



Engineering & Applied Science

UNIVERSITY OF COLORADO **BOULDER**

# **Advance Embedded Software Development**

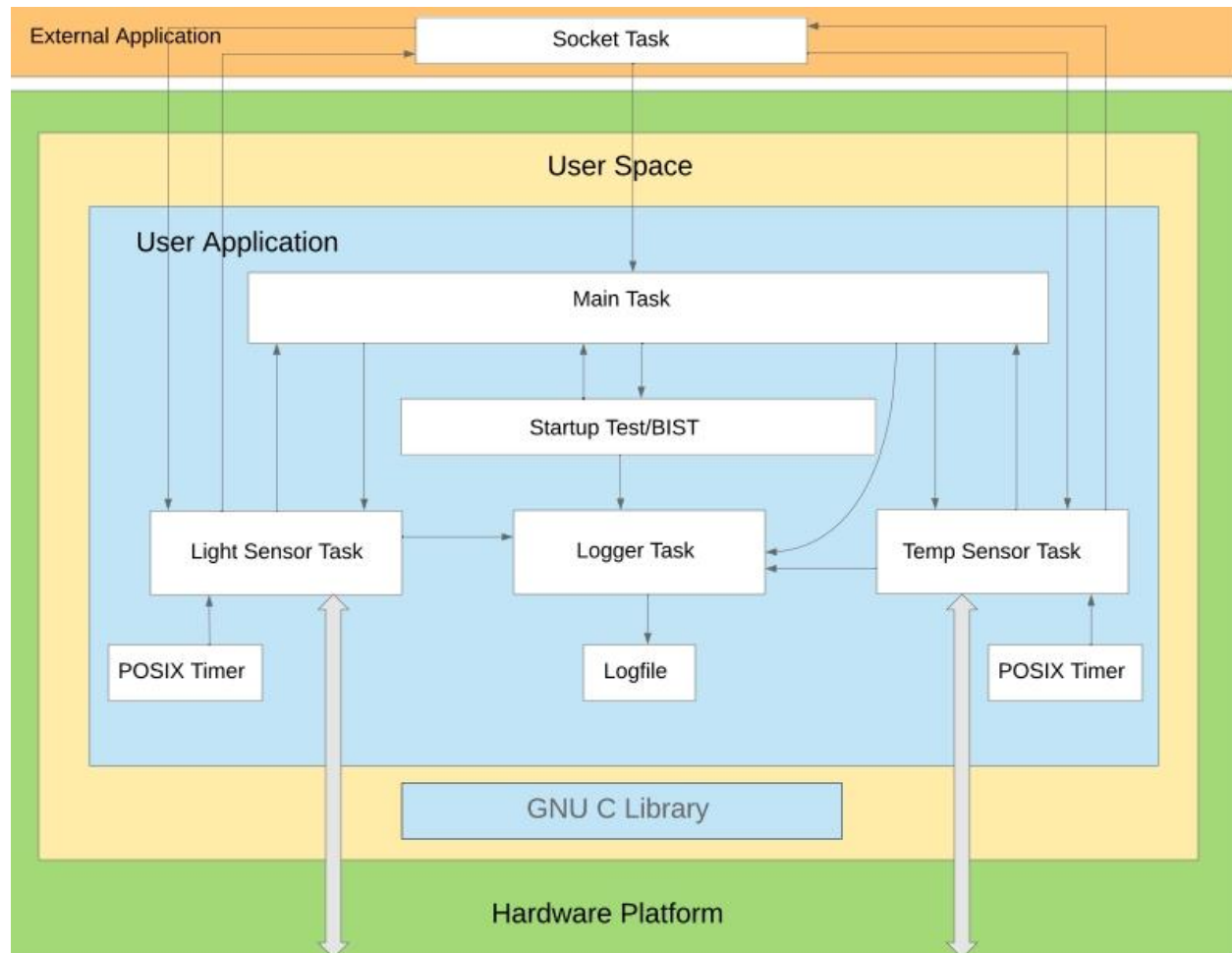
## **Project #1**

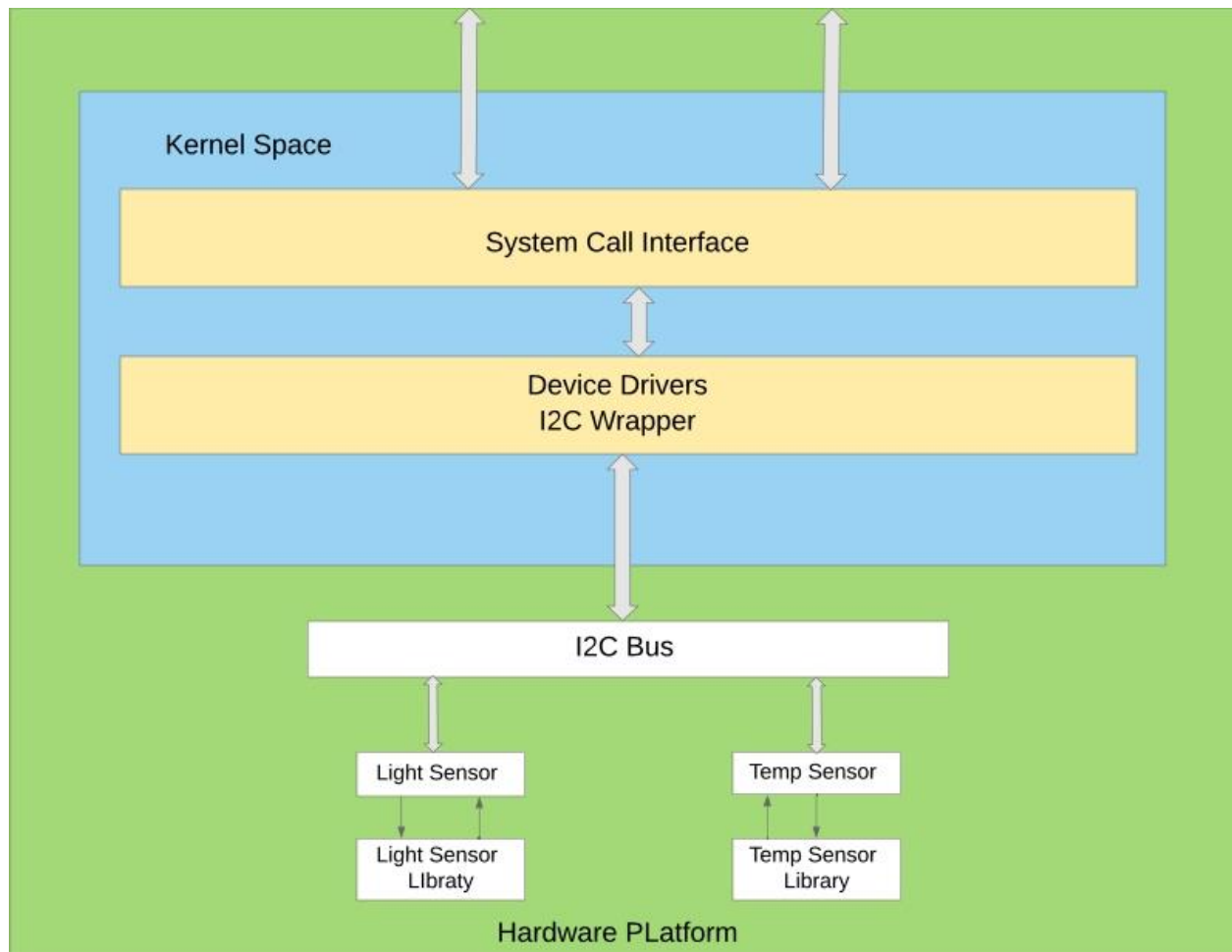
Design Document

Hardik Senjaliya

03/01/2019

# Software Architecture Diagram





*Please consider both the images above are connected with each other which makes a single hardware platform having user space and kernel space interacting with each other.*

# **Tasks and their responsibilities**

- **Main Task**

The main task is responsible for spawning child tasks for temperature sensor, light sensor and socket-client task. This task also ensures all tasks are alive and running periodically using the 'heartbeat' mechanism. The main task waits until the child tasks completes their functionalities and cleans used resources and exits the application. This task is also responsible for performing startup check to ensure all the software and hardware modules are working as required and if not then logs error into the log file and indicates the error using console or user LEDs on the development board.

- **Light Sensor Task**

The primary responsibility of the light sensor task is to collect data from the APDS-9301 light sensor which is capable of sensing visible and infrared lights. The task is interfaced with I2C driver which resides inside the kernel. This task runs periodically and collects data of both type of lights and depending upon the threshold values set by the user, the task logs the change in the state to the logfile. Moreover, it also accepts requests from the other tasks requesting light data and current state of the, dark or light.

- **Temperature sensor task**

The primary responsibility of the temperature sensor task is to collect data from the TMP102 temperature sensor and converts the collected binary temperature values into equivalent celsius, kelvin and fahrenheit values. The task is interfaced with I2C driver which resides inside the kernel. This task runs periodically and collects temperature data on a regular interval. Moreover, it also accepts requests from the other tasks requesting temperature data.

- **Synchronized Logger Task**

The primary responsibility of this task is to accepts log messages from the main task, light sensor task, and temperature sensor task. The logger task accepts name and path of file using the command line. If the provided file name matches

the file name presents on the given path then the task either deletes or creates a back of the old log file. Moreover, when the process ends or forced closed then this task closes file handles after writing buffered log messages into the file. Each log entry in the file contains Timestamp, Log level, Logger Source ID, and Log message. It prints numerical data in float/integer values with string values.

- **Remote Request Socket Task**

The primary responsibility of this task is to request data from the Light sensor and temperature sensor task. The required data is collected by establishing a socket connection between the host application and the remote application. The main purpose of this task to test the application when it is running.

# **A first-cut definition of the needed API/functions for tasks and major functions**

- **APIs for Main task**

- 1) `startup_test()`

- This API is used to perform the startup test to ensure all the software and hardware modules are working as required.

- 2) `heartbeat_test()`

- This API is used to monitor each child task's current running status and it returns error in case of unexpected behavior.

- **APIs for Light Sensor Task**

- 1) `light_sensor_setup()`

- This API is used to initialize the light sensor to perform required functionality.

- 2) `read_lux()`

- This API is used to read Lux data through ADC and converts the value into actual luminosity of the sensor.

- 3) `read_visible_light()`

- This API is used to convert/get the value of the visible light from the collected ADC data.

- 4) `read_infrared_light()`

- This API is used to convert/get the value of the infrared light from the collected ADC data.

- **APIs for Temperature sensor task**

- 1) `temp_sensor_setup()`

- This API is used to initialize the temperature sensor to perform required functionality.

- 2) `read_temperature()`

- This API is used to read the temperature data and converts the data into decimal format.

- **APIs for Logger task**

- 1) `read_logs()`

This API is used to read log messages from Main task, light sensor task and temperature sensor task.

2) write\_logs()

This API is used to write the collected log messages into the log file specified by the user at the user specified path.

- **Device Drivers for I2C/Wrapper Function for the I2C**

1) I2C\_write()

This API is used to write the values into the registers to setup the sensor's functionality.

2) I2C\_read()

This API is used to read the sensor values.

- **APIs for socket task**

1) request\_data()

This API is used to request and collect the sensor data from the light sensor task and temperature sensor task.

# **User Space and Kernel Space software**

- **User Space Software**

The main task, light sensor task, temperature sensor task, logger task, and socket will be developed under the user space.

- **Kernel Space Software**

The I2C device driver and read-write wrapper APIs will be developed under the kernel space.

## **Startup Tests and Unit Tests**

- **Startup Tests**

The main task will have some tests running at the startup to ensure the hardware and software in working order. The following tests will be performed,

- 1) communication with temperature sensor to ensure the I2C hardware and sensor is working.
- 2) Communication with light sensor to ensure I2C hardware and sensor is working.
- 3) Communication with all the threads to ensure all threads have started and are in running status.
- 4) Writing log messages to ensure user specified logs are being written to user specified filename at the specified path.

- **Unit Tests**

Following unit tests will be performed to make sure the application is working as expected.

- 1) Inter-thread communication
- 2) Logger
- 3) Temperature sensor conversions with mocked data
- 4) Light sensor conversions with mocked data.