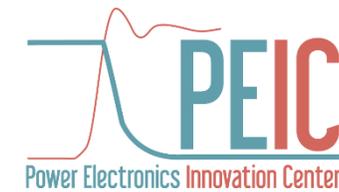




POLITECNICO
DI TORINO



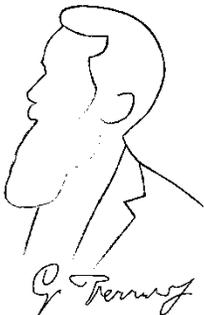
Controllo digitale firmware per microcontrollore per un caricabatteria ultrafast per veicoli elettrici

Relatore:

Prof. Radu Bojoi

Correlatore:

Matteo Gregorio



Dipartimento Energia "Galileo Ferraris"

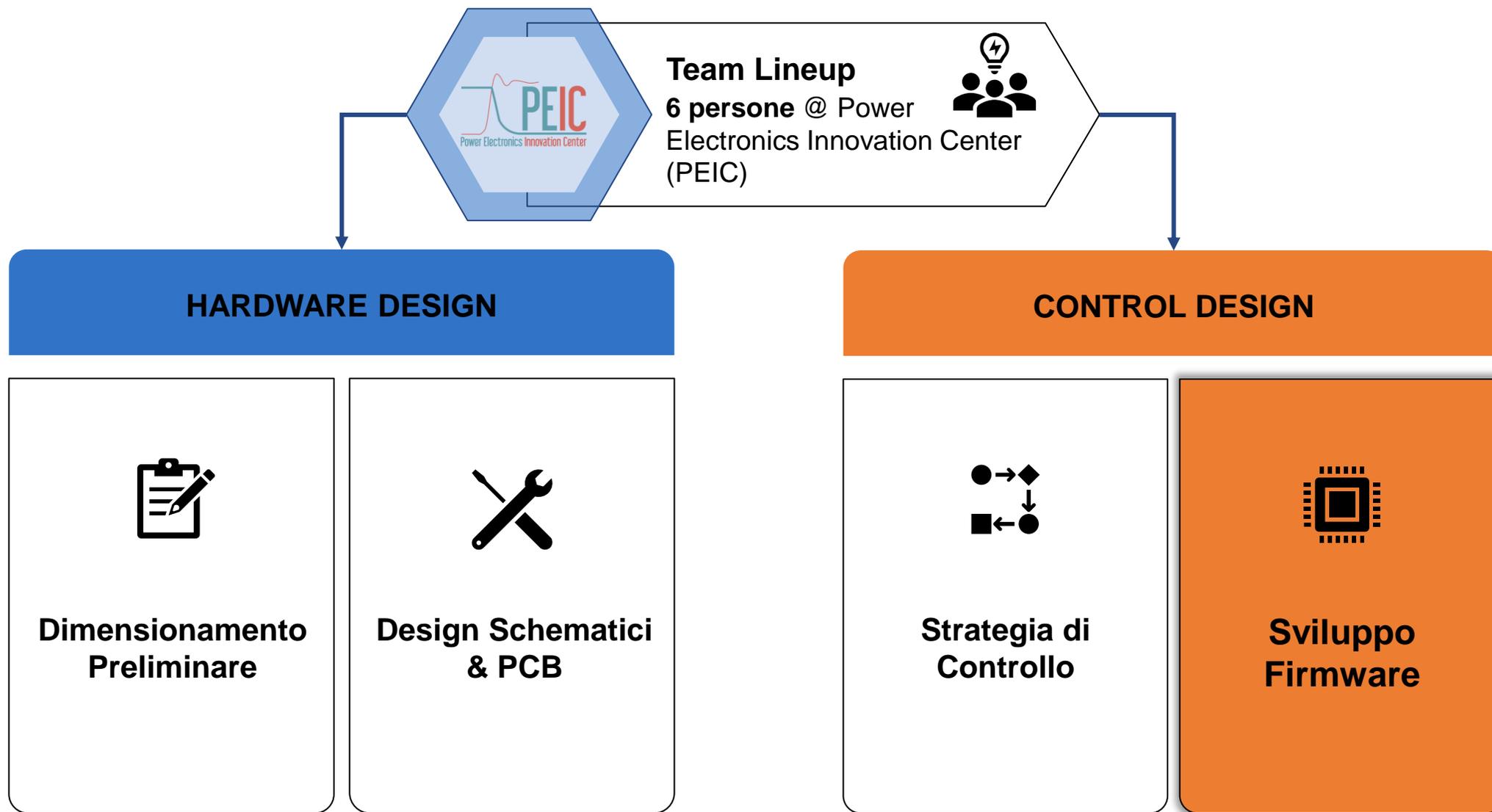
Politecnico di Torino, Italy

Candidato:
Simone Caiazza

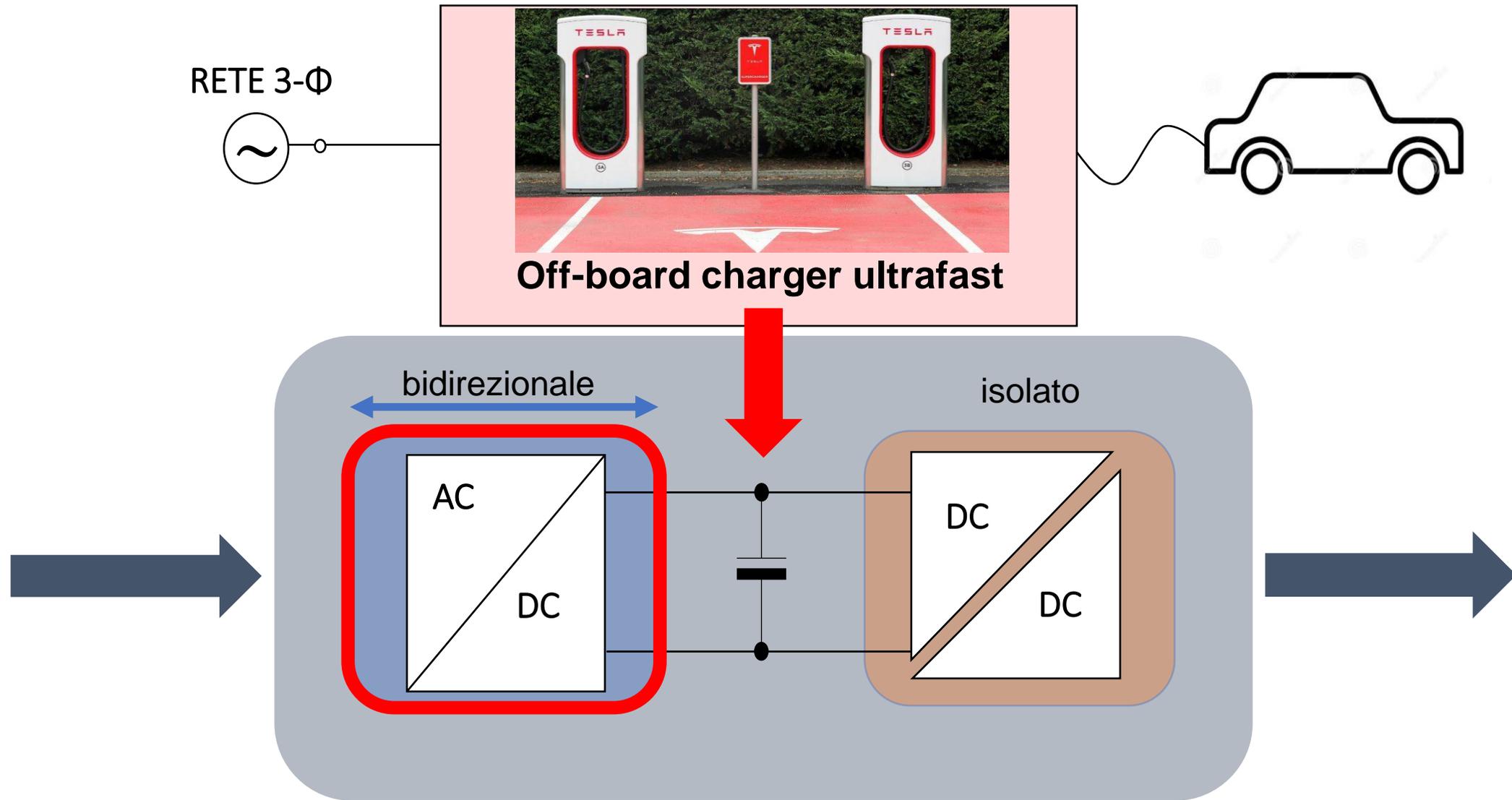


13/11/2020

AFE Design Team

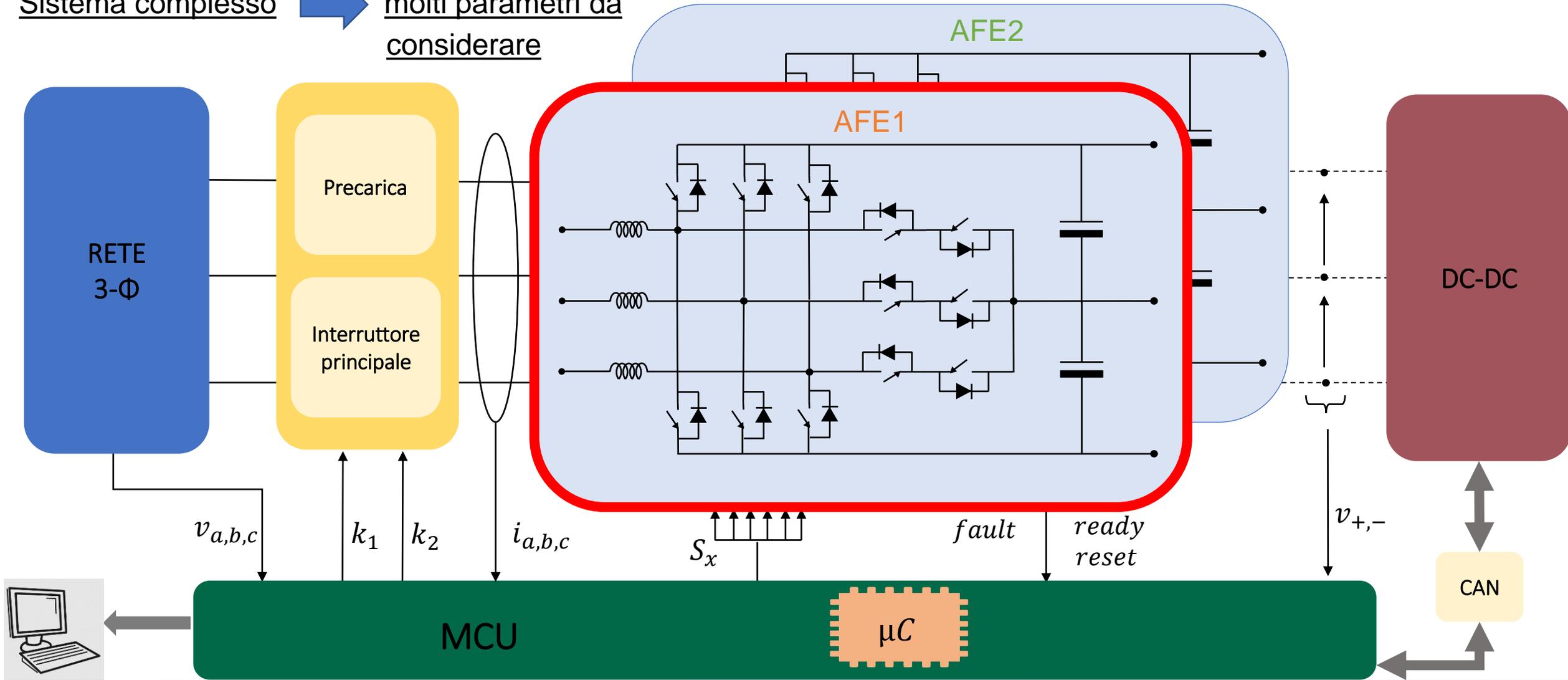


Off-board charger per veicoli elettrici

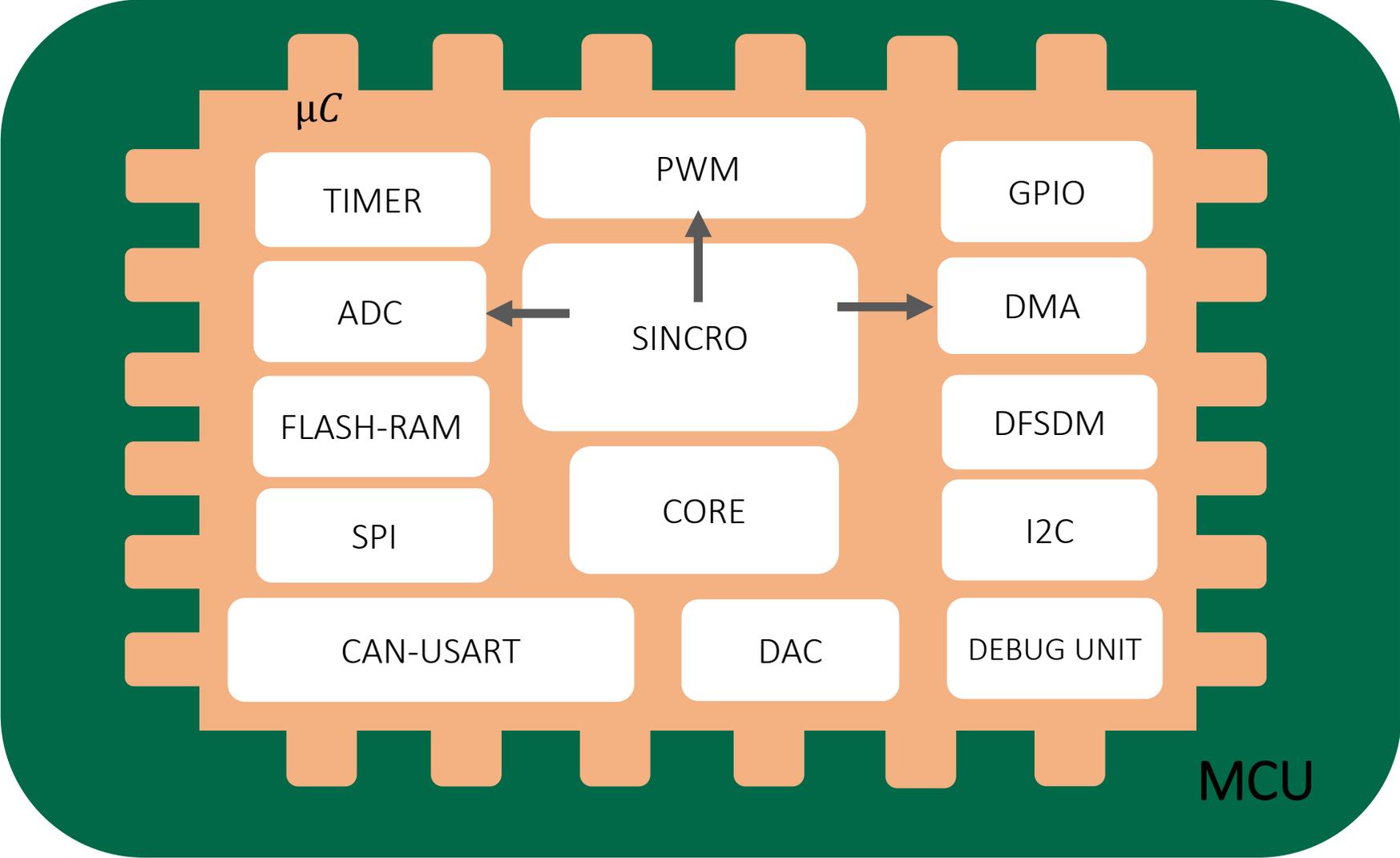


AFE T-Type 3 livelli

Sistema complesso → multi parametri da considerare



MCU

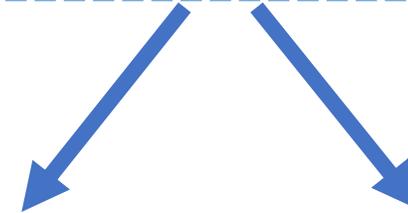


Microcontrollore scelto



Microcontrollore STM32H745ZI dual core:

1. ARM Cortex-M7 con $f_{clock} = 480 \text{ MHz}$
2. ARM Cortex-M4 con $f_{clock} = 240 \text{ MHz}$



PACKAGE TFBGA:

- Necessario per generare contemporaneamente tutti i segnali PWM

C-M7:

- Anelli di tensione
- Gestione comunicazioni

C-M4:

- Anelli di corrente
- Gestione PWM
- Gestione acquisizioni

Problematiche affrontate

➤ Programmazione e debug μC dual core



- Rappresenta una novità l'uso di un μC dual core per UFC, necessario per eseguire il controllo a **72 kHz ($< 14 \mu s$)**

➤ Implementazione PWM 3L in un μC



- Il μC può generare solo portanti triangolari da 0 ad un valore predefinito
- Come si implementa la tecnica PWM 3 livelli?

➤ Gestione timer PWM

➤ Acquisizioni di corrente e di tensione

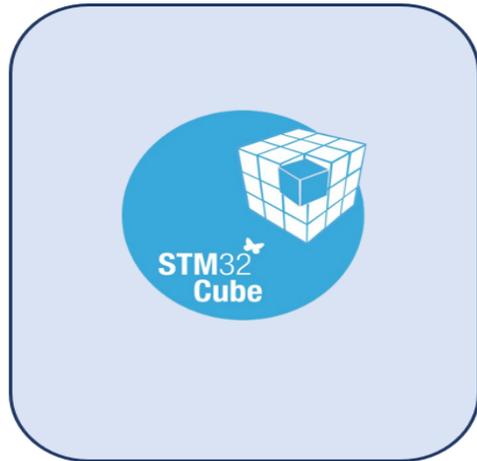
➤ Gestione comunicazioni SPI e I2C



- È necessario che tutte le periferiche coinvolte siano sincronizzate tra di loro

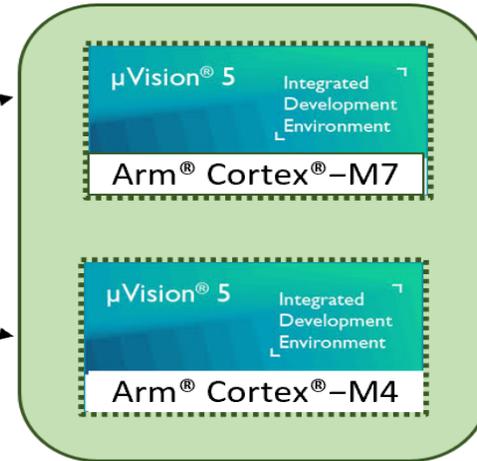
Toolchain

Programmazione μ C dual core



FileName.ioc

Tool grafico che fornisce il codice di inizializzazione di sistema e delle periferiche del μ C



FileName.uvprojx

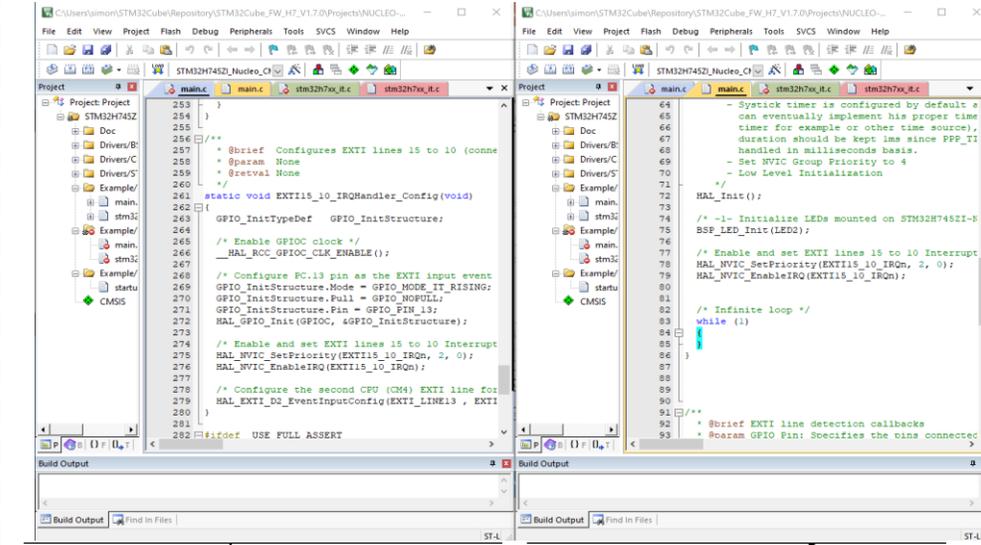
IDE che consente di gestire in maniera agevole la programmazione ed il debug di un μ C dual core

→ *_CM7*

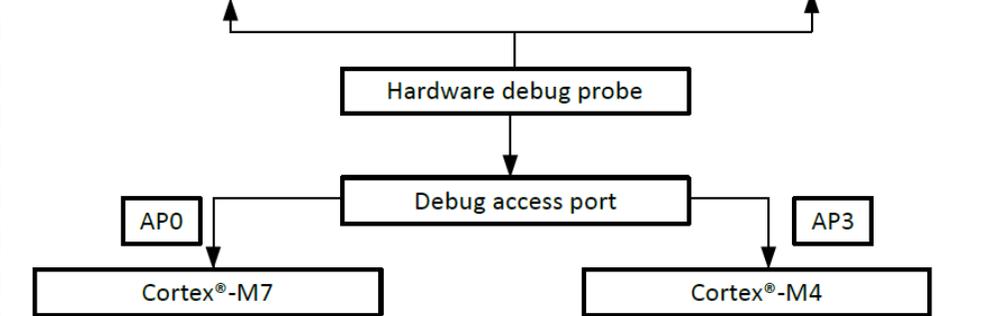
→ *_CM4*

Debug μ C dual core

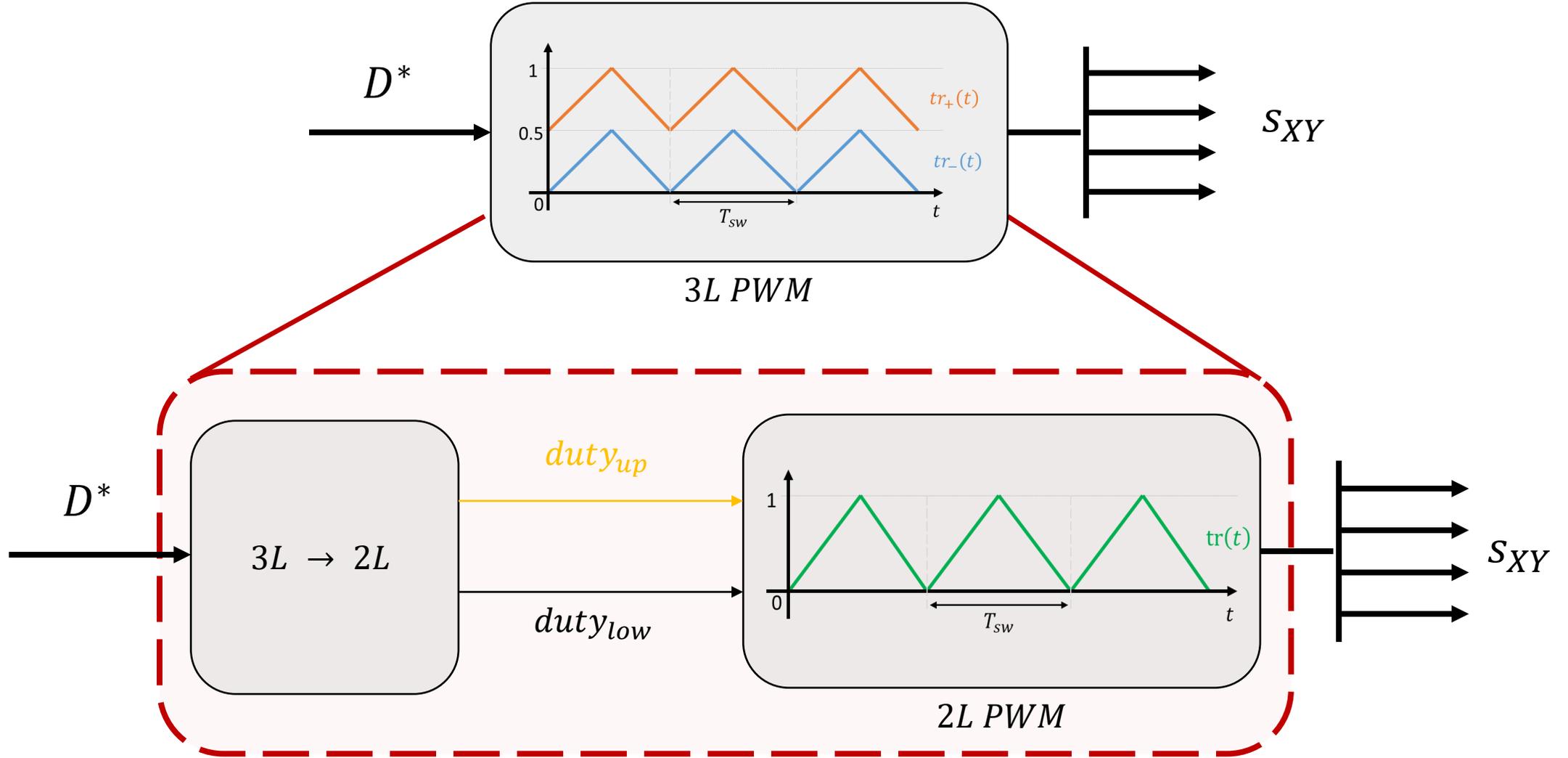
Cortex-M7



Cortex-M4

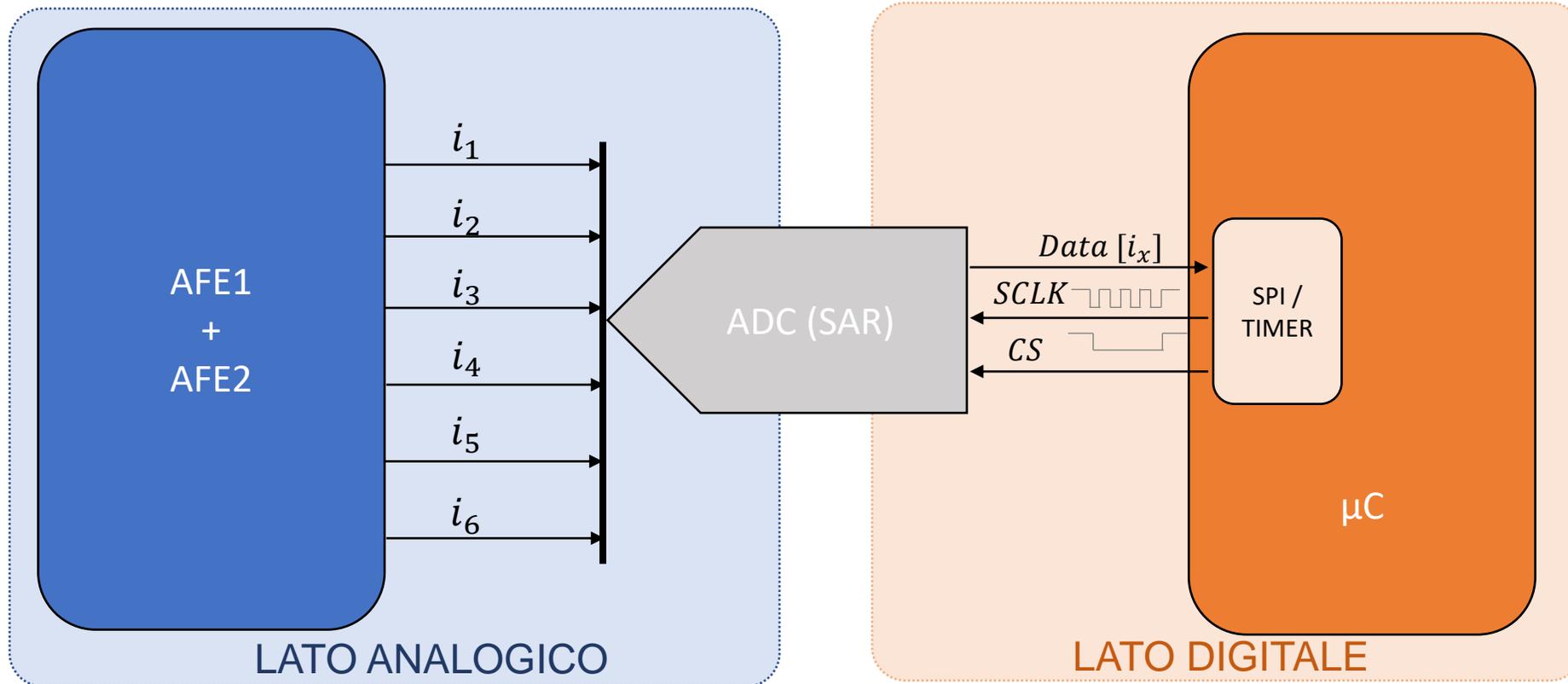


3L T-Type PWM



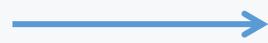
Acquisizioni di corrente - Implementazione

Convertitore analogico-digitale SPI per acquisire le correnti di fase



Campionamenti:

Sincronizzati



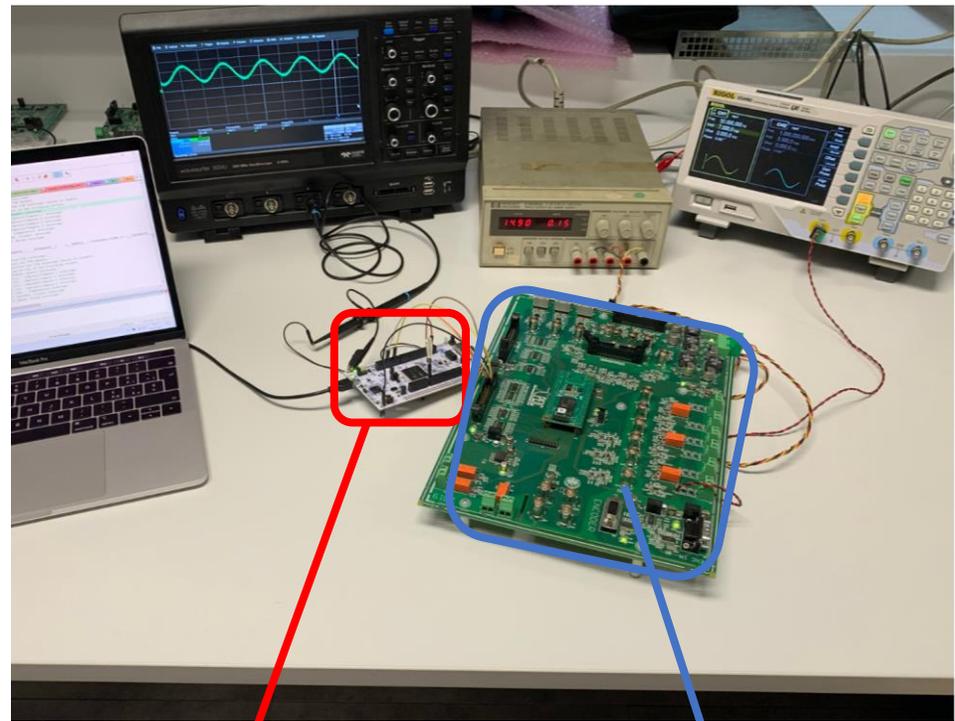
Ritardo da ISR configurabile

N bit



Configurabile da timer (CS)

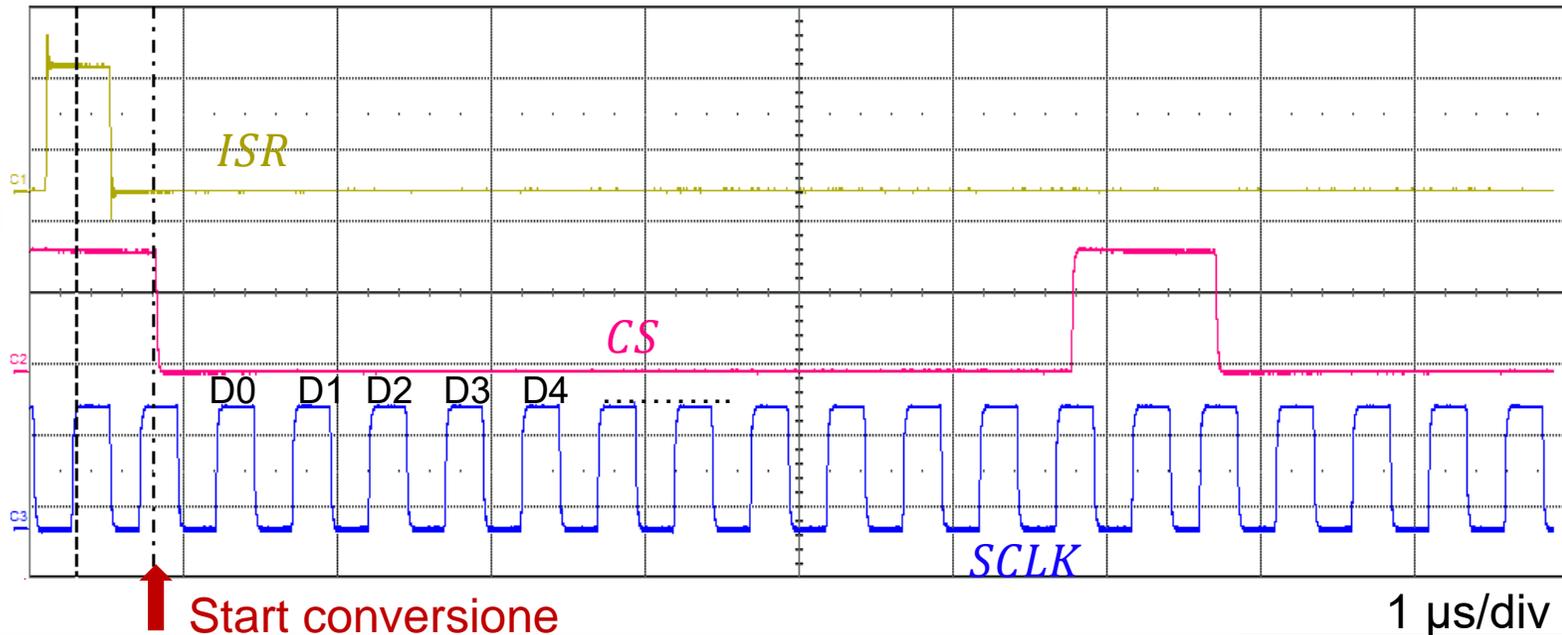
Acquisizioni di corrente – Validazione sperimentale



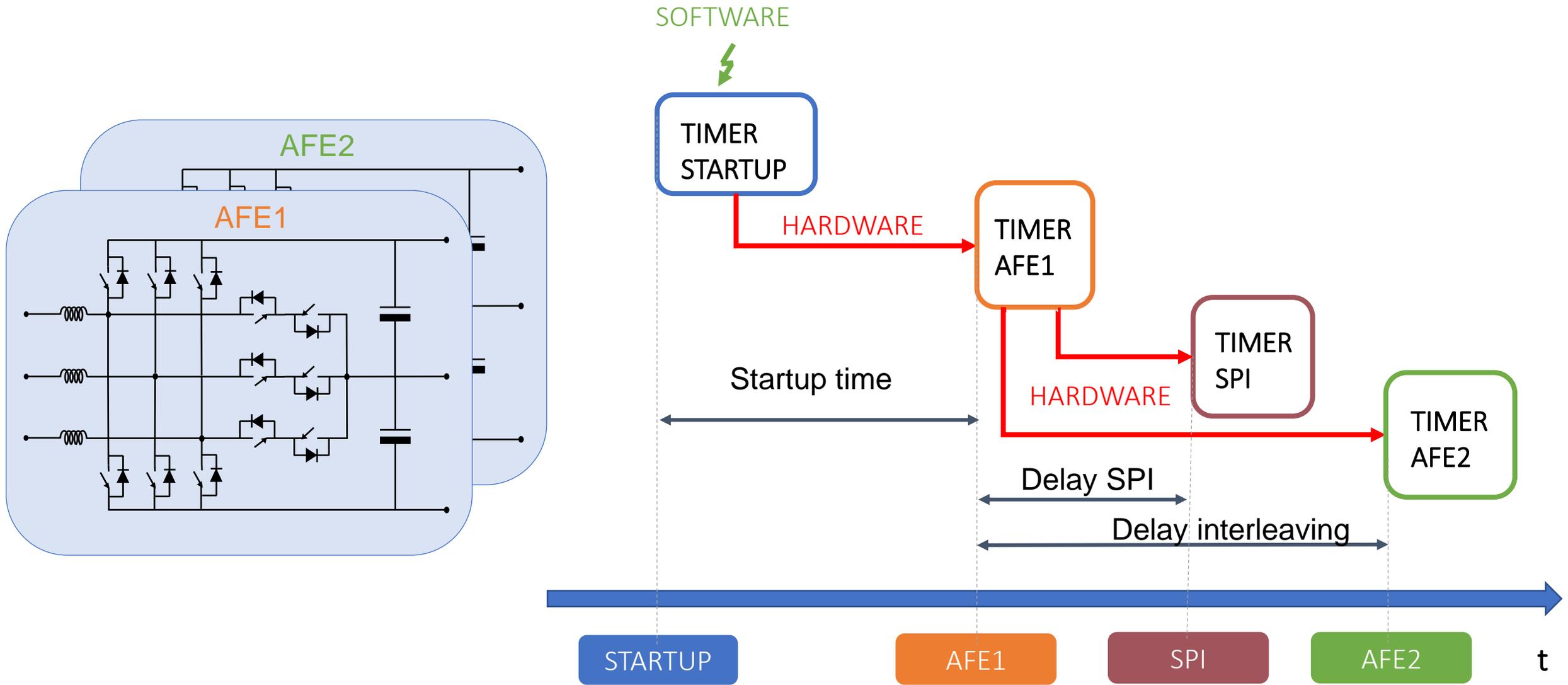
NUCLEO

SCHEDA
CON
ADC SPI

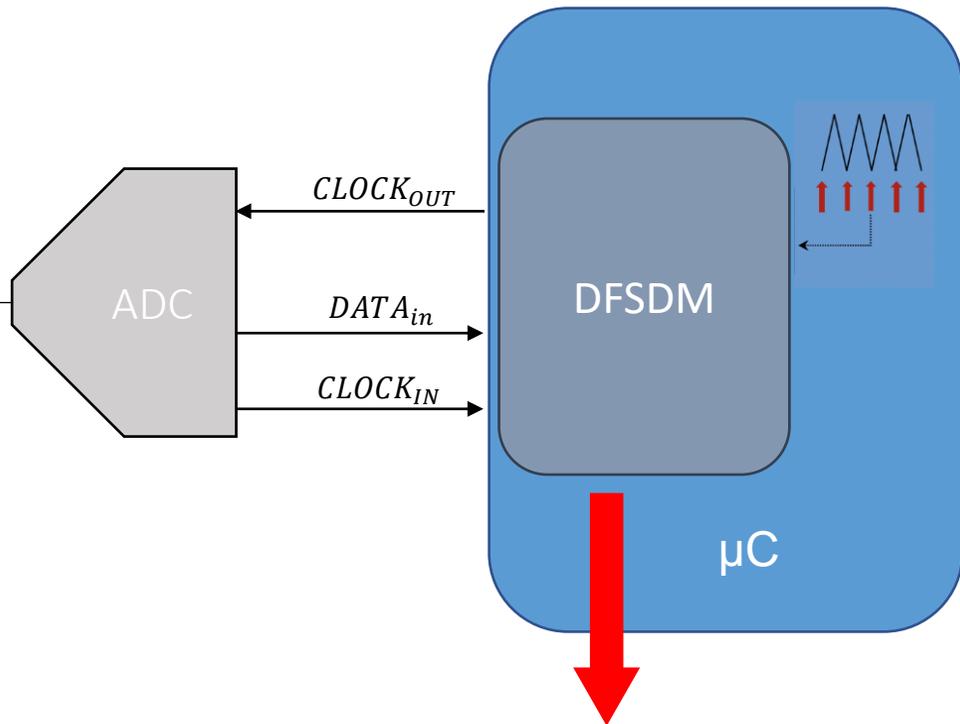
- *ISR* → $f = 72 \text{ kHz}$
- *CS* → $f = 144 \text{ kHz}$
- *SCLK* → $f = 2 \text{ MHz}$
 $N = 14 \text{ bit}$



Sincronizzazione timer

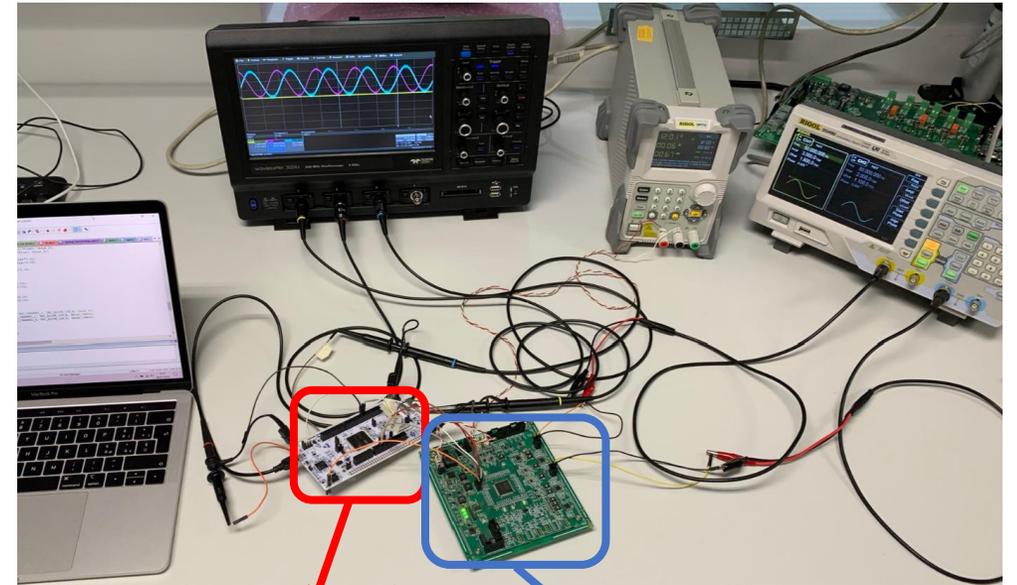


Acquisizioni di tensione – Implementazione + Validazione



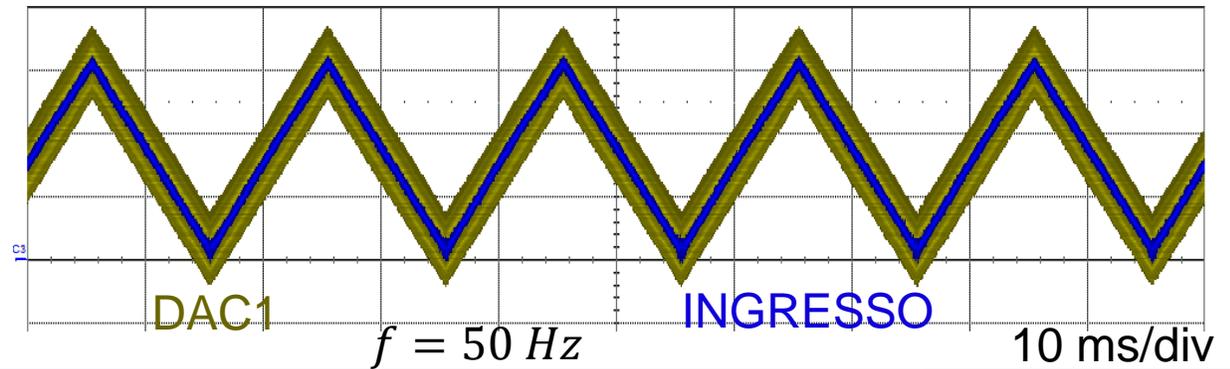
Parametri filtro:

- FOSR (Filter Over Sampling Ratio)
- FORD (Filter Over Sampling Order)
- IOSR (Integrator Over Sampling Ratio)



NUCLEO

SCHEMA CON ADC SD

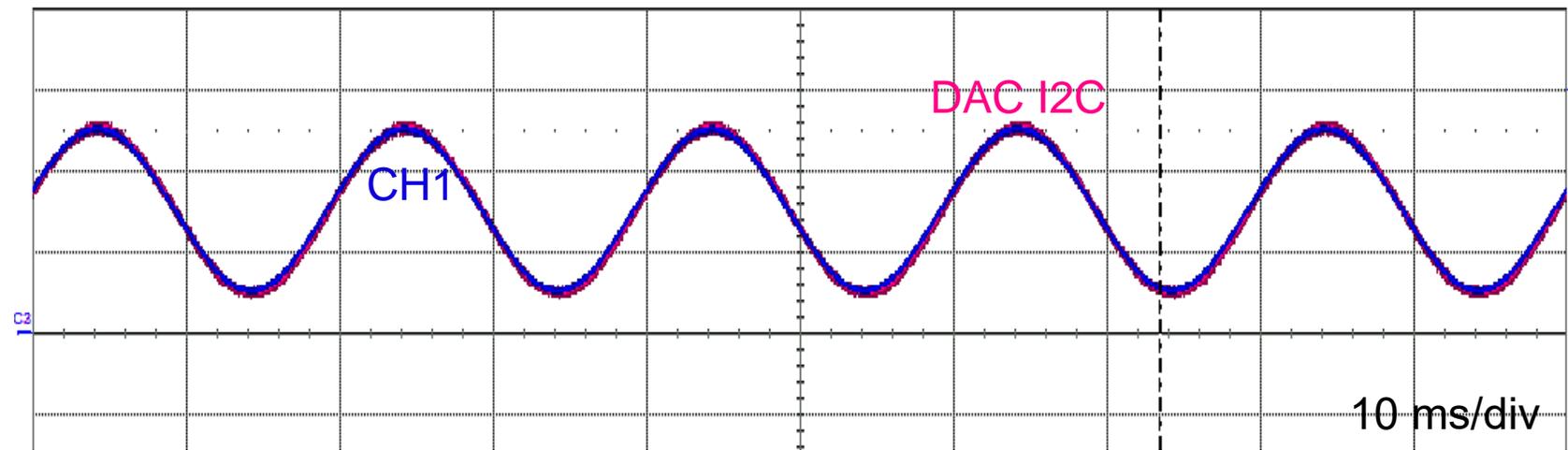
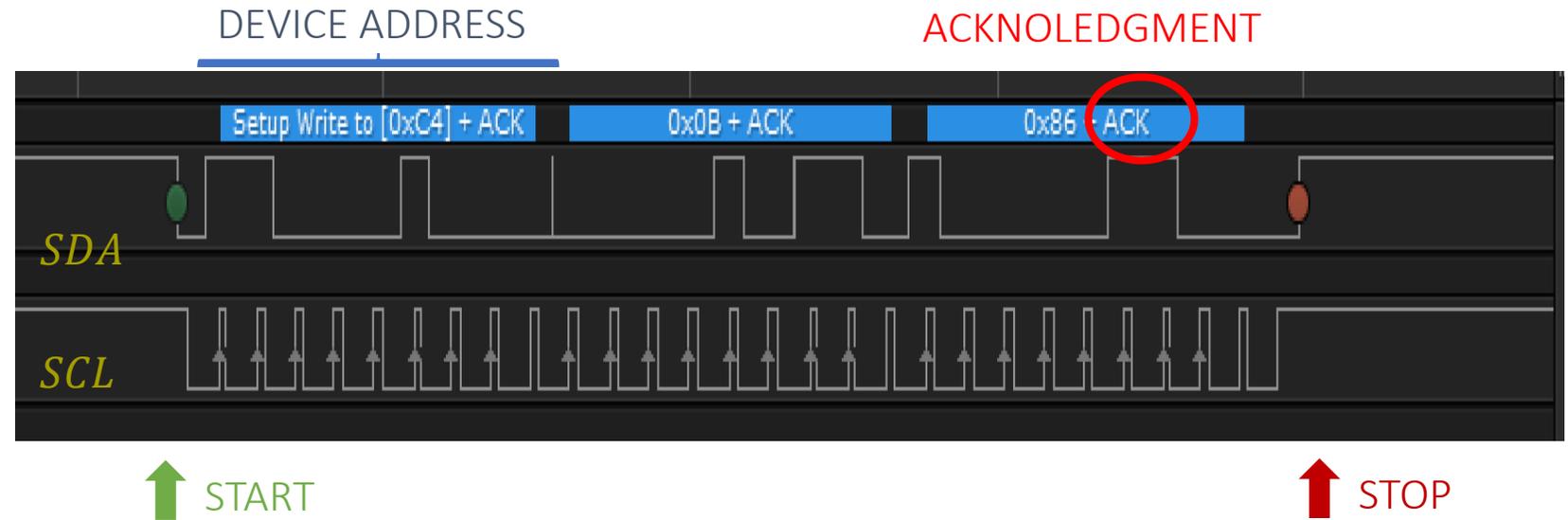


Comunicazione DAC I2C

Per ricostruire word digitali in forme d'onda analogiche si è utilizzato un **DAC esterno**, il quale comunica con il micro con **protocollo I2C**

$$f = 50 \text{ Hz}$$

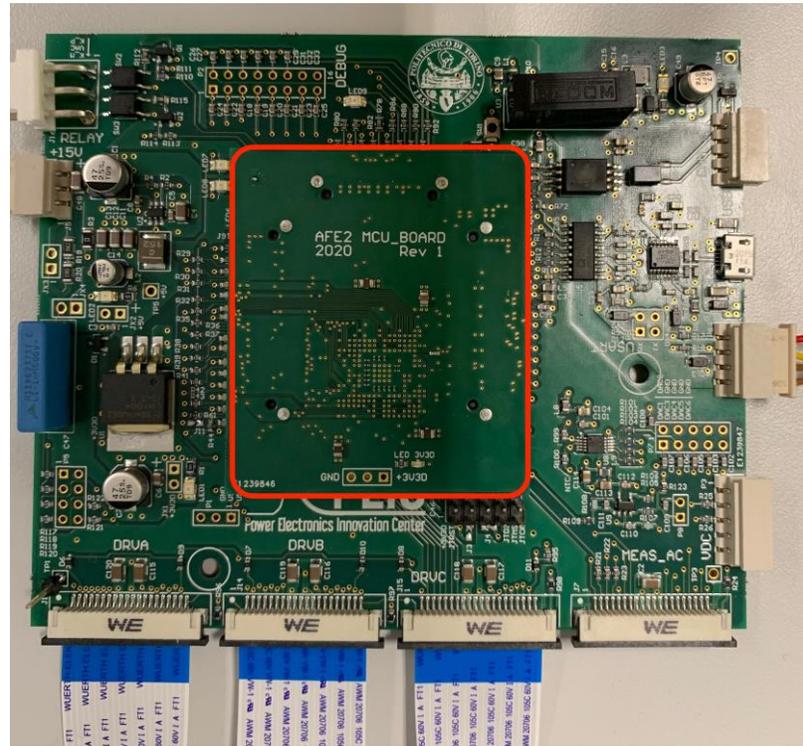
- Segnale analogico in ingresso
- Uscita DAC I2C



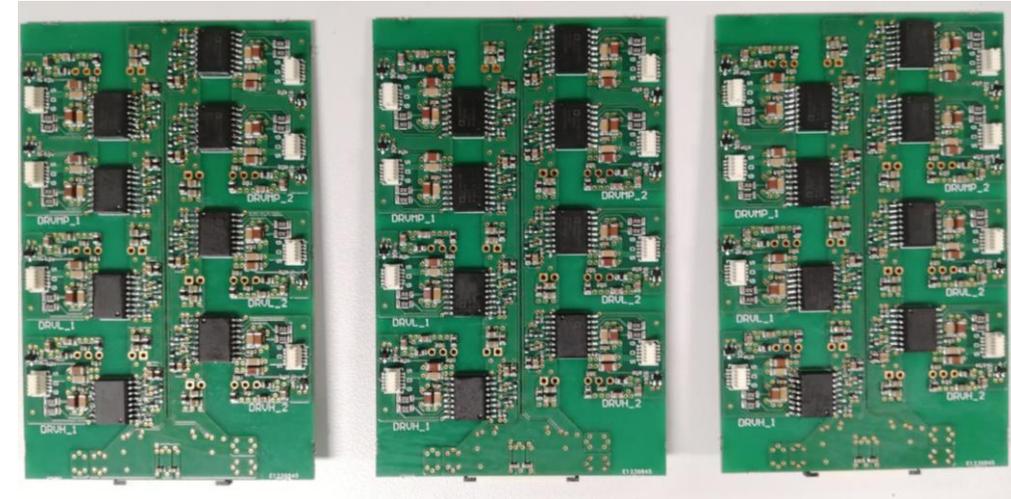
Assemblaggio schede



Schede MCU



Scheda Carrier + Scheda **MCU**



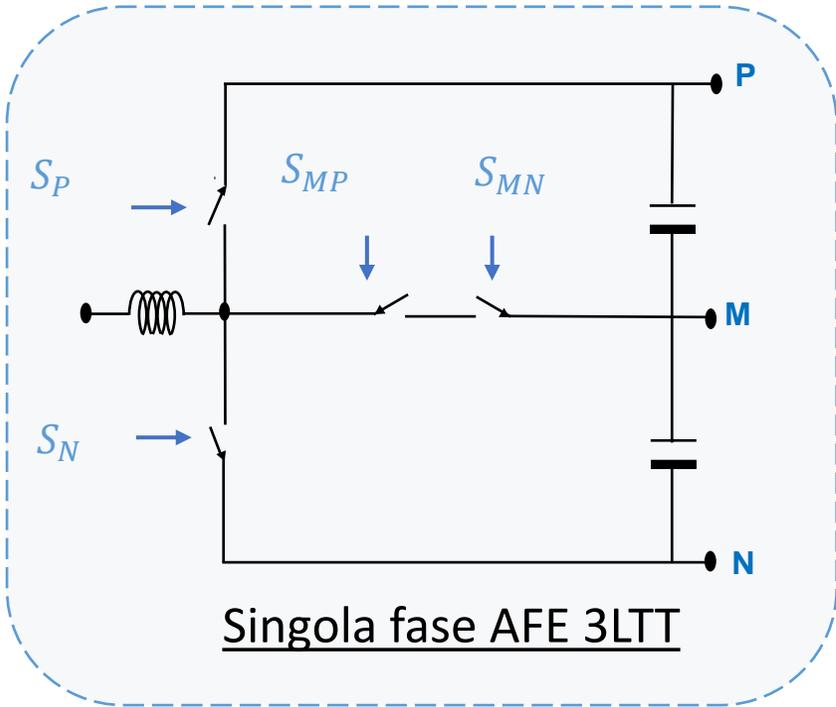
Schede Driver

Test segnali PWM su schede finali

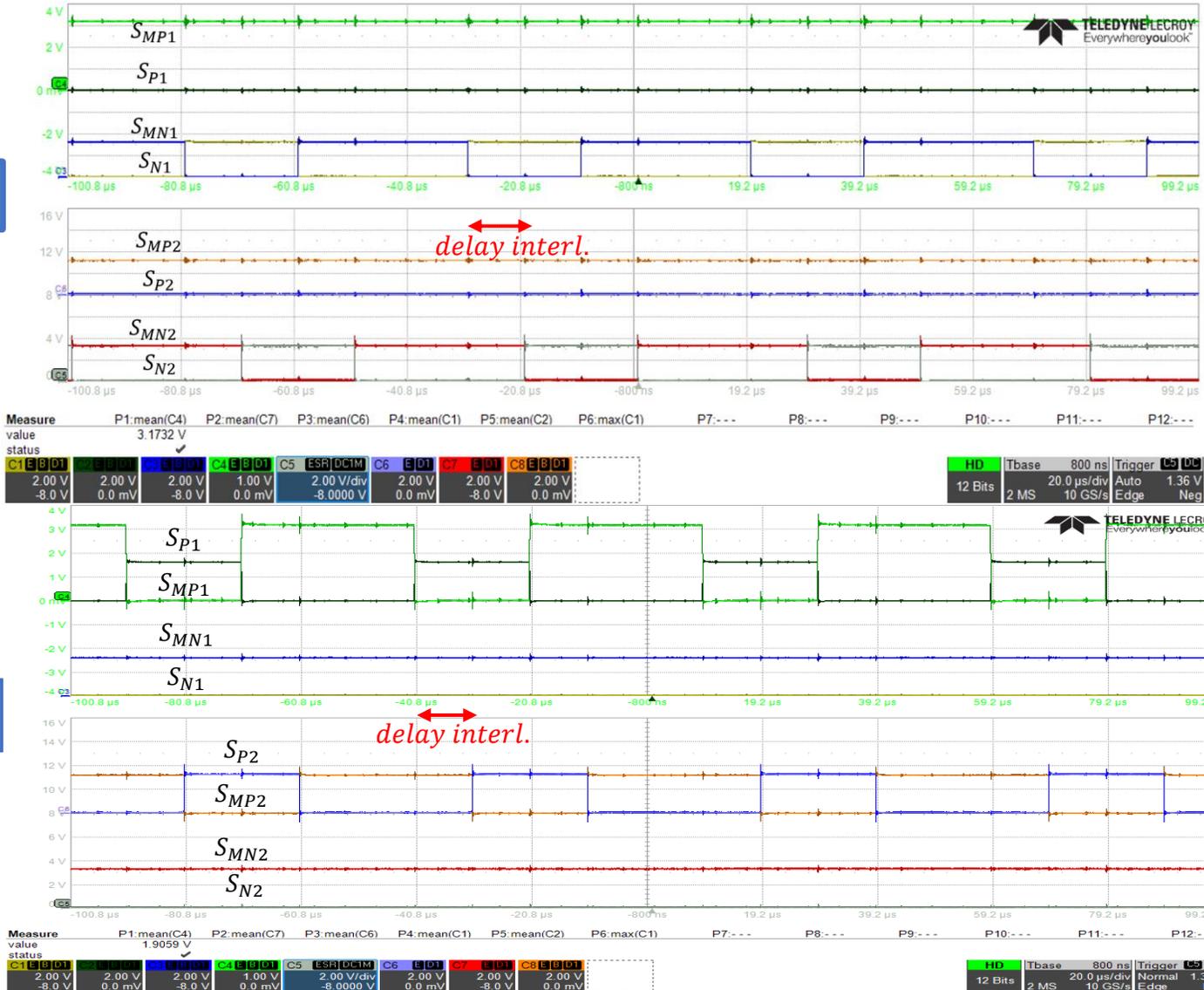
Comandi switch stessa fase con:

Delay interleaving = $10 \mu\text{s}$

$d < 0.5$



$d > 0.5$



Conclusioni e contributi personali

Durante il lavoro di tesi ho seguito tutte le fasi di progettazione del firmware di controllo

I miei contributi personali sono stati:

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

Grazie per l'attenzione!