Software System Review

Joesphan Lu

**Summary**

The software of the insulin cooler is responsible for controlling the pettier coolers, battery balancing, datalogging, and BLE communications to a phone app.

The phone app is responsible for presenting the user with battery percentage and temperature data.

**Tools**

Visual Studio code with the ESP-IDF SDK will be used for coding, debugging, and flash programming. Gitlab and Github desktop will be used for revision control.

A generic usb to UART converter will be used as the programmer. CH340G, FT232RL or similar

**MCU Resources**

A screenshot of a computer screen

Description automatically generated with medium confidence

On-chip peripherals:

Bluetooth/BLE, PWM, ADC, Radio, RTC/Low Power Management Subsystem

**Kernel and Task/ISR design**

freeRTOS will be used, as the BLE SDKs by EspressIf use it. The BLE is the biggest software design challenge in this project.

ISRs:

BLE device connection detection, USB power availability, temperature control timer interrupt

Time-Slices TBD

**Inter-Task Communications**

Diagram

Description automatically generated

**Module/Library Description**

TemperatureControlModule:

Controls temperature of insulin, measures and controls peltier

BatteryBalancer:

Reads the cell voltage and drains off cells that have excess voltage

BatteryMonitor:

Reads the battery percentage

Datalog:

Records the temperature of insulin, later to be sent via BLE to the mobile device