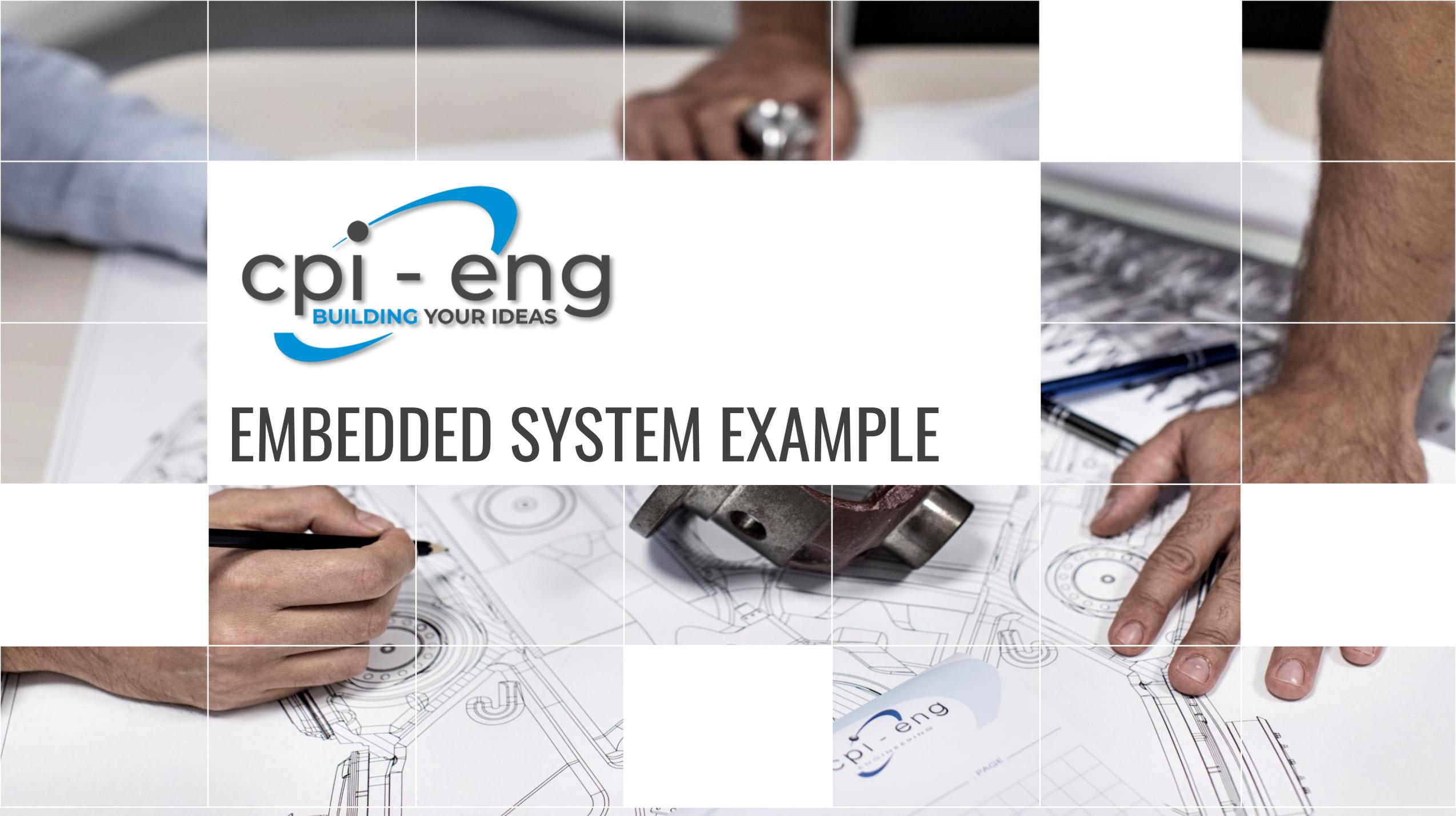
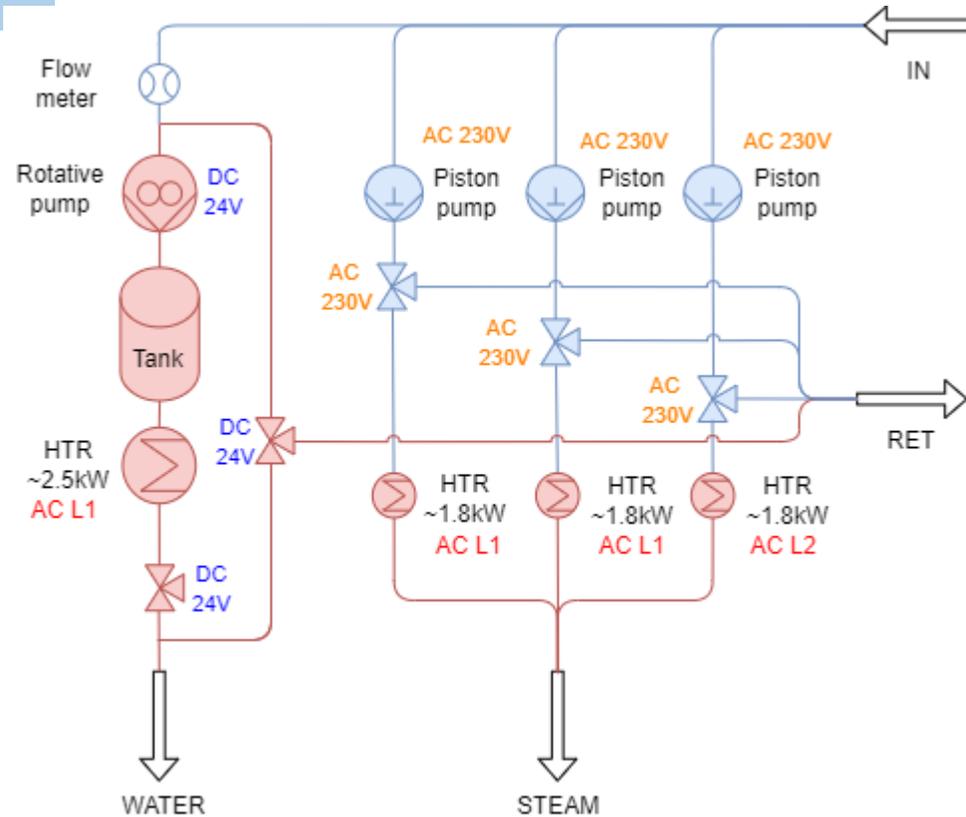




EMBEDDED SYSTEM EXAMPLE



Water Steam module for professional coffee machine



Module for steam hot water generation for a coffee machine without traditional boiler that uses “instantaneous” electric heater elements

Steam system

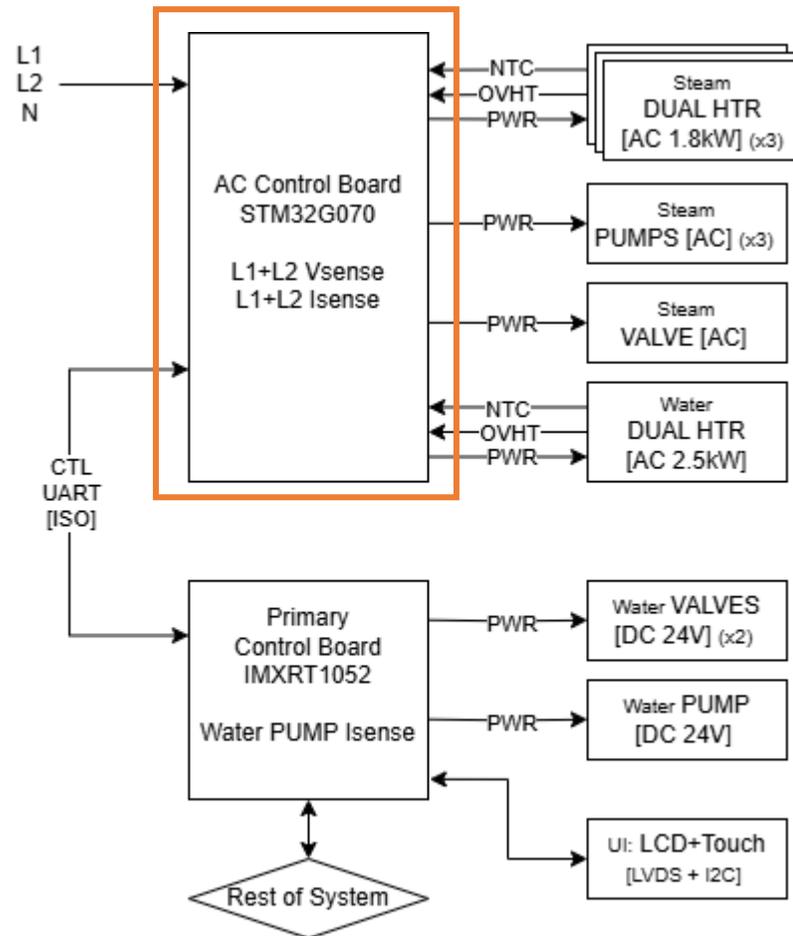
- 3 identical circuits comprising a pump, valve and 1.8kW heater. All elements powered by 230V AC
- Total power: 5.2kW on two phases

Hot water system

- Single circuit with 2.4kW heater (230V AC), pump and valves are powered by 24V DC

Steam and hot water system cannot be used simultaneously due to input power limits

Water/Steam Module – Electronics Architecture



The system is divided in two PCBs

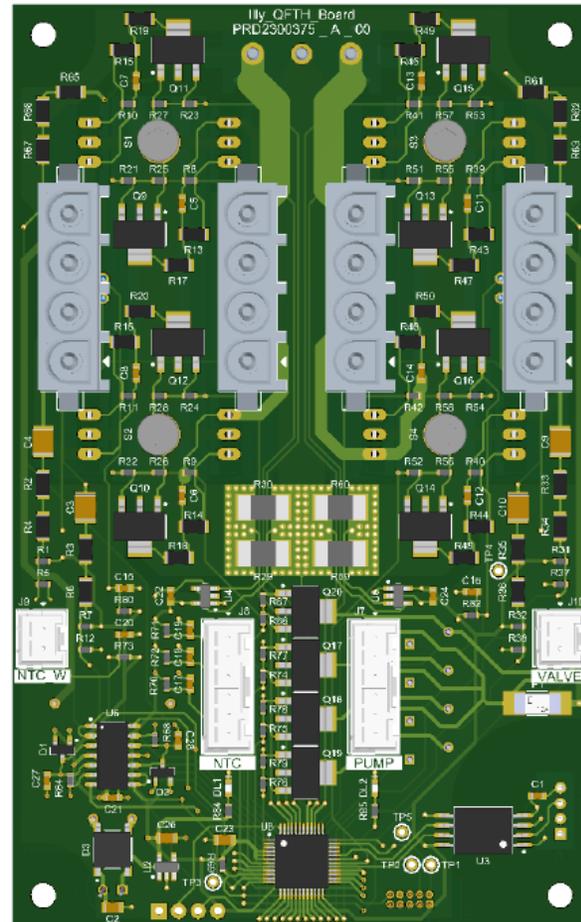
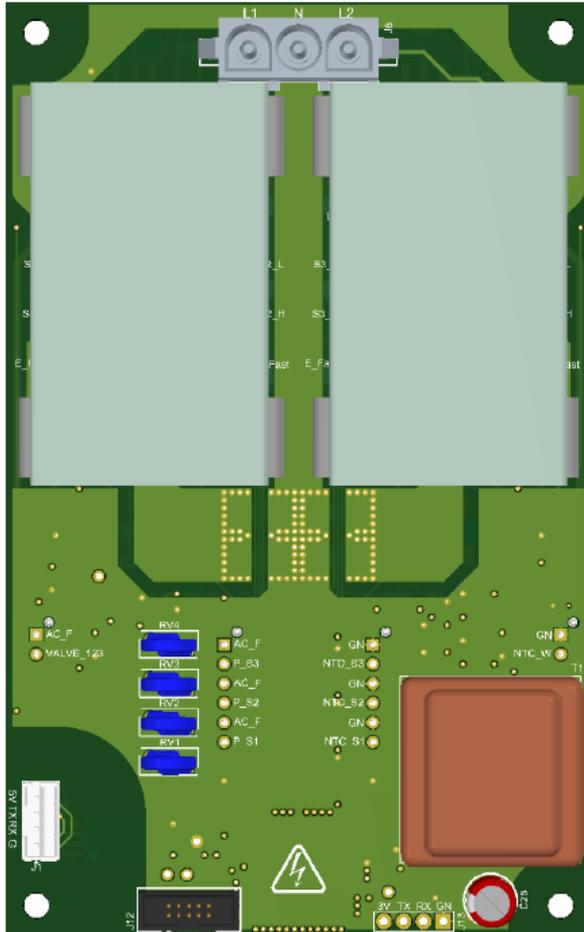
AC elements control board

- Drivers all elements powered by 230V AC and reads associated sensors. The entire system is referred to AC neutral
- Handles the control (temperature loop) for the system
- MCU: ST STM32G070CB (Cortex-M0+ @ 64MHz).
- Talks to the main control board via an isolated serial interface (UART) at 50 kb/s

Main control board

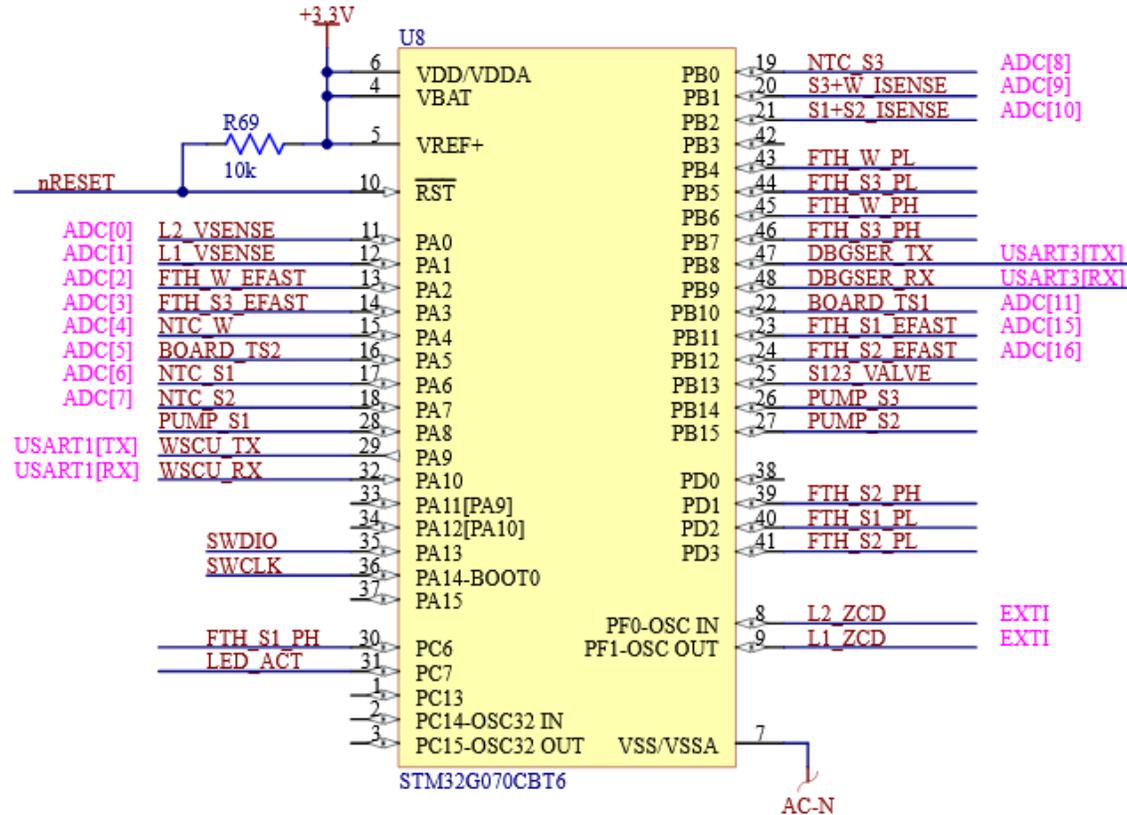
- Drives all elements powered by DC 24V of the water system and handles the control loop (flow and temperature) for said system
- Handles the main user interface (7 in touch display)
- MCU: NXP iMXRT1052 (Cortex-M7 @ 600MHz) with external SDRAM + 2xNOR Flash

AC elements control board - HW



- 8 power TRIACs (2 per heater), with total current measurement (L1 and L2)
- 4 low power TRIACs for pumps and valves
- NTC sensor readout for outlet temperature and overheat sensor readout (AC output) for each heater
- Line voltage readout for L1 and L2. Zero Cross signal generation for TRIAC driving.

AC elements control board – MCU



Used peripherals

- GPIO
- ADC (14ch) with DMA
- EXTI (2 external interrupt for Zero-Crossing)
- USART for communication with the main controller (in Interrupt)
- 3 x TIMER (TIM15, TIM16, TIM17) for AC-synchronized event handling (ADC acquisition, TRIAC driving...)

STM32G0: Development environment

- Development environment based on Visual Studio Code, Cmake with Arm GCC compiler + picolibc
- RTOS: FreeRTOS 11.2.0
- Internally developed STM32 drivers based on ST LL libraries or direct register access (UART, ADC, GPIO) with full FreeRTOS integration
- For example, the UART driver is about 100% faster in interrupt mode compared to the ST HAL implementation
- No dynamic memory allocation
- Custom bootloader to allow for firmware updates from main control board
- The same environment is used also for other products using the same MCU or other STM32s
- 4 Main tasks (in priority order)
 - AC Loads Control
 - Main Control Loop
 - UART Rx Task (and System Init)
 - UART Tx Task
- ADC Acquisition in DMA mode and basic data processing uses 10-15% CPU time
- Main control loop running at 10Hz, AC load driving loop running at 50Hz (synchronized with AC Zero-Cross)

ADC acquisition details

14 signals of 3 different classes:

- 4 AC Full-Wave Signals (heater overheat sensors) [3 related to L1, 1 related to L2], peak detect type acquisition
- 4 AC Half-Wave Signals (misura tensione e corrente) [2 related to L1, 2 related to L2], peak detect type acquisition
- 6 DC Signals DC (NTC thermistor measurement)

ADC is used in free-running mode with the built-in sequencer: it always converts all 14 channels in order. Oversampling enabled to increase the ADC resolution from 12 to 13 bits (for extra dynamic range on overheat sensors).

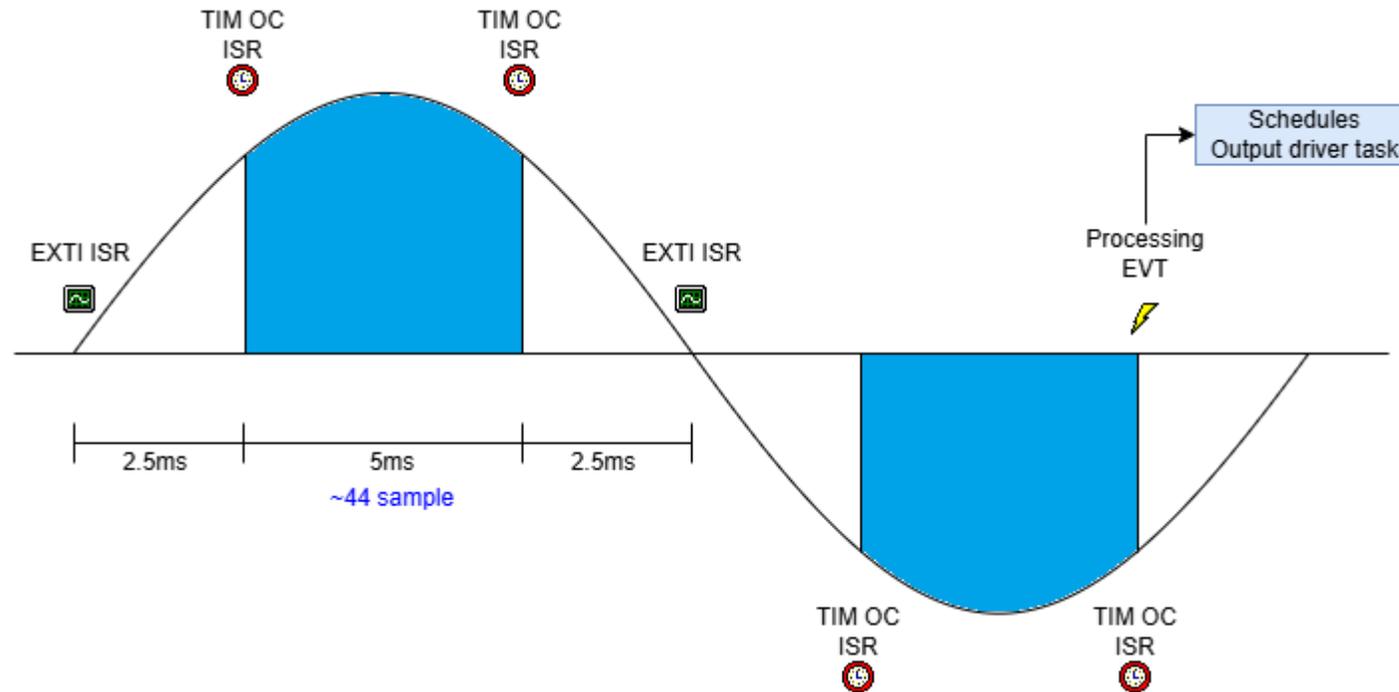
Conversion time for all 14 channels: 112µs

Conversion results transferred to RAM via DMA in two 14 uint16 buffers (buffer in use alternates between conversions). DMA is used in circular mode, using the HalfTransfer and TransferComplete interrupts to do data processing directly in the ISR → ~8.92kHz interrupt rate

ADC Acquisition – Zero Crossing Events

Timing diagrams for Z-Cross synchronized ADC related events

(For both power phases)





www.cpi-eng.it
info@cpi-eng.it

