Team - Project 2

Created: Nov 21, 2017 | Updated: Dec 12, 2017

Project Leader

Alexander Fosdick	Dec 12, 2017
NAME	**DATE**

Project team member

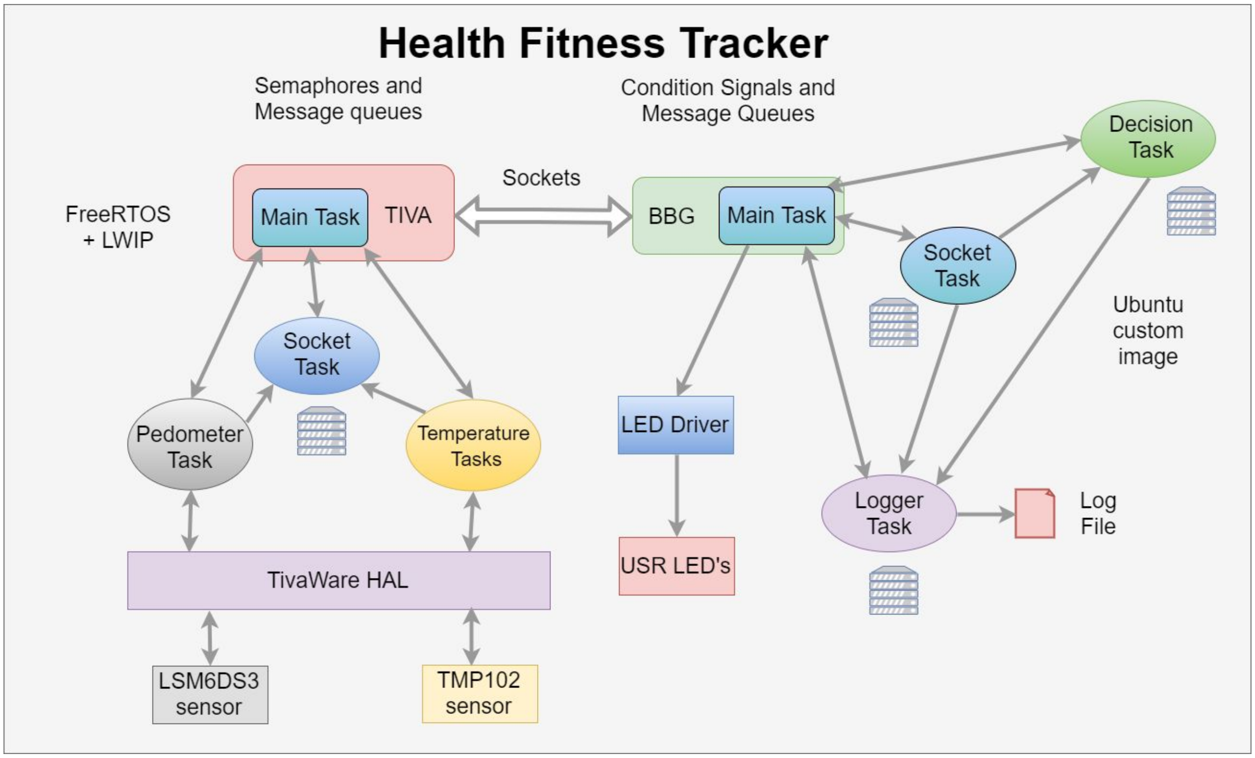
Vihanga Bare	Dec 12, 2017
NAME	**DATE**

Project team member

Virag Gada	Dec 12, 2017
NAME	**DATE**

Topic: **Health Data logging and Storage system**

Software Architecture:



Components:

1. BeagleBone Green
2. Linux OS on BBG
3. TIVA Board
4. Free RTOS on TIVA
5. TIVA ware HAL library
6. Temperature Sensor (Sparkfun TMP 106)
7. Pedometer (Sparkfun LSM6DS3)
8. LwIP Sockets API
9. BSD Sockets API

Description: Health Monitoring application (like FitBit) which tracks footsteps and body temperature and sends this data using socket communication to log the data.

Tasks list on the TIVA Launchpad which runs on FREE RTOS:

- Pedometer task (Sparkfun LSM6DS3) retrieves data from pedometer sensor (I2C communication). A timer overflow event will send signal to the task using semaphore and cause the task to run and retrieve data.
- Temperature task (Sparkfun TMP106) retrieves data from temperature sensor (I2C communication). A timer overflow event will send a signal to the task using semaphore cause the task to run and retrieve data.
- Socket task to send above data values over TCP sockets which use Free RTOS TCP API and LWIP sockets library.
- Main task to create above tasks.
- Heartbeat task to monitor heartbeats from above task using semaphore signalling. If heartbeat not received this task kills above tasks.

Task on the BeagleBone Green which runs Linux OS:

- Socket task to receive data over socket connections created using BSD sockets API.
- Logger task to log this data into log file depending on the log levels and request. We have also used colors in the log file to determine log levels.
- Decision task is used to analyze the data sent by the sensors and give a notification to the user based on the values. Decision task also taks care of shutting down the system. If the log packet received from TIVA has a system shutdown message, this shuts down the whole system.
- Main task create above tasks and to monitor them.

Functionality:

Userspace Functions:

- Sockets API for TCP sockets
- TIVA ware HAL library for interfacing with all sensors
- I2C libraries to define our own I2C communication API
- Pthreads API(mutex, condition variables) to synchronize communication between multiple thread tasks

Kernel Functions:

- Socket system calls (open, close, read, write)
- Semaphores to synchronize any kernel module data
- Led Character Driver to set led on or off using loadable kernel modules

Data structures overview:

```
typedef enum loglevel
{
        ALERT,
        WARNING,
        INITIALIZATION,
        INFO,
}LogLevel;

typedef enum{
         MAIN_TASK,
         TEMP_TASK,
         PEDO_TASK,
         SOCKET_TASK,
         LOGGER_TASK,
         DECISION_TASK
        }Sources;
```

```
typedef enum{
        LOG_DATA,
        HEARTBEAT,
        DECIDE,
        SYSTEM_SHUTDOWN
}reqCmds;

typedef struct logger
{
  uint8_t sourceId;
  uint8_t requestID;
  uint8_t level;
  float data;
  char timestamp[32];
  char payload[100];
}LogMsg;
```

Functions or API:

BeagleBone:

```
void initialize_queue(char * qName, mqd_t *msgHandle);
mq_send (queue,(const char*)&loggerstruct, sizeof(LogMsg), 1);
mq_receive (queue,(const char*)&loggerstruct, sizeof(LogMsg), 1);
void create_interval_timer(float timer_val);
void sighandler_sigint(int signum);

void *SocketThread(void *);
void *LoggerThread(void *);
void *DecisionThread(void *args);
write_to_driver();
```

TIVA:

```
void setupLSM6DS3();
void setupI2C2();
void setupTMP102();
void readTMP102(double *digitalTemp);
void readStepCount(uint16_t *stepCount);
void temperatureTask(void *pvParameters);
void pedometerTask(void *pvParameters);
void loggerTask(void *pvParameters);
void socketTask(void *pvParameters);
void vTimerCallBack(void *);
echo_init();
lwIPInit(g_ui32SysClock, pui8MACArray, 0, 0, 0, IPADDR_USE_DHCP);
tcp_write(tpcb,&logmsg,sizeof(LogMsg),1);
tcp_output(tpcb);
```