This is documentation(guide) for setup of rust embedded developement for esp32/esp32 wroom32

For Linux:

> Parent Reference: https://docs.esp-rs.org/book/installation/riscv-and-xtensa.html

Setup Rust:

> Note: In this guide I will target the `no-std` developement and not `std` developement for rust.

- Install rust using the following command: `curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh`
- Setup toolchain for riscV chips: `rustup toolchain install stable --component rust-src`

  then:

  `rustup toolchain install nightly --component rust-src`
- Add Targets:

```
rustup target add riscv32imc-unknown-none-elf # For ESP32-C2 and ESP32-C3
rustup target add riscv32imac-unknown-none-elf # For ESP32-C6 and ESP32-H2
```

- Setup EspUp for Xtensa cores:
  - Install EspUp: `cargo install espup`
  - Setup tools using EspUp: `espup install`
  - Add export-esp to terminal path: `cat $HOME/export-esp.sh >> ~/.bashrc`
- Install LDProxy for std developement: `cargo install ldproxy`
- Install Esp Generate:

  > Helps in generating cargo template for developement! `cargo install esp-generate`
- Install ESP Flash to flash to the Esp32: `cargo install espflash`
- Install necessary developement libs:

  ```
  sudo apt-get update
  sudo apt-get install libudev-dev
  ```
- Create a new Esp rs developement project: `esp-generate --chip=esp32 your-project`

  > Here your-project should be replaced by the name of your project!
- Go to your project dir: `cd your-project`
- Give necessary permission on linux:

  > This will allow you to access the esp32! `sudo usermod -a -G dialout $USER`
- Reboot your pc/laptop!
- Connect your ESP32 using usb!
- Then finally open a terminal in your-project and run: `cargo run --release`