Advanced Practical Embedded Software Development

Proposal for Project 2

6th April, 2018

Team Members:

Pavan Dhareshwar

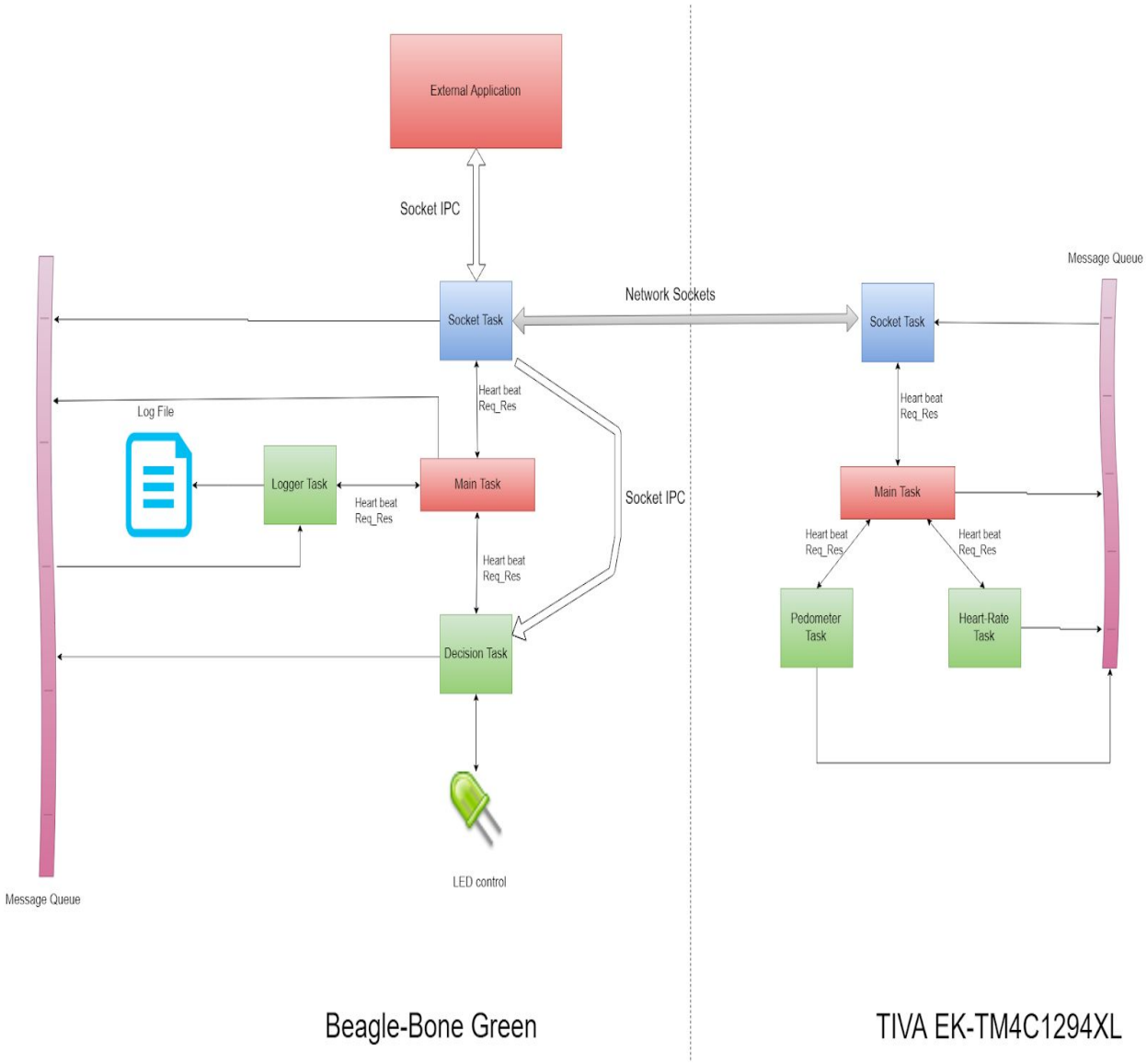
Sridhar Pavitrapu

Github Repository: https://github.com/pavandhareshwar/APES_Project_2

Description:

The overall system that will be designed will have the Tiva and BBG counterparts interacting via network sockets and using the sensor data to monitor health statistics. The BBG will serve as a remote logging server, logging numerous events and information to a file. The BBG will have a decision module, that will perform some action based on sensor data of the respective client. The Tiva will act as a sensor hub, with separate tasks to manage the sensors used. The sensor data retrieved and the sensor requests will be communicated over network sockets.

Software Architecture Diagram:



API description and functionality:

We will have the following tasks functioning on BBG and Tiva to achieve the goal described above.

Tiva:

1. Pedometer Task: This task is primarily responsible for interacting with the IMU sensor using I2C protocol to retrieve step count information and also configure the sensor.
2. Heart Rate Task: This task is task primarily responsible for interacting with the heart rate sensor using I2C protocol to retrieve heart rate information and also configure the sensor.
3. Socket Task: This task handles the functionality of sending sensor data from Tiva to BBG and receiving requests from BBG.
4. Main Task: This task handles the creation of all the sub-tasks and checking of periodic heartbeats of the sub-tasks.

BeagleBone Green:

1. Socket Task: A user-level task to handle receiving of sensor data from Tiva as well as sending requests to Tiva.
2. Logger Task: A user-level task responsible to log sensor data and event occurrence information to a log file. The log file attributes will be configurable via a config file (similar to the one used in project 1).
3. Decision Task: A user-level task to notify the user about the current state of the system and perform some action based on the state like LED blink.
4. Main Task: A user-level task that handles the creation of all the sub-tasks and checking of periodic heartbeats of the sub-tasks.

In both BBG and the Tiva, once the main task has spawned all the sub-tasks, it will sleep and periodically wake up to request the status of each sub-tasks (the heart-beat) to ensure that all are up and running. If one or more tasks don't respond with an 'alive' response (no heartbeat), the main task will interpret this task(s) as not-running anymore and log this information. The request-response messaging scheme between the main thread and the sub-tasks will be named and referred to by 'HB_req_rsp' until a definitive IPC scheme is finalized.

(If needed) We are planning to establish communication between the main tasks on both BBG and Tiva to communicate heartbeat information of BBG and Tiva. If one of the tasks, other than the socket task on either BBG or Tiva isn't alive anymore, the main task of the system with one or more unalive tasks will communicate this information to the main task of the other system. If the socket task isn't alive, then a timeout mechanism will be used to detect an unalive scenario.

Pedometer Task:

As mentioned in the description above, the pedometer task will be responsible to interact with the IMU sensor using I2C protocol and perform read and write operations to configure the sensor registers and get the sensor data.

Pedometer task will have the following set of API's:

int pedometer_task_initialize():

This API will perform all the initializations necessary to access the pedometer sensor.

int get_step_count():

This API will be used to retrieve the step count information from the IMU sensor.

Heart Rate Task:

As mentioned in the description above, the heart rate task will be responsible to interact with the heart rate sensor using I2C protocol and perform read and write operations to configure the sensor registers and get the sensor data.

Heart Rate task will have the following set of API's:

int heart_rate_task_initialize():

This API will perform all the initializations necessary to access the heart rate sensor.

int get_heart_rate_data():

This API will be used read the heart rate data from the heart rate sensor.

Socket Task:

The socket task will be responsible for communicating sensor data and sensor configuration requests between BBG and Tiva. The communication is done using BSD sockets. The sensor data communicated is packed into a structure containing the sensor data, log_type, log_level and source id.

We will use the following enumerations and structures to pack the sensor data:

/* Enumeration for log levels */

typedef enum {

LOG_LEVEL_INFO,

LOG_LEVEL_STARTUP,

LOG_LEVEL_SHUTDOWN,

LOG_LEVEL_CRITICAL

```
}e_log_level;
/* Enumeration for log type */
typedef enum{
        LOG_TYPE_DATA,
        LOG_TYPE_ERROR,
        LOG_TYPE_REQUEST,
        LOG_TYPE_RESPONSE
} e_log_type;
/* Enumeration for message source */
typedef enum {
        TASK_PEDOMETER,
        TASK_HEART_RATE
}e_source_id;
/* Socket message structure */
typedef struct {
        e_log_level log_level;
        e_log_type log_type;
        e_source_id src_id;
        char data[MAX_MSG_LEN];
}  x_sock_msg;
```

Decision Task:

This task is responsible for receiving messages from the socket task on BBG using a message queue and checking for different functionalities like if Tiva is alive/not and if any sensor task isn't alive. In case of any failure mechanism, LED(s) can be glown accordingly.