

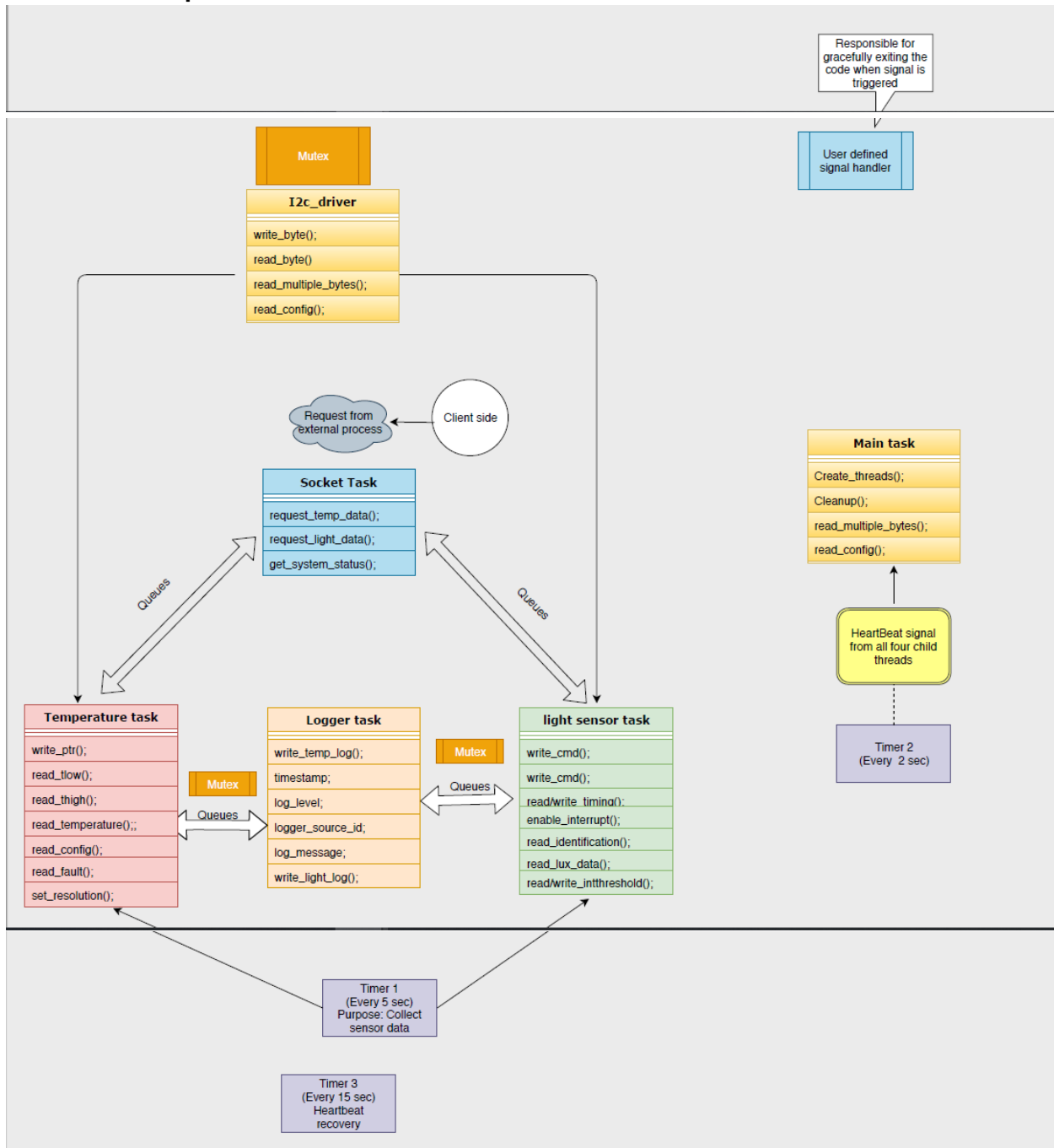
# Advanced Embedded Software Development

Project 1

Tanmay Chaturvedi, Vikrant Waje

Date: 31 March, 2019

**Architecture: Updated**



## **Functions and Threads:**

### **Main thread:**

The main function is responsible for creating four threads.

It is also responsible to monitor each of its thread for the heartbeat and take necessary action if it does not receive heartbeat from its thread.

Function call needed to be implemented are:

Create\_thread(): Create four child threads.

Receive\_heartbeat(): The purpose of this is to read the heartbeat message sent by the child thread using IPC mechanism.

Join\_thread(): Wait for completion of execution of child threads and then exit to cause proper deallocation of resources used.

Log\_error\_console(): The purpose of this is to log any errors or any status onto the console.

### **Temperature sensor thread:**

The purpose of this thread is to establish the communication with temperature sensor using I2C protocol.

It includes:

Write\_ptr(): write the pointer register in the temperature sensor.

Read\_tlow(): Read the contents of Tlow register in the temperature sensor.

Read\_thigh(): Read the contents of Thigh register in temperature sensor

Read\_temperature(): Read the value of temperature.

Read\_config(): reads the configuration of temperature sensor.

Read\_fault(): Read three fault bits.

Set\_resolution(): sets the resolution of temperature sensor

Heartbeat\_response(): send the heartbeat signal to main thread.

### **Light sensor thread**

The purpose of this thread is to establish communication with light sensor using I2C protocol.

Write\_cmd(): Write value to command register

Read\_control(): read value of control register.

Write\_control(): Write value to control register.

Read\_timing(): Read value from timing register.

Write\_timing(): Write value to timing register.

Enable\_interrupt(): Enable the interrupts specific to light sensor.

Read\_identification(): read the identification value of light sensor.

Read\_lux\_data(): read the light intensity value.

Read\_intthreshold(): read the interrupt threshold.

Write\_intthreshold(): Write a value to interrupt threshold register.

Send\_heartbeat(): Sends heartbeat to main thread.

### **Logger thread**

This function receives log from various threads through IPC mechanism. The logs are then written to a log file protected by mutex.

Write\_temp\_log(): Used to write value of temperature logs into the logger file.

Write\_light\_log(): Used to write the value of light sensor into the logger file.

Heartbeat\_send(): sends the value of heartbeat to main thread.

### **Socket task:**

The purpose of this task is to act as a unix socket server which will listen for request from client process residing in external host machine. The host machine will request for temperature and light sensor values The socket task will then use IPC mechanism to get temperature and light sensor value from temperature thread and light sensor thread.

### **I2C device driver**

The device driver is used in user space. No kernel module is planned to be implemented.

BIST Test:

Our BIST test is intensive and tests all the functions implemented in our system.

> Temperature Sensor:

1. Write to THigh Register
2. Read from Thigh Register
3. Write to TLow Register
4. Read from TLow Register
5. Config Temperature Fault
6. Read Temperature Fault
7. Config Temperature EM mode
8. Read Temperature EM off
9. Config Temperature Conversion Rate
10. Read Temperature Conversion Rate
11. Read Temperature Sensor Resolution

> Light Sensor Test

1. Light Sensor Power On
2. Read Identification
3. Set Integration
4. Get Integration
5. Set Gain
6. Get Gain
7. Set Config Interrupt Control
8. Get Config Interrupt Control
9. Set High Threshold
10. Get High Threshold
11. Set Low Threshold
12. Get Low Threshold

>Thread Test

1. Pthread Create

Unittest Test Suites:

1. Pthread Create and Join Test
2. Message Queue Test
3. Temperature Conversion Test