

Refrigerator Monitoring System

Mutli-thread application on Beagle Bone

Developers

Steve Antony Xavier Kennedy

Sanju Prakash Kannioth

University of Colorado Boulder



Contents

Introduction:	3
Hardware used:.....	3
Software Architecture Diagram:	3
Threads:	4
Logger Thread:	4
Temperature Task:	5
Light Task	6
Remote Request Task	7
Heart Beat	7
Power On Self-Test (POST).....	8
Unit Tests	8
Future Scope of the Project	8

Introduction:

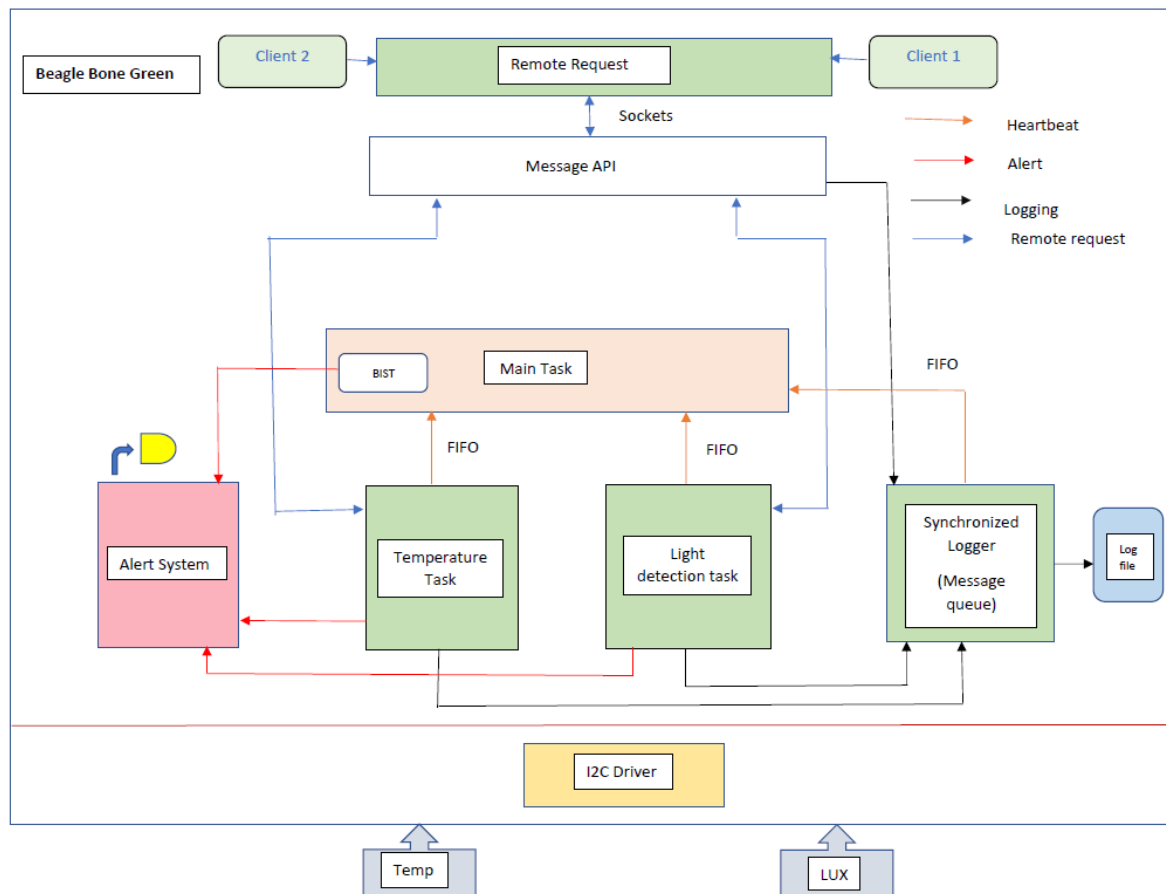
At times, the door of the fridge may be left opened and unnoticed. This project has been developed to alert the consumer if the fridge door is left opened or if the fridge temperature is not maintained within the limits.

This project helps to **reduce energy consumption** if the door is opened for long time and thereby reducing the emission of CFC and thereby **reducing global warming**

Hardware used:

- Beagle Bone Green
- Temperature sensor – TMP 102
- Light sensor – APDS-9301

Software Architecture Diagram:



Threads:

Logger Thread:

The logger task has been implemented as that of Python Logging. The logger levels available **DEBUG, INFO, WARN, ERROR** (increasing order of severity).

The log file name and folder name are passed as command line arguments with the executable.

- **Set Level**

Sets the threshold for this logger to *level*. Logging messages which are less severe than *level* will be ignored; logging messages which have severity *level* or higher will be emitted

- **IPC**

All tasks in the application communicate with the logger task using message queues.

- **Log items**

Log info consists of the following

- Log level
- Source Thread id, function name
- Time stamp

- **Synchronized logger**

Mutex log has been implemented when the logger is invoked to make the log emits in a synchronized manner

- **Signal Handler**

- The logger task can be killed from the main task using the signal SIGALRM
- Killing it using SIGALRM will clean up all the resources allocated in the logger task and the task will be gracefully exited

- **Heartbeat**

The logger task sends heartbeats to the main tasks to inform that it is alive. The heartbeat is sent at regular intervals with the help of POSIX timer

Temperature Task:

The temperature task is spawned from the main task. This task does the work of measuring the temperature from the sensor registers. It uses i2c for communication with the sensor.

- **Heartbeat**

The temperature task sends heartbeats to the main tasks to inform that it is alive. The heartbeat is sent at regular intervals with the help of POSIX timer.

- **Interrupt driven alert system**

- An interrupt is trigger on the hardware pin of the sensor when the temperature range is outside the set threshold high and low temperature. This helps to maintain the refrigerator temperature within the limits thereby preserving the food when the door is left opened.
- The lower threshold is set as 27C and higher threshold is set as 29C. This default configuration can be overridden.
- The interrupt hardware pin is given as input to the BB where it is polled for any change in state.
- If there occurs a change in state of the interrupt pin, an alert system is triggered when glows LED2 on the BBG

- **Signal Handler**

- The temperature task can be killed from the main task using the signal SIGUSR1
- Killing it using SIGUSR1 will clean up all the resources allocated in the temperature task and the task will be gracefully exited

- **Behavior when temperature sensor fails after booting**

- The temperature thread does not die when the temperature sensor fails after successful reboot.

- It resumes temperature measurement function when it is back to form i.e when the sensor is reconnected
- LED0 indicates an error on light sensor
- **Remote Request Task**
 - Further, the customer can know the fridge temperature from anywhere in the world. It serves data through remote request task which uses sockets

Light Task

The light task is spawned from the main task. This task does the work of measuring the brightness inside the refrigerator from the sensor registers. It uses i2c for communication with the sensor.

- **Heartbeat**
 - The light task sends heartbeats to the main tasks to inform that it is alive. The heartbeat is sent at regular intervals with the help of POSIX timer.
- **Alert system**
 - The threshold value for determining whether the fridge door is open or closed is set as 70 lux. This threshold is configurable
 - When the fridge door changes state from open to close or close to open, it sends notification
 - LED3 is used to indicate if the door is open
- **Remote Request Task**
 - Further, the customer can know the state of the fridge door from anywhere in the world. It serves data through remote request task which uses sockets
- **Signal Handler**
 - The light task can be killed from the main task using the signal SIGUSR2
 - Killing it using SIGUSR2 will clean up all the resources allocated in the light task and the task will be gracefully exited
- **Behavior when light sensor fails after booting**
 - The light thread does not die when the light sensor fails after successful reboot.

- It resumes lux measurement function when it is back to form i.e when the sensor is reconnected
- LED1 indicates an error on light sensor
- **Lux value measurement**
 - Lux value is obtained from CH0 and CH1 channels from the sensors.
 - The procedure to obtain lux value is performed as per the data sheet

Remote Request Task

The remote request task is spawned from the main task. This task does the work for serving the remote client requests

- **Simultaneous Multi client support**
 - Each client request to the server gets is forked as a new process and hence can support multiple clients simultaneously
 - This facility any number of user to know the state of the fridge from anywhere in the world
- **IPC**
 - Sockets are used for the remote clients to communicate with the server to get the state of the fridge
 - The transport layer used here is TCP due to its reliability
- **Client services supported**
 - Temperature in
 - Kelvin
 - Celsius
 - Fahrenheit
 - Brightness in lux
 - The state of fridge door

Heart Beat

Heart beat is the main task of the application. All tasks send its status repeatedly to the heartbeat task. The heartbeat task has a POSIX timer which checks the heartbeat of the tasks at regular intervals.

It used **non-blocking named pipes** for the tasks to send heartbeat to the main task

Power On Self-Test (POST)

This is performed at the time of booting to ensure all the connected sensors are working properly. If any sensor fails on POST, that particular **thread** is **killed**, and the remaining application works as expected. It is ensured that the non-operation of any sensor doesn't affect the other operations of the application. The status of the POST is logged on the logger file.

Further, an LED glows to indicate that sensor is non-operational

- LED0 indicated that startup failed for temperature sensor
- LED1 indicated that startup failed for temperature sensor

Unit Tests

Unit testing has been performed for

- Lux sensor module
- Temperature sensor module
- Logger module

It checks the read and write functionality of all registers in the temperature and light sensors

Future Scope of the Project

- The project can be extended to send messages to phone using Twillio if the fridge door is opened for long time.
- Automatic closure of door can be implemented if it is opened for long time. This helps to save energy and reduce the emission of CFC to the atmosphere