Hardware bitácora

In this document I will put any important research, information, decisions, about the hardware, component selection.

# MCU selection

This project will need a cortex-M4, because will use an LCD TFT 7” screen for control and mode selection, will use basic graphics, a telecommunication module, multiple sensors, an actuator, battery management, will have an internet connectivity, buttons, indicators leds.

One of the main characteristics is the memory, and communication protocol to use the LCD screen, so, I will need to choose also a development board to make the POC, and write over the HAL, because first I need to prove all the functions and the firmware to improve it.

The development boar to choose is [STM32F429I-DISC](https://www.mouser.mx/ProductDetail/STMicroelectronics/STM32F429I-DISC1?qs=79dOc3%2F91%2Fed3%252BRc5JUCEw%3D%3D&mgh=1&vip=1&gad_source=4) the board have a lcd integrate to prove the firmware and will be good for POC.

## STM32F429I

This board has the following characteristics

**General**

- Architecture: ARM Cortex-M4

- Maximum frequency: 180 MHz

- Flash memory: 2 MB

- SRAM: 256 KB

- EEPROM: Not integrated

- Operating voltage: 1.7V - 3.6V

- Power consumption: ~80-150 mA

**Processing and graphics**

- FPU (Floating Point Unit): Yes, supports floating point operations

- Graphics accelerator: Chrom-ART Accelerator™ (DMA2D)

- LCD interface: LTDC (for TFT displays up to 1024x768)

- External memory controller: FMC (for SDRAM, NOR, NAND, SRAM, etc.)

**Peripherals**

- Timers/PWM: 14 advanced and general timers

- ADC: 3x 12-bit ADCs, up to 24 channels

- DACs: 2x 12-bit channels

- GPIOs: 114 general purpose pins

- DMA: 16 DMA channels

**Communication interfaces**

- USART/UART: 8 (up to 12 Mbit/s)

- I2C: 3

- SPI: 6

- CAN: 2 (CAN 2.0B)

- USB: USB 2.0 OTG FS and HS (with external PHY for HS)

- Ethernet: Yes (MAC with IEEE 1588 support, external PHY required)

- SDIO: Yes, SD/MMC compatible

**Security**

- Watchdog (IWDG and WWDG)

- CRC (Cyclic Redundancy Check)

- Hardware Protected Memory (MPU)

Ideal applications

- Embedded systems with basic UI

- Devices with medium-sized TFT display

- Applications with Ethernet or USB connectivity

- Motor and sensor control

## STM32H723ZG

Main characteristics

**General**

Architecture: ARM Cortex-M7

Maximum frequency: 550 MHz

Flash memory: 2 MB

SRAM: 1 MB

EEPROM: Not integrate

Operating voltage: 1.62V - 3.6V

Power consumption: ~180-250 m

**Processing and graphics**

FPU (Floating Point Unit): Yes, supports floating point operations (single and double precision)

Graphics accelerator: DMA2D (for fast graphics handling)

LCD interface: LTDC + MIPI DSI (for high resolution TFT displays)

External memory controller: FMC (supports SDRAM, NOR, NAND, SRAM, etc.)

**Peripherals**

Timers/PWM: 32 advanced and general timers

ADC: 3x 16-bit ADCs, up to 36 channels

DACs: 2x 12-bit channels

GPIOs: 168 general purpose pins

DMA: 18 DMA channels

**Communication interfaces**

USART/UART: 8 (up to 20 Mbit/s)

I2C: 4

SPI: 6

CAN: 2 (CAN FD 2.0B, better than the standard CAN of the F429ZI)

USB: USB 2.0 OTG FS and HS with integrated PHY for HS

Ethernet: Yes (MAC with IEEE 1588 support, requires external PHY)

SDIO: Yes, SD/MMC compatible

**Security**

Watchdog (IWDG and WWDG)

CRC (Cyclic Redundancy Check)

Hardware Protected Memory (MPU)

Hardware encryption (AES, HASH, TRNG)

**Ideal applications**

Embedded systems with advanced UI and complex graphics

Intensive real-time data processing

Industrial and automation applications with CAN F

Telecommunication and networking systems (Ethernet, USB HS)

Advanced motor control with multiple sensors

## Resume

For this application I’m going to use the [STM32F49ZI](https://www.mouser.mx/ProductDetail/STMicroelectronics/STM32F429ZIT6?qs=DqCdCwOw4%2F5LWpMGYjTR8w%3D%3D) because this is the best fit for the project, the price is not the best per unit, something like 22 dollars, but for the project implications will need the power of processing, also will need to design the graphics and use linux, to run the program.

The expensive components will be the battery, the MCU and the screen, so, this design will need to get another iteration to get a lower price.