RUST support on ESP32-----------------------

1. Install RUST support on your computer

**Install Rust**

1. follow the instructions on RUST website

**Install related support**

1. "cargo install cargo-generate"
2. "cargo install ldproxy"

1. Install espup

**Introduction of espup:**

espup is a tool for installing and maintaining the required toolchains for developing applications in Rust for Espressif SoC’s. In theory, it should do all the heavy lifting for you. On Windows this includes installing: Python, Git, Windows10SDK, and a Windows MSVC compiler of the right version. It also sets some environment variables necessary for compiling to ESP32 “xtensa” targets.

**Steps to install**

1. "cargo install espup"
2. "espup install"

Note: espup will create an export file that contains some environment variables required to build projects.

Note: The documentation says “There is no need to execute the file for Windows users. It is only created to show the modified environment variables.” **however this is wrong**.

A screenshot of a computer program

Description automatically generated

1. Create project from template

"cargo generate esp-rs/esp-template"

A screenshot of a computer program

Description automatically generated

 (this is an example to create a bare-metal project, you can create projects with other templates)

1. Build your program

"cargo build"

A screenshot of a computer program

Description automatically generated

1. Deploy your program on ESP32

**Setup the connection with esp32\_s3 dev kit**

1. Connect the UART port of esp32\_s3 dev kit with computer.
2. Install the esp32\_s3 usb port driver – CP210x Universal Windows Driver.
3. Confirm the driver is installed and you can see the device as a COM Port.

**Install espflash**

1. "cargo install espflash"

Note 1: If you are installing espflash from source (ie. using cargo install) then you must have rustc>=1.76.0 installed on your system.)

Note 2: (If you are running **Linux** then [libudev](https://www.freedesktop.org/software/systemd/man/latest/libudev.html) must also be installed; this is available via most popular package managers. If you are running **Windows** or **macOS** you can ignore this step.

#Debian/Ubuntu/etc.apt-get install libudev-dev  
#Fedoradnf install systemd-devel)

**Flash your program to dev kit**

1. "cargo run"

