

Karim Ait-Allaoua

ENSE 481

20 March 2024

## ENSE 481 Project Plan

## Changelog

REV-A: Initial submission

REV-B: Updated to reflect power consumption measurement changes

REV-C: Updated to include freeRTOS considerations, cleanup

## Project Description

I will study and investigate low power design of the `stm32f103` lineup of microcontrollers (MCU's). More and more electronics keep getting smaller and from that the desire to incorporate them on items such as wearables (or any other battery operated device) grows with the passing years. As such the universal agony that is your battery operated device needing a recharge, or simply the battery life not meeting expectations occurs more than it should from my experiences. Coupled with the fact that highly capable (thereby potentially high power consuming (assuming all peripherals used, etc)) 32 bit MCU's are growing in global markets (Alsop, 2022), understanding how to optimize for low power consumption needs to be in the toolkit of all embedded developers. For my application I will implement a cheap yet effective way to measure the power the MCU is taking by the MCU itself.

## Requirements

- I shall configure an Nucleo-F103RB (Nucleo) for programming with its associated ST-LINK module detached for better power consumption measurements
- I shall configure the Nucleo in near maximum power consumption (PC) mode to be used as an upper bound of possible PC
- I shall configure the Nucleo in a “typical” PC mode as a control

- I should configure the Nucleo in a lowest possible while-still-functional PC mode where lowest power is achieved with no built in low power modes used (turn off all clocks as possible, CPU clocks remain on)
- I shall configure the Nucleo in **Sleep-now** and **Stop** modes (AN2629)
  - Implement these as separate CLI modes
- I should configure the Nucleo in **Standby** mode (AN2629)
  - Implement these as separate CLI modes
- I shall measure power consumption in the mW range for all above mentioned modes
- I shall compare and assess the various above mentioned modes qualitatively for their difficulty in implementation and effectiveness
- I should determine the incremental power consumption of all **stm32f103rb** peripherals by subtracting the power with the peripherals on and the power with all peripherals on.
- I shall discuss freeRTOS compatibility with sleep modes
- I should implement tickless freeRTOS sleep
- ~~● I should use an existing project or code and determine its current power consumption, and attempt to use methods to minimize its power consumption, then measure the power.~~
  - The above has been clarified to be a power measurement system with a shunt resistor.
- ~~● I should calculate minimum theoretical power consumption for all above mentioned modes~~
- ~~● I should investigate other algorithms for low PC that do not use built in low power modes~~

## Scenario

Since this is more of a research and investigation, a scenario of a user interacting with my project is not necessarily applicable. However, all source code and project results will be available online publicly for reading. As such, the only possible scenario is a user could attempt to replicate my results via cloning, building, flashing and measuring the various PC tests.

## Functional Components

1. Computer with serial terminal: a laptop with putty or similar serial to usb program to send serial commands to nucleo
2. Power measurement device: ~~TBD how I will measure this, Doug mentioned he had one that may work using Doug's Nordic Power Profiler 2~~ using a Joulescope JS-110.
3. Nucleo: Software subsystems
  - CLI: allow user to choose which specific test to run
  - Test functions: called when the chosen test is to be run. Will involve use of interrupts, either external GPIO or communications such as I2C to wake in case of built in sleep modes.

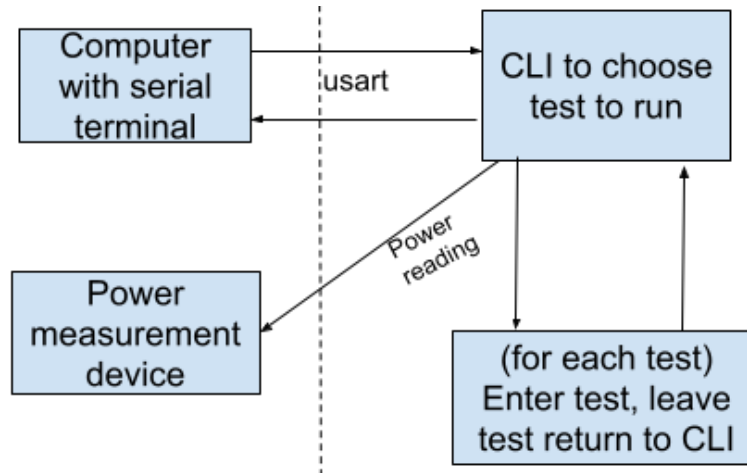


Figure 1: Block Diagram of System

## References

Alsop, T. (2022). *Global microcontroller market size by type 2021*. Statista.

<https://www.statista.com/statistics/553426/worldwide-microcontroller-unit-market-size/>