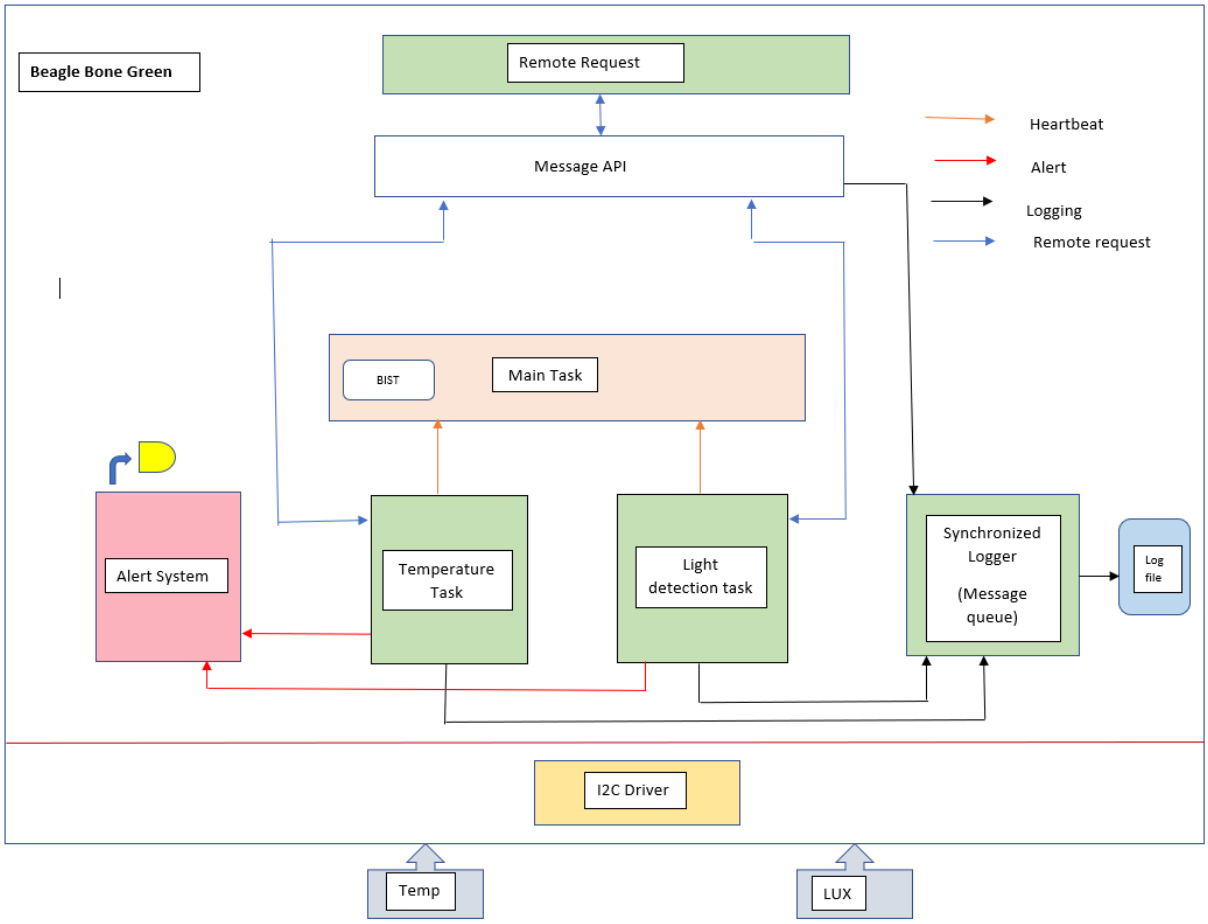
Refrigerator Monitoring System

Developers:

Steve Antony Xavier Kennedy

Sanju Prakash Kannioth

Software Architecture Diagram:



BIST:

Start up tests to confirm that the devices are functioning as expected. Write functions to check I2C hardware functionality, liveliness of threads and log error messages for failures.

Main task:

Handles the spawning of child tasks, application clean up on exit and heartbeat. Ping request to the threads to check its heartbeat at periodic intervals using POSIX Timer API.

Temperature sensor task:

This task takes care of the temperature sensor measurements on a timed basis. We plan to implement a POSIX timer for this purpose. Other duties of this task include logging temperature data and responding to the remote request task using message API. We plan to use a socket to facilitate this communication.

Light sensor task:

The duty of this task is to monitor the amount of light on a timed basis. We plan to use a POSIX timer for this. If there is a change in light, we log it using the synchronized logger. Other duties of this task include logging LUX data and responding to the remote request task using message API.

Synchronized logger:

This is a thread spawn and is used to temporarily store all the incoming data to be logged in an IPC message queue. The other end of the message queue is responsible for reading data and actually storing data into a log file.

Remote request task:

This task uses the message APIs to communicate with the different tasks that are running. We plan to use sockets to facilitate this communication.

Message APIs:

It uses POSIX queue to faciliate the communication between remote request task thread and other threads in the system. We plan to use user defined API calls such as getTempData(), getLuxData(), LogData(), Alert(), StartUpTest().

I2C Driver:

We plan to use the kernel libraries available for I2C to initialize and get data from the sensors.

Alert system:

Failures or errors are sent to this system from the different tasks and this systems performs predefined actions for the error conditions.

Unit tests:

The unit tests will be written to test functionality using Cunit.