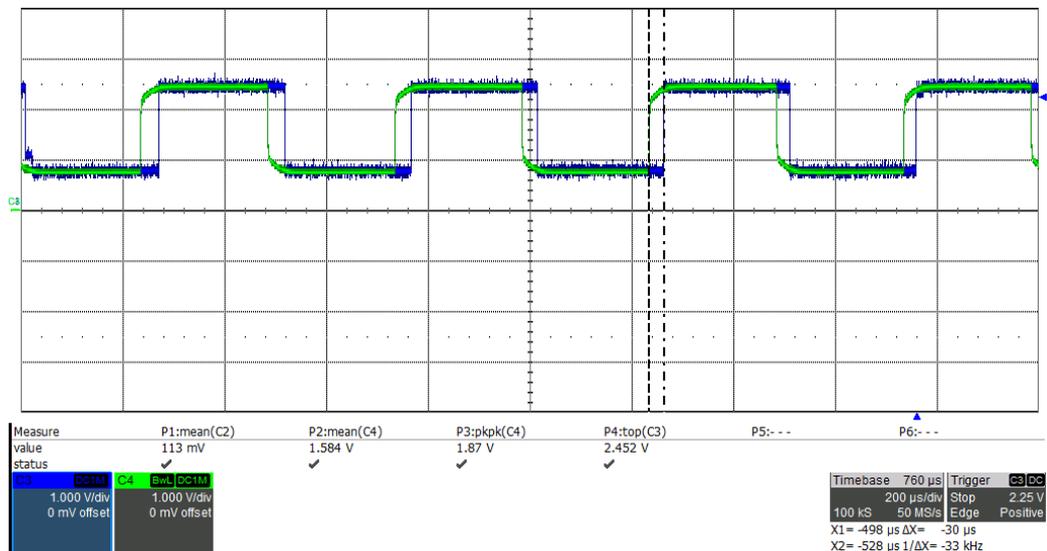


Figura 8.24: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 2 kHz

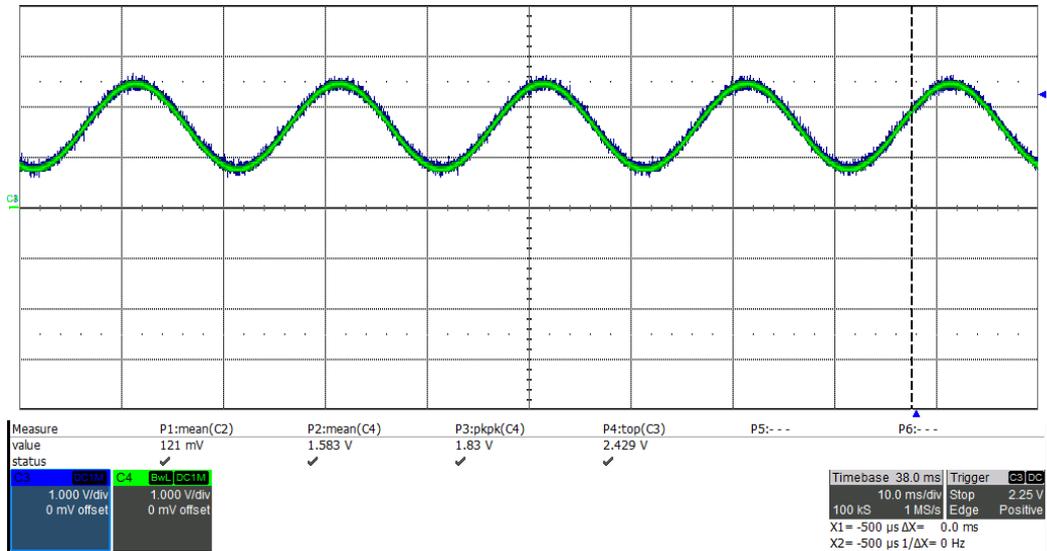


Figura 8.25: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), sinusoidi a 50 Hz

esterno. Per completezza, si può aggiungere che si è utilizzata la periferica DMA (Direct Memory Access) per alleggerire il lavoro alla CPU spostando il dato dalla periferica SPI alla memoria in 'Circular mode'.

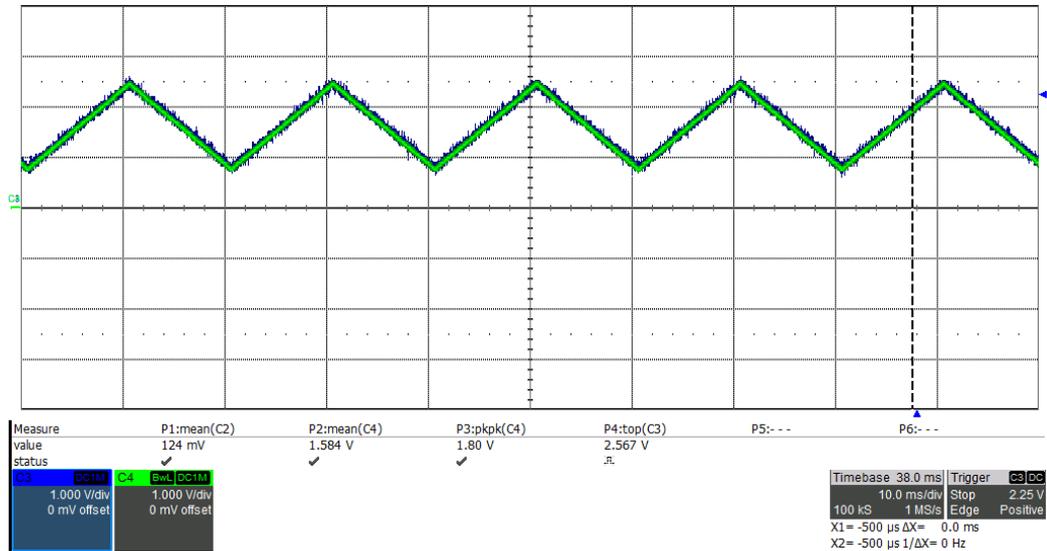


Figura 8.26: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda triangolare a 50 Hz

8.3 Acquisizioni di tensione

Oltre all'acquisizione delle correnti di fase, ci si è soffermati sulla gestione delle acquisizioni di tensione, la cui implementazione è riportata in Fig. 8.27; in particolare, le tensioni da acquisire per garantire il corretto controllo del sistema sono: le tre tensioni di rete V_{ab} , V_{bc} , V_{ca} e le due tensioni del DC-link V_+ e V_- . Inoltre, la stessa figura mette in evidenza che ad ogni vertice basso (o indifferentemente ogni vertice alto) e quindi ad ogni ISR, viene triggerata l'acquisizione delle tensioni. Il convertitore analogico-digitale utilizzato per trasformare il segnale analogico in una parola digitale è l'ADS1205 di Texas Instruments. Esso è un modulatore Sigma-Delta a 2 canali, alimentato a +5V, con package QFN-24, come mostrato in Fig. 8.28. Il clock viene fornito dal μC , mentre esso riceve il clock di ritorno dal modulatore esterno ed il dato convertito sotto forma di parola digitale.

Il segnale digitale disponibile sui pin di uscita OUTx (con x=A,B) è un 1-bit stream ad elevato output rate che bisogna processare, con una serie di operazioni matematiche, per raggiungere la risoluzione e la frequenza del dato in uscita desiderato. A questo scopo si è utilizzata la periferica, interna al μC , Digital Filter Sigma Delta Modulators.

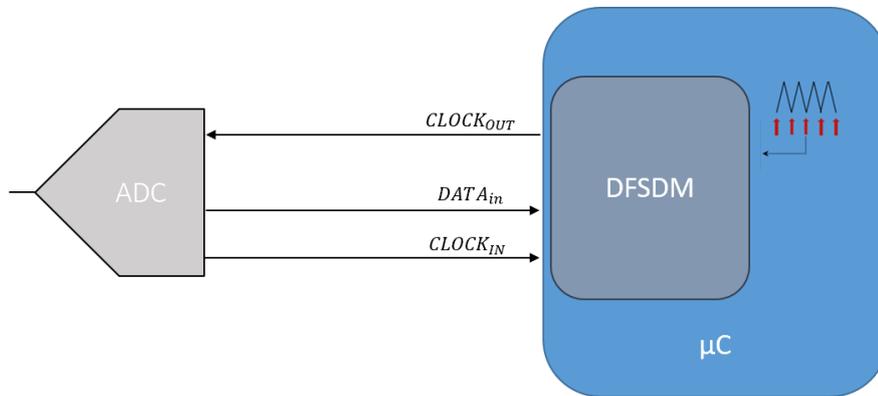


Figura 8.27: Overview gestione acquisizioni di tensione

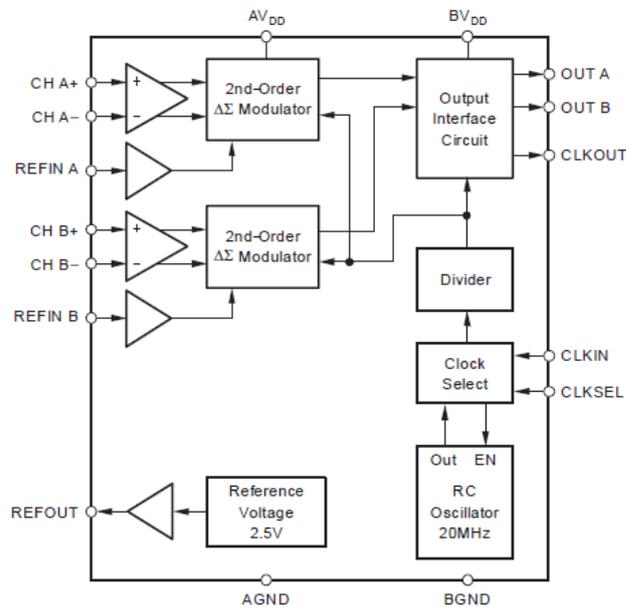


Figura 8.28: Schema a blocchi ADS1205

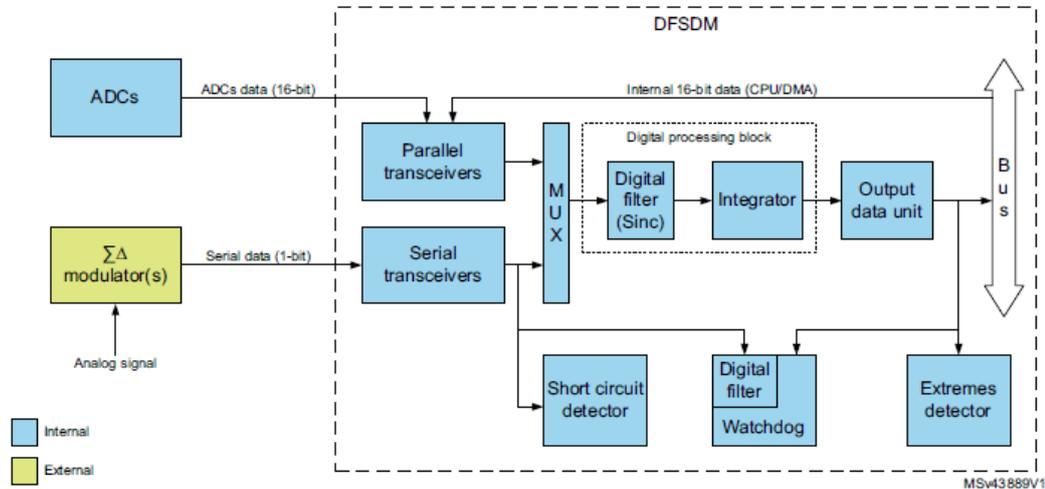


Figura 8.29: Schema a blocchi periferica DFSDM[19]

8.3.1 Configurazione periferica DFSDM

Lo schema a blocchi della periferica DFSDM, con i blocchi funzionali interni e le loro connessioni interne ed esterne, è riportato in Fig. 8.29. Come si può notare, tale periferica non può essere ridotta ad un solo filtro digitale, poichè essa raggruppa un insieme di blocchi che gestiscono l'intero processo di conversione analogico-digitale, quando connessa ad un modulatore sigma-delta esterno.

Tra i componenti principali si distinguono:

- I trasmettitori, seriali o paralleli, che ricevono il dato dall'esterno
- Il blocco digital filter che rappresenta il componente chiave che processa il dato in input
- Un integratore che si comporta come un semplice sommatore
- L'output data unit, la quale effettua la correzione finale sul dato in uscita per mezzo di right bits shifting ed applicando un offset al dato proveniente dall'integratore

La velocità di conversione e la risoluzione sono regolabili secondo parametri configurabili per l'elaborazione digitale: tipo di filtro, ordine del filtro, lunghezza del filtro, lunghezza dell'integratore. La risoluzione massima dei dati in uscita è di 24 bit. Sono disponibili due modalità di conversione: conversione singola modalità e modalità continua. Entrando più nello specifico,

DFSDM contiene un'implementazione del filtro digitale di tipo Sincx. Questo filtro Sincx esegue il filtraggio del flusso di dati digitali, che si traduce in una diminuzione della velocità dei dati in uscita e aumentando la risoluzione dei dati di output. I principali parametri configurabili sono:

- FOSR: Filter Oversampling Ratio
- FORD: Filter Oversampling Order
- IOSR: Integrator Oversampling Ratio

Come già specificato, una delle conseguenze dell'operazione di filtro Sinc (media mobile) è aumentare la risoluzione del segnale campionato (di un fattore FOSR). La risoluzione totale del segnale di uscita è quindi:

$$Risoluzione_{OUT} = Risoluzione_{IN}^{(FOSR * FORD)} \quad (8.5)$$

dove $Risoluzione_{IN}$ corrisponde alla risoluzione dei dati in ingresso (2 in caso di ingresso dati seriale, come nel nostro caso). Bisogna prestare attenzione a non aumentare $Risoluzione_{OUT}$ oltre l'intervallo di 32 bit perché il dato viene memorizzato in un registro a 32 bit. Oltre all'aumento della risoluzione, l'altra conseguenza dell'operazione di media mobile è la riduzione della velocità dei dati di output (output data rate). Essa può essere calcolata, come specificato nel reference manual, nel seguente modo:

$$Datarate_{OUT} = \frac{f_{sampling}}{FOSR} \quad (8.6)$$

con $f_{sampling}$ uguale alla frequenza di campionamento del dato in ingresso. Poiché $Datarate_{OUT}$ dipende solo da FOSR mentre $Risoluzione_{OUT}$ dipende sia da FOSR che da FORD, si consiglia di regolare prima il FOSR per ottenere la velocità dati di uscita richiesta, e successivamente regolare il FORD per ottenere la risoluzione di uscita desiderata.

Si riporta in Fig. 8.30, il foglio di calcolo excel messo a disposizione da ST per la configurazione di un singolo filtro della periferica DFSDM. Poiché il μC utilizzato STM32H745ZI integra solo quattro unità di filtro digitale e le tensioni da acquisire sono cinque, si è dovuto fare in modo che un filtro venisse usato per gestire due diverse tensioni. Per questo motivo, come riportato in Fig. 8.30, volendo avere a disposizione i valori delle tensioni ogni 72 kHz, si è posta una frequenza di trigger pari a 144 kHz e che i 2 canali relativi alle tensioni da acquisire dallo stesso filtro, facessero parte dello stesso gruppo injected. Si può notare, inoltre, che con un FOSR=16 e un FORD=4, si riescono a soddisfare i valori desiderati in termine di risoluzione (17 bit) e di output data rate (acquisizione di tensione ogni 72 kHz).

| Input parameters (conversion) | unit | option |
|---------------------------------------|-------------|---------------|
| Serial/parallel mode | | serial |
| Sampling frequency | [Hz] | 10 000 000 |
| Parallel data resolution | [+/-bits] | 12 |
| Continuous mode | | no |
| Regular/injected mode | | injected |
| Number of scanned channels | | 2 |
| Trigger | | Timer |
| Trigger frequency | [Hz] | 144 000 |
| Filter order (FORD) | | 4 |
| Filter oversampling ratio (FOSR) | | 16 |
| Integrator oversampling ratio (IOSR) | | 1 |
| Right bit shift | [bits] | 0 |
| Offset correction | | 0 |
| | | |
| | | |
| | | |
| | | |
| Output parameters (conversion) | unit | option |
| Max output data rate | [samples/s] | 144 927,54 |
| Real output data rate (trigger) | [samples/s] | 144 000,00 |
| Real output data rate per channel | [samples/s] | 72 000,00 |
| Internal resolution | [+/- range] | 65 536 |
| Internal resolution | [bits] | 17,00 |
| Internal data overflow alert! | | OK |
| Trigger overflow alert! | | OK |
| Output resolution | [+/- range] | 65 536 |
| Output bits resolution | [bits] | 17,00 |
| Output bits effective resolution | [bits] | 17,00 |
| Output data overflow alert! | | OK |

Figura 8.30: Schema a blocchi periferica DFSDM

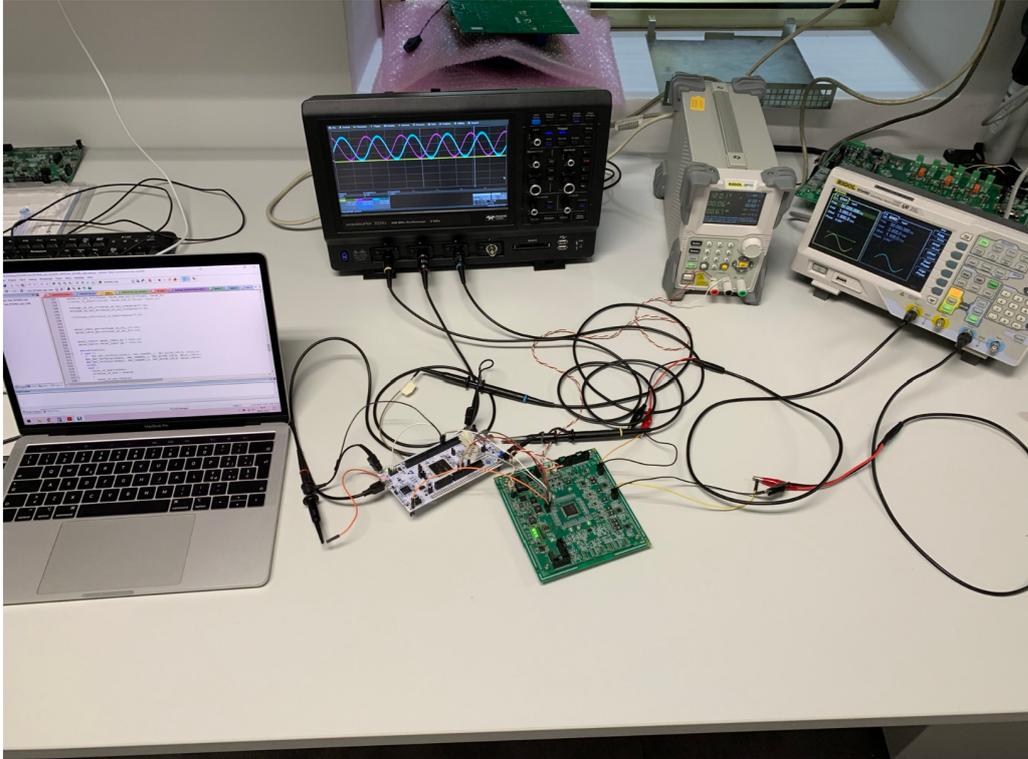


Figura 8.31: Setup sperimentale validazione acquisizioni di corrente

8.4 Validazione acquisizioni di tensione

Per quanto riguarda la validazione della gestione acquisizioni di tensione, i test sono stati effettuati esclusivamente in laboratorio. Si riporta in Fig. 8.31 il setup sperimentale relativo ai test per validare la corretta configurazione della periferica DFSDM. Si identificano, in senso orario: notebook, oscilloscopio, alimentatore unipolare, generatore di funzioni, scheda che integra il modulatore sigma-delta ADS1205 ed infine la NUCLEO-H745ZI. Le prove sono state effettuate, come per il caso delle acquisizioni di corrente, con una f_{sw} a 72 kHz. La Fig. 8.32 mostra come il segnale analogico in uscita al DAC (curva blu) venga aggiornato ad ogni ISR (curva gialla), quindi alla frequenza di 72 kHz.

Successivamente sono state eseguite le stesse prove, ed acquisito forme d'onda simili, al caso presentato nella sezione 'Validazione acquisizioni di corrente'. In particolare, le Fig. 8.33, 8.34 e 8.35 riportano gli andamenti della tensione analogica in ingresso al modulatore sigma-delta (linea gialla) e dell'uscita della periferica DAC del μC (linea blu) nel caso di ingresso, rispettivamente, a 50 Hz, 150 Hz e 2 kHz. Si può osservare che, nel caso

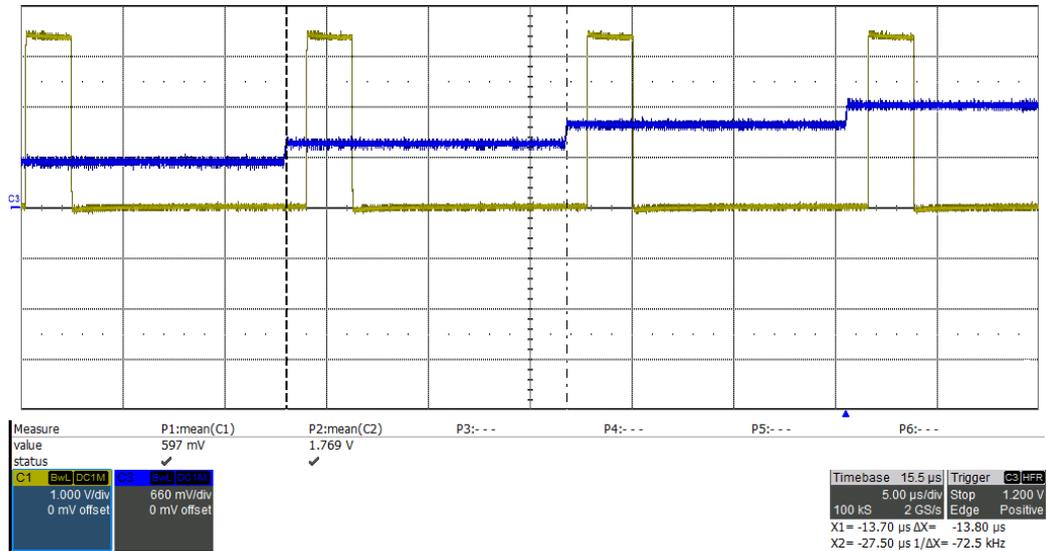


Figura 8.32: Aggiornamento segnale DAC (linea blu) alla frequenza della ISR (linea gialla) 72 kHz

di ingresso in tensione alla frequenza di 2 kHz, il segnale disponibile sul pin associato al DAC presenta un'attenuazione di ampiezza ed un ritardo di fase più evidente rispetto agli altri due casi.

La Fig. 8.36 mostra il corretto funzionamento del processo di acquisizione, nel caso di due tensioni in ingresso all'ADS1205 sigma-delta, nel caso di forma d'onda sinusoidale a 50 Hz. Gli ingressi (linea gialla e linea verde) sono pressochè uguali, fatta eccezione per il rumore introdotto dall'oscilloscopio, delle forme d'onda relative ai 2 canali del DAC, interno al micro controllore (linea rosse e linea blu). Si riporta, infine, come nel caso della validazione delle acquisizioni di corrente, l'andamento della tensione generata dal generatore di funzioni (linea gialla) e l'uscita del DAC (linea blu) nel caso di forma d'onda triangolare con $f = 50$ Hz

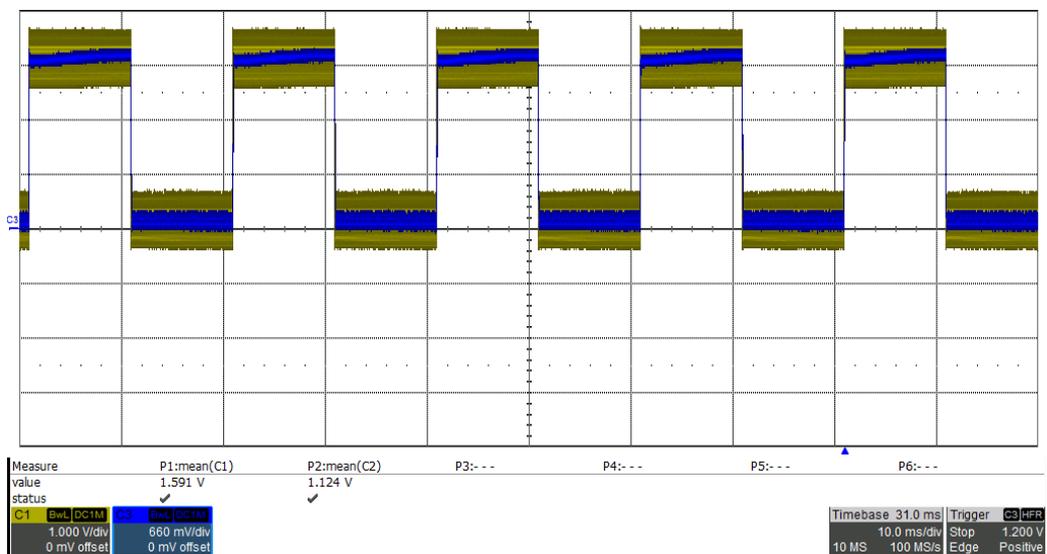


Figura 8.33: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 50 Hz, ADC Sigma-Delta

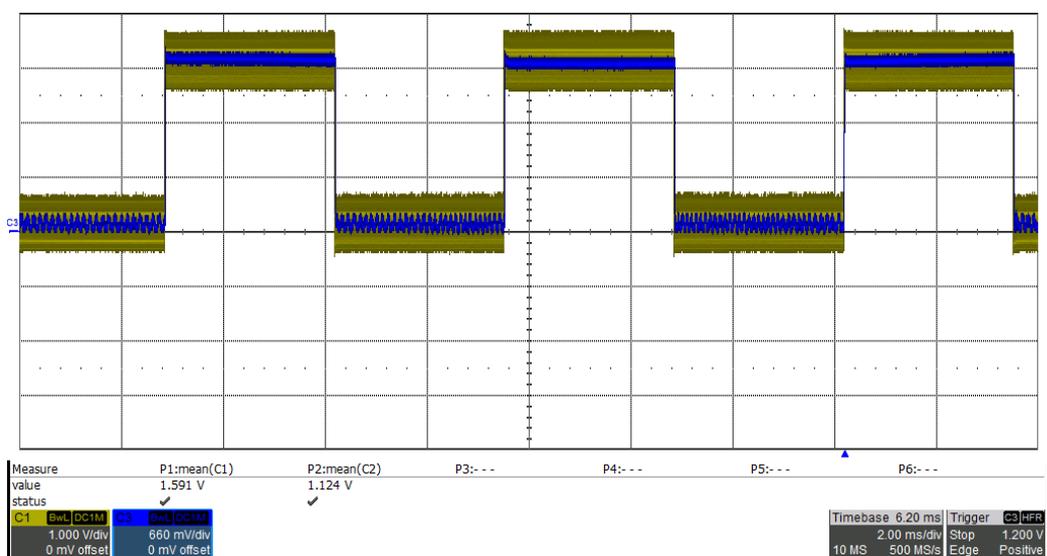


Figura 8.34: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 150 Hz, ADC Sigma-Delta

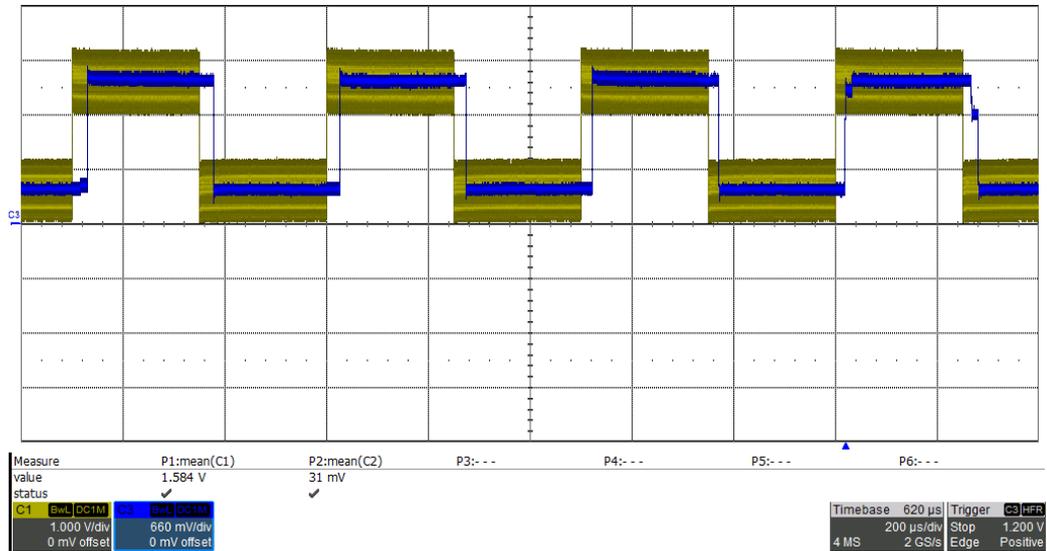


Figura 8.35: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda quadra a 2 kHz, ADC Sigma-Delta

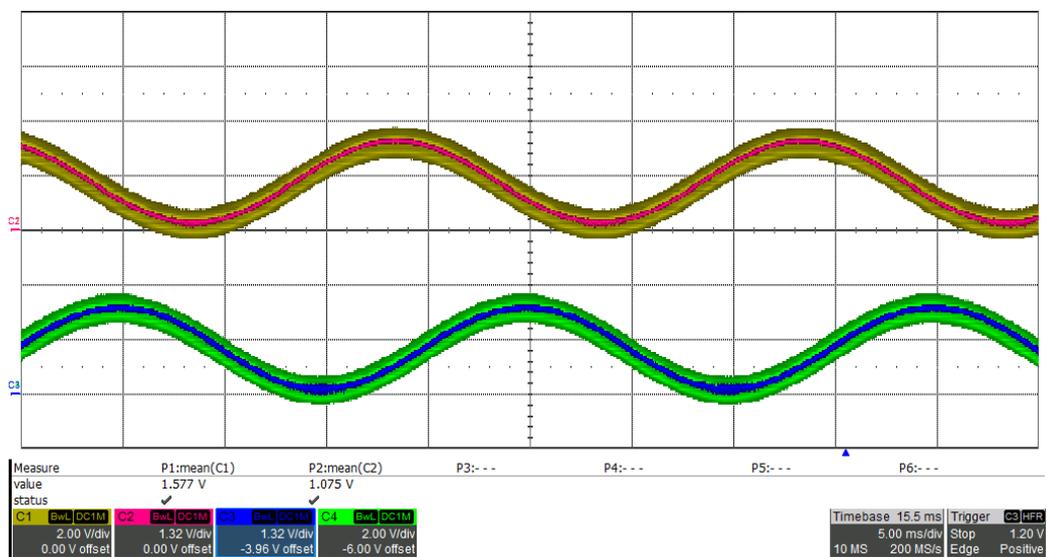


Figura 8.36: Andamenti di: segnali analogici in input (linea gialla e linea verde) e uscita DAC (linea rossa e linea blu), onda triangolare a 50 Hz, ADC Sigma-Delta

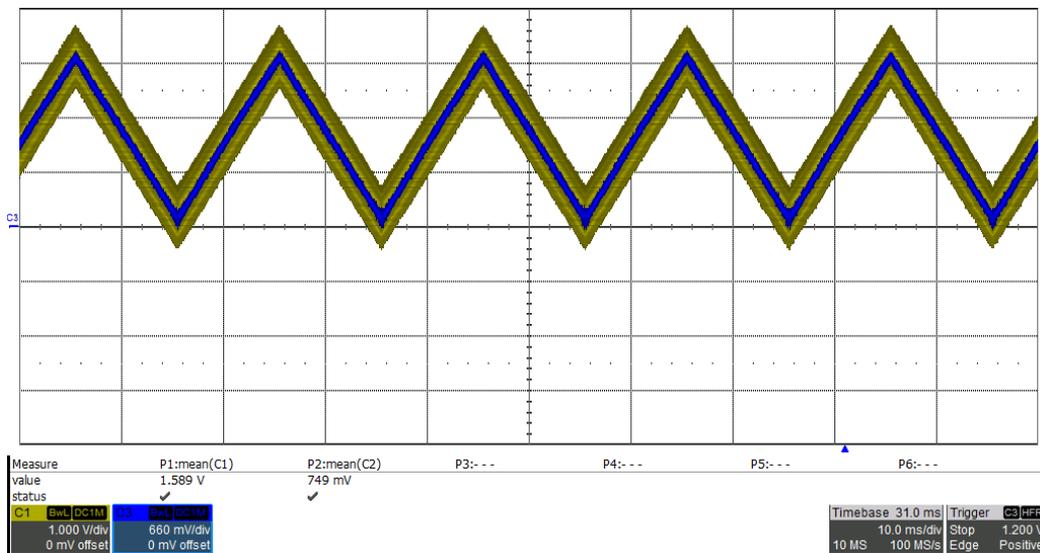


Figura 8.37: Andamenti di: segnale analogico in input (linea verde) e uscita DAC (linea blu), onda triangolare a 50 Hz, ADC Sigma-Delta

Capitolo 9

Assemblaggio e test AFE

Nell'ultima fase del presente progetto di tesi ci si è dedicati all'assemblaggio ed al successivo testing delle schede che costituiscono il convertitore Active Front-End T-type. In particolare, a causa di ritardi nelle spedizioni di alcuni componenti, la scheda di misura non è stata assemblata completamente. Per questo motivo, le acquisizioni non sono state testate sul convertitore finale a differenza dei segnali PWM, oggetto del presente capitolo. La fase di assemblaggio e di validazione sperimentale del convertitore è stata svolta con altri membri del team PEIC, principalmente con il tesista che si occupato del design hardware del convertitore. In Fig. 9.1 viene riportata la scheda MCU contenente il microcontrollore scelto, ossia il μC STM32H745XI6.

Tale scheda è stata poi montata sulla scheda Carrier (9.2) che rappresenta la parte dell'Active Front End che contiene il sistema di alimentazione DC a bassa tensione per tutti i dispositivi elettronici, i circuiti di comunicazione, il circuito di pilotaggio dei relè, i connettori di debug ed i connettori che comunicano con la scheda MCU [20]. Nello specifico:

- comunicazioni UART, USB, CAN
- segnali in entrata e in uscita scambiati con le altre schede (Misure scheda, driver, alimentazione)
- relè di precarica
- segnali di debug e comunicazioni interne (I2C, DEBUG, JTAG)



Figura 9.1: Scheda MCU con μ C STM32h745XIH6

Uno dei motivi per cui si è deciso di escludere dalla scheda Carrier l'MCU e i suoi filtri è perchè questa soluzione fornisce un elevato grado di libertà nell'implementazione della tecnologia di controllo, con la possibilità di utilizzare MCU e anche FPGA, considerando futuri sviluppi.

La Fig. 9.3 mostra le tre schede contenenti i driver dei MOSFET di potenza; si distinguono su ciascuna scheda 8 driver, di cui 4 corrispondono ad una fase di un'unità e gli altri 4 corrispondono all'unità in interleaving.

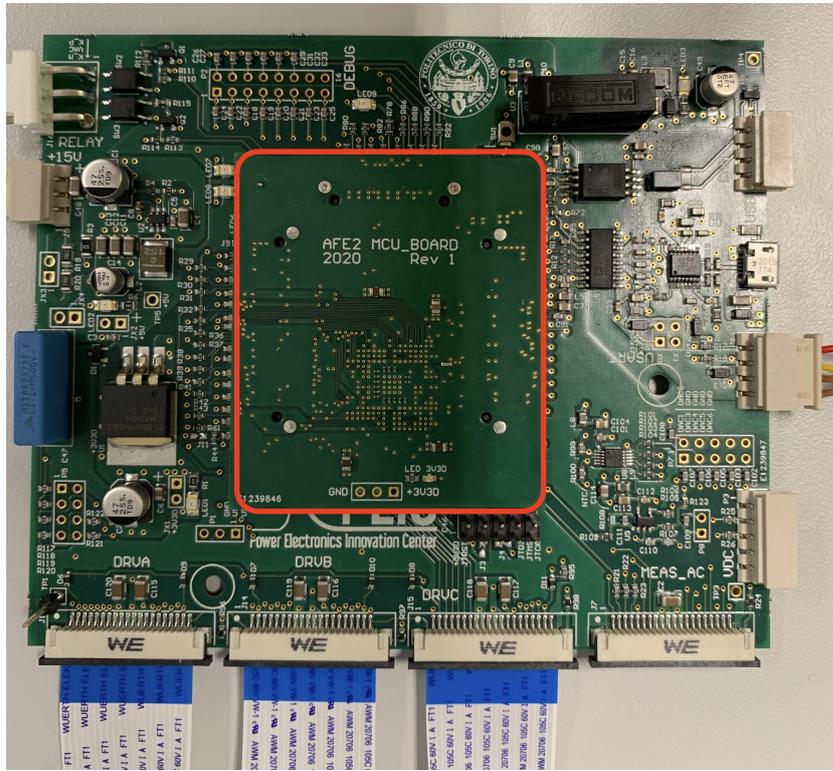


Figura 9.2: Scheda Carrier e scheda MCU (in rosso)

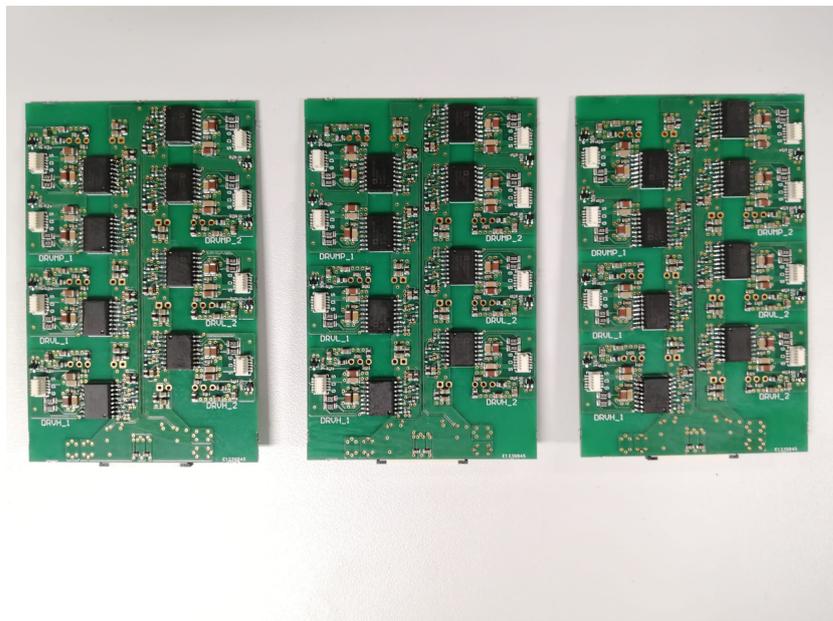


Figura 9.3: Schede driver

9.1 Test segnali PWM su schede finali

Si riporta in Fig. 9.4, il setup sperimentale utilizzato per testare la corretta configurazione delle periferiche legate alla generazione dei segnali di gamba degli switch di potenza. Si distinguono: le schede presentate nella sezione precedente, ossia le schede driver e la scheda Carrier con la scheda MCU sovrapposta ad essa; il PC, il generatore di funzioni, un oscilloscopio ad otto canali ed un pulsante utilizzato per lo start e stop dei segnali PWM.

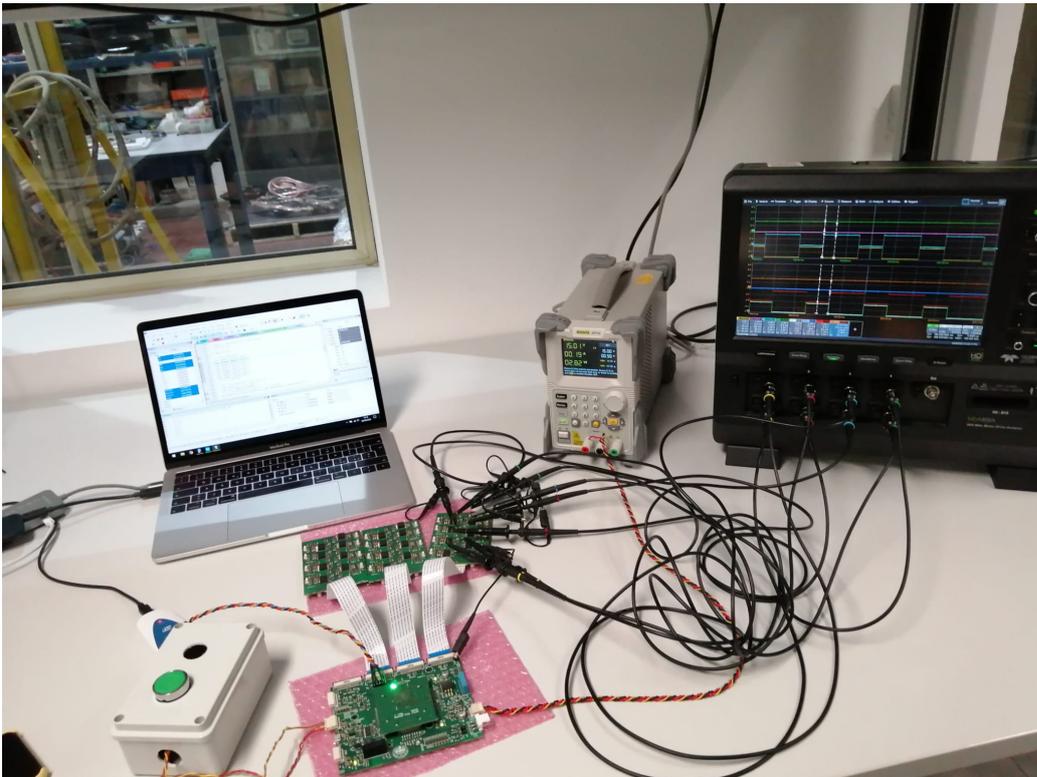


Figura 9.4: Setup sperimentale test segnali PWM

Si è utilizzato un oscilloscopio ad otto canali per poter visualizzare contemporaneamente le otto forme d'onda corrispondenti agli switch della fase omologa per il convertitore 1 ed il convertitore 2. Le Fig. 9.5 e 9.6 mostrano i segnali destinati ai gate driver nel caso, rispettivamente, si voglia realizzare sul carico una tensione nulla ($d = 0.5$) o positiva ($d > 0.5$). Si osserva, nella Fig. 9.5 che il segnale in verde scuro (2 V/Div) ed il segnale in verde chiaro (1 V/Div) hanno solo la scala di tensione diversa.

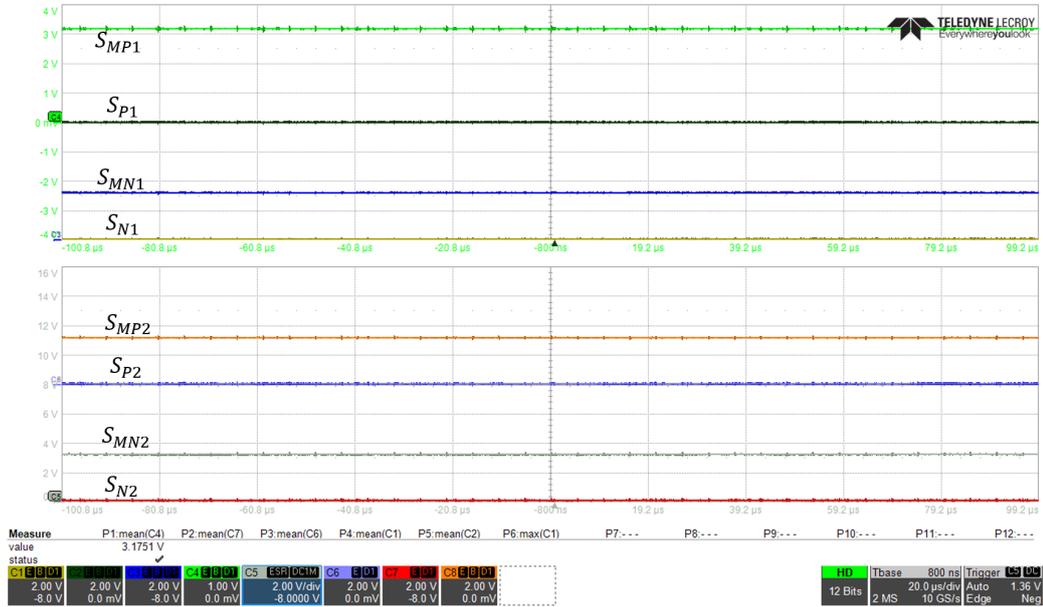


Figura 9.5: Segnali PWM $d = 0.5$.

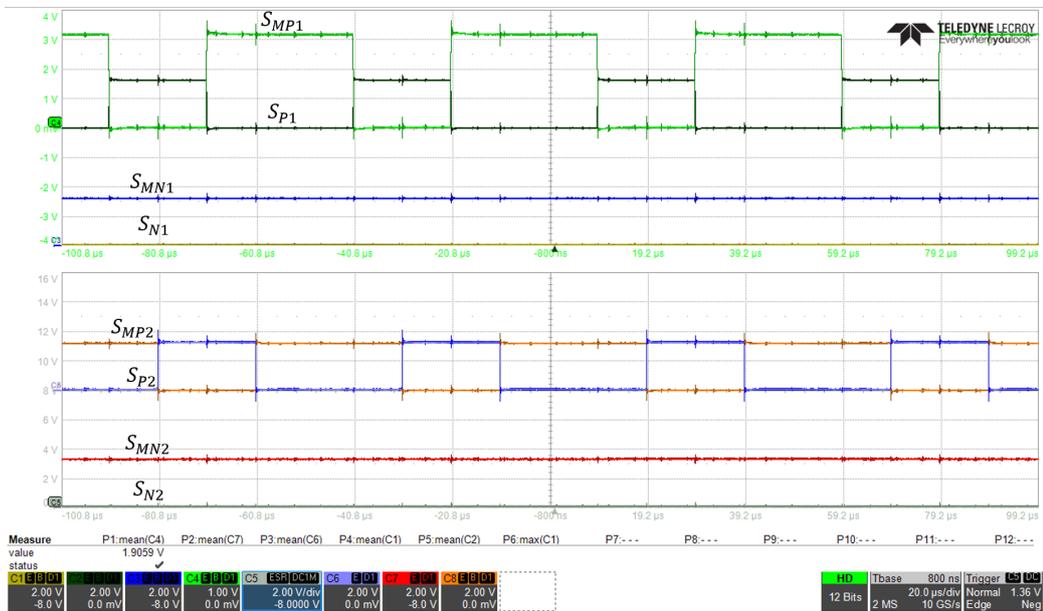


Figura 9.6: Segnali PWM $d > 0.5$.

La Fig. 9.7 mostra il caso in cui le due unità in parallelo lavorano con un ritardo di fase prestabilito (nel caso in esame il delay è di $10 \mu\text{s}$) con $d > 0.5$.

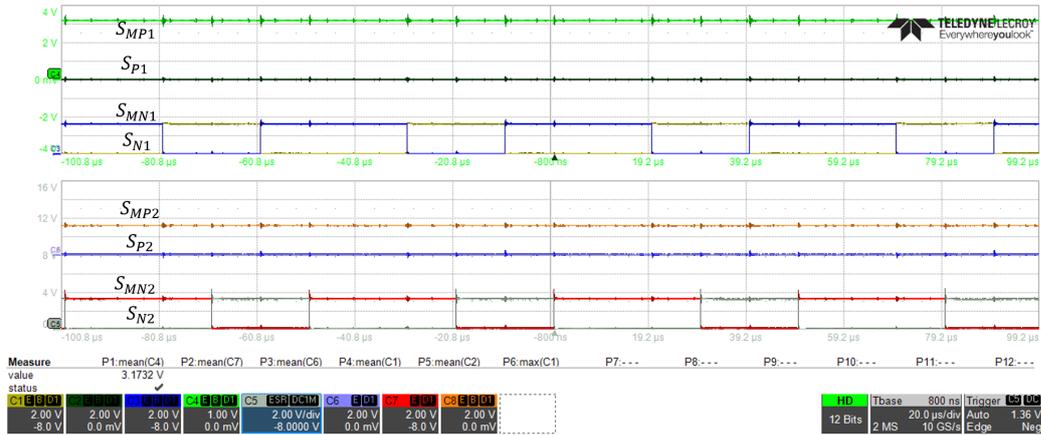


Figura 9.7: Segnali PWM $d < 0.5$ con $\text{delay}=10 \mu\text{s}$.

Capitolo 10

Conclusioni

Durante il lavoro di tesi si sono seguite tutte le fasi di progettazione del firmware di controllo. Si è partiti dalla scelta e all'analisi della toolchain di un μC dual core, passando per la configurazione delle periferiche necessarie fino ad arrivare alla fase di testing sperimentale sulla scheda MCU in modo da validarne il funzionamento.

Il presente progetto può rappresentare un punto di partenza per eventuali sviluppi futuri come:

- Fine assemblaggio e validazione sperimentale delle schede di misura e delle schede power
- Implementazione delle comunicazioni, con protocollo CAN, con il DC-DC modulare (4 moduli LLC da 12.5 kW)

Inoltre, durante la stesura del pinout, i pin del microcontrollore utilizzati per il debug non sono stati scelti a caso. Poichè tali pin possono svolgere anche altre funzioni (se opportunamente configurati), ne sono stati riservati alcuni che consentono di implementare eventuali comunicazioni (CAN e USART) aggiuntive. Un'altra soluzione possibile è data dalla flessibilità introdotta con la scheda Carrier e con la scheda MCU; infatti, sarà possibile implementare il firmware di controllo su FPGA semplicemente progettando una nuova scheda di controllo da sostituire alla scheda MCU.

Appendice A

Valori di configurazione timer

Vengono riportate le formule utilizzate per configurare i parametri principali dei timer; tali formule sono esposte all'interno del Reference Manual del produttore. Viene fatta la differenza tra timer che possono contare solo up, ossia da 0 fino ad ARR-1 (o solo down, da ARR a 1), che possono contare sia up che down, cioè da 0 ad ARR e da ARR ad 1, e i timer della periferica HRTIM che rappresentano un discorso a parte, ovvero da 0 a PERIOD. Si riporta il modo in cui sono stati ricavati tali valori in funzione della f_{sw} voluta e della frequenza in ingresso alle periferiche TIMER ed HRTIM f_{CLOCK} . Si ricorda che quest'ultimo parametro è settato al massimo valore possibile per il μC STM32H745ZI e può essere diminuito, variando il parametro PSC (Prescaler), rallentando il conteggio fino a $2^{16} - 1$ volte.

$$ARR_{up,counter} = \frac{f_{clock}}{f_{sw}} - 1 \quad (A.1)$$

$$ARR_{up/down,counter} = \frac{f_{clock}}{2 * f_{sw}} \quad (A.2)$$

$$HRTIM_{PERIOD} = \frac{f_{clock}}{f_{sw}} \quad (A.3)$$

Si riporta in Fig. A.1, come aiuto alla comprensione, uno schema a blocchi riassuntivo dei timer utilizzati ed il proprio scopo.

APPENDICE A. VALORI DI CONFIGURAZIONE TIMER

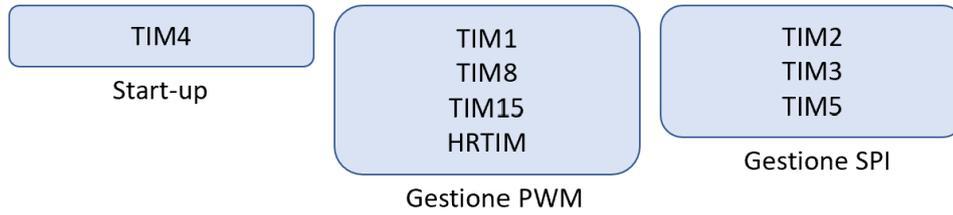


Figura A.1: Riassunto timer utilizzati

| TIMER | f (kHz) | ARR | f_{clock} (MHz) |
|-------|-----------------------------------|-------------|-------------------|
| TIM1 | $f_{sw} = 20$ | 6000 | 240 |
| TIM8 | $f_{sw} = 20$ | 6000 | 240 |
| TIM15 | $f_{sw} = 20$ | 11999 | 240 |
| HRTIM | $f_{sw} = 20$ | 12000 | 240 |
| TIM2 | $f_{sw} = 40$ | 5999 | 240 |
| TIM3 | ONE PULSE MODE | COMPARE2_CS | 240 |
| TIM5 | $f_{SCLK} = 1000$ | 239 | 240 |
| TIM4 | ONE PULSE MODE START-UP=400 ms | 65520 | 240 PSC=1500 |

Figura A.2: Esempio valori ARR

Inoltre, si riporta una figura (Fig. A.2) in cui vengono messi in evidenza i valori contenuti nel valore di Auto Reload Register e, fatta eccezione per i canali dei timer che riproducono in uscita il segnale PWM, i valori di CCR. È opportuno osservare che sono stati fissati, in maniera del tutto arbitraria i seguenti parametri:

con:

$$COMPARE2_{CS} = T_{acqui,US} * f_{clock} + COMPARE1_{CS} \quad (A.4)$$

$$COMPARE1_{CS} = DELAY_{CS} * f_{clock} \quad (A.5)$$

dove $DELAY_{CS}$ è il ritardo tra inizio ISR e fronte di discesa del CS, $T_{acqui,US} = N_{BIT} * T_{SCLK}$, è la durata del processo di acquisizione definito in μs , ed infine T_{SCLK} è il periodo del segnale del clock SPI.

Bibliografia

- [1] Electric vehicle on-board chargers and charging stations. <https://circuitdigest.com/article/electric-vehicle-on-board-chargers-and-charging-stations>.
- [2] Stazione di ricarica per veicoli elettrici. <https://it.wikipedia.org/wiki/Stazione-di-ricarica-per-veicoli-elettrici>.
- [3] Guida alla ricarica. <https://www.e-station.it/guida-alla-ricarica.html>.
- [4] Normativa per stazioni di ricarica pubbliche per veicoli elettrici. <https://www.emobitaly.it/normativa-per-stazioni-di-ricarica-pubbliche-per-veicoli-elettrici/>.
- [5] F. Mandrile M. Gregorio e R. Bojoi D. Cittanti, E. Vico. Iterative design of a 60 kw all-si modular llc converter for electric vehicle ultra-fast charging. *in press*.
- [6] M. Schweizer and J. W. Kolar. Design and implementation of a highly efficient three-level t-type converter for low-voltage applications. *IEEE Transactions on Power Electronics*, 28(2):899–907, 2013.
- [7] H. Mirante. Progettazione di convertitori ac/dc per sistemi di ricarica ultrafast delle batterie. Master’s thesis, Politecnico di Torino, 2019.
- [8] S.M. Di Franca. Modulation techniques for ac/dc converters in ultrafast battery charger applications. Master’s thesis, Politecnico di Torino, 2019.
- [9] M. Gregorio e R. Bojoi D. Cittanti. Digital multi-loop control of a 3-level rectifier for electric vehicle ultra-fast battery chargers. *in press*.
- [10] Datasheet stm32h745xi/g. DS12923 Rev 1.
- [11] μ vision ide. <http://www2.keil.com/mdk5/uvision/>.

BIBLIOGRAFIA

- [12] STMicroelectronics. Stm32h7x5/x7 dual-core microcontroller debugging, 01-Jul-2019. Rev 1.
- [13] STMicroelectronics. Getting started with projects based on dual-core stm32h7 microcontrollers in stm32cubeide, 21-Nov-2019. Rev 1.
- [14] Reference manual stm32h745/755 e stm32h747/757. RM0399 Rev 3.
- [15] Datasheet mcp4725. Revision E (October 2010).
- [16] Datasheet mcp4728. Revision D (June 2009).
- [17] STMicroelectronics. Stm32f3 technical training.
- [18] Adc ad approssimazioni successive. <https://it.wikipedia.org/wiki/ADC-ad-approssimazioni-successive>.
- [19] STMicroelectronics. Getting started with sigma-delta digital interface on applicable stm32 microcontrollers, 30-Mar-2018. Rev 1.
- [20] A. Locatelli. Hardware design and testing of a 50 kw t-type active rectifier for fast charging applications. Master's thesis, Politecnico di Torino, 2020.