What I’ve learned ?

Through this project I discovered rust language & embedded software development. I learned how to compile software on another device (cross-compilation) to upload it on an embedded item.

I implemented 4 main features (Gpio,Usart,SPI & I2C) for two different types of targets (atmega328p & ESP32-S3) using the corresponding datasheet for each target to manipulate registers.

All those features constitute a Hardware abstraction layer (HAL) allowing the user to easily use some features of the MCU (the four implemented).

The first difficulty in this project was being able to compile with right toolchain the code. In fact, there is a different toolchain that we must use if we want to compile the rust project either on the ESP32-S3 or on the atmega328p. The problem was that some toolchains are harder than others to set up (they are categorized as tiers (1 2 or 3), 3 being the hardest & the toolchain for the atmega is tiers 3). After the toolchain set up another obstacle was the linking step which was a nightmare on windows (with all those huge paths) that’s why I moved on Linux to do the project.

After the compiling steps, I needed to update the configuration file, the Cargo.toml in which we must write the “panic = “abort”” parameter in the profile sections since we are targetting a bare-metal environment (without the standard library std). In this same file we also had to precise the lib.rs (for the modules imports) & main.rs (for the tests) files paths without forgetting to add the two features “atmega328p” & “esp32_s3” that will be essential to make our compilation successful.

Once the set-up finished I implemented each of the 4 features for both targets. Each feature has a directory in /src (/src/gpio , /src/usart ...) and in each of those directory there are 3 rust files (two for the implementation of the feature for bith targets & one to export the structures implemented in those files (mod.rs))

The /src/libs.rs files imports all the needed functions & structures defined for the 4 features in order to use them in the main.rs file.

In the main.rs file there are commented tests for all the features & both targets.

To test a feature for a target you just have to decomment it in the right part of the code and then change target if needed.

To change target, you need to change 4 variables in the Makefile and then do “make buildatmega” or “make buildesp”.

I think the most challenging part in this project was being able to read a 1500+ pages long datasheet and find the right sections with the right registers and steps to implement the features. But it was fun.