

# Basic Setup of RP2040 Zephyr firmware Build on GitHub

---

Our board is based on an [RPI RP2040](#). It will run as a battery driven, always on sensor monitoring system. The firmware would either run on our custom board or a vanilla RaspberryPi Pico board.

You should already have a RaspberryPi Pico board to test with.

## Objective

Create a GitHub repository that configures a Yocto build, which produces artifacts for the supported boards.

- Use GitHub Actions
- Artifacts can be downloaded from Actions Summary
- Fork of <https://github.com/experimentals/ziloo-firmware> main branch.
- Zephyr RTOS based
- Which compiler chain to use Pico SDK or the regular ARM toolchain?
- Make sure that firmware can be applied using USB UF2(BOOTSEL) and OpenOCD
- Provide a debug console comms USB driver if possible
- Use a source code structure that can combine App and Device Driver source code in built firmware
- Provide working sample UART echo driver in Rust
- Can MicroPython or CircuitPython be compiled in as an optional library?
- Can TensorFlow Lite be compiled in as an optional library?

## Milestones

- \$200 Docker Container config and build of the basic firmware with Artifact generation
- \$100 Add App and Device driver source structure
- \$200 Explore options for linking MicroPython, CircuitPython and TensorFlow

## Existing Information

- [Ziloo Firmware repository with auto build for other CPU\(RV1109\)](#)
- [Pico C/C++ SDK](#)
- [A Rust-based UART echo server for the Raspberry Pi Pico](#)
- [Zephyr RTOS Development in Linux for nRF52](#)
- [MicroPython Pico Port](#)
- [MicroPython Zephyr Port](#)
- [TensorFlow Lite Micro](#)
- [LittleFS 2.5](#)
- [Getting Started with Rust on a Raspberry Pi Pico](#)