ESP32-WROOM-RP Project Planning

Instructions: If you see a research task that you'd like to lead, start by creating a new discussion item under the project's Discussions tab titled the same as the research/design task that you've chosen to work on. Place any documentation, community discussion and general output under your newly created discussion thread to keep as much of our work transparent and available for others to understand and learn from. And finally, add a link below to the discussion.

If you're doing purely a task without any research or documentation, then please create an Issue to describe and track your work.

Research + Design

These are items that need to be researched, designed and documented before we start implementation.

1. `[Discussion + Task]` Create initial source tree layout
2. `[Research]` To workspace or not to workspace at the beginning of this project?
3. `[Research]` Research and document the exact models of WiFi co-processor boards we can support, what are the differences?
4. `[Research]` What major version(s) of the Adafruit (and Arduino) ESP32 WiFi co-processor firmware do we want to support? Document the decision we make and our rationale.
5. `[Discussion + Research]`What other RP2040 microcontroller board do we want to support that compliments the RPi Pico?
6. `[Task]` Put our git pre-commit hook in place for running cargo fmt
7. `[Research]` Research and design our testing strategy:
    - What level of unit testing do we need?
    - What level of integration testing do we need?
    - What types of things do we need to mock?
    - What should be automatically run in CI, with every pushed commit vs nightly?
8. `[Research]` Research and document our use of rustdoc for documenting our crate's public API and example API usage
9. `[Discussion + Task]` Create a PR template in GitHub for this project?
10. `[Discussion + Task]` Create an issue template in GitHub for this project?
11. `[Task]` Create an extremely helpful README at the root of the project tree that details how to:
    - Set up a home development hardware kit
        i. Include detailed instructions for how to purchase, wire, flash and run on both chosen RP2040 series boards
    - Flash and run an example embedded application that uses our crate

A project from the Rust Never Sleeps community

   - Flash and run on-device tests

12. [Discussion + Task] Community discussion and creation of high level system architecture diagram(s)
13. [Research + Discussion] Decide on the core set of WiFi co-processor functionality to support for significant pre-1.0 releases vs a 1.0 stable release
    - Define an explicit semantic versioning scheme (see Rust for Rustaceans chapter 3 on making APIs in Rust)
    - E.g. What defines the v0.1 release vs the v0.2? How many < 1.0 releases will we need before we want to declare it stable?
14. [Discussion] Time-based release schedule or milestone-based?
15. [Discussion] Git workflow
    - Merge vs rebase
16. [Research + Task] Get us ready to be able to publish our crate releases with crates.io
17. [Discussion] How do we raise awareness of our project for use in other people's embedded Rust projects, and call other developers to participate in developing it?

Implementation Milestones + Planning

1. Implement initial unit/integration testing plan from research

Design Philosophy

- Prefer initial functional interface designs
- Each release should deliver features that immediately useful
- Design discussion scratchpad

Release Milestone Planning

Unstable

0.1

- Available on Crates.io
- Join a provided WIFI network (non-enterprise)
- Monitor WIFI network connection status
- Send a valid TCP data stream to remote server
- Support both IP address and host name

- Disconnect from remote TCP server and from the WIFI network

Stable

1.0.0