Engineering & Applied Science
UNIVERSITY OF COLORADO **BOULDER**

Advance Embedded Software Development

Project #1

Report

Hardik Senjaliya
03/31/2019

Click Here for GitHub Repo

Tasks and how they are implemented?

Main Task:

- Creates four child threads, temperature sensor thread, light sensor thread, logger thread and socket thread.
- Creates five posix message queues
    1. qMain - responsible for reading heartbeat responses from all child threads
    2. qLight - responsible for receiving heartbeat and socket requests
    3. qTemp - responsible for receiving heartbeat and socket requests
    4. qLogger - responsible for receiving heartbeat request and log messages
    5. qSocket - responsible for receiving socket request responses
    - qMain and qSocket are BLOCKING queues while all other are NON_BLOCKING queues.

- Heartbeat Functionality:
  Main task requests heartbeat status from each thread periodically every 5 sec. After receiving the request all the thread sends required response.
  If it fails to receive a heartbeat response it sends EXIT command to the threads to exit the application.
- Startup Test:
  Main task requests startup test after spawning required threads. Light sensor threads reads the value of id_register to check I2C functionality while Temperature sensor thread writes value to the tlow register and reads the same value. Logger and Socket tasks just sends the response back as required.

Light Task:

- Light task runs every 25ms and reads the lux value. After reading lux value it checks for the state of the light and accordingly logs the message into the log file by sending a message to logger queue.
- It also responds to the heartbeat and socket requests from the main thread and socket thread.
- After initialization, the task runs into the while loop waiting for the sem_post from the timer handler and starts execution after receiving the semaphore.

- Inside while loop, it runs in a state machine and handles current state based on the requests received and sends response back to the requester.

Temperature Task:

- Temperature task runs every 10ms and reads the temperature value. After reading the value it logs the message into the log file by sending a message to logger queue.
- It also responds to the heartbeat and socket requests from the main thread and socket thread.
- After initialization, the task runs into the while loop waiting for the sem_post from the timer handler and starts execution after receiving the semaphore.
- Inside while loop, it runs in a state machine and handles current state based on the requests received and sends response back to the requester.

Logger Task:

- Logger task receives log messages from all the queues and writes the message into the logfile.
- It creates a new file with the user given command line path and name or takes a back up if a file is already presents.
- It responses to the heartbeat request whenever requested and exit command if requested.

Socket Task:

- It reads the requests from the remote client and sends the data as requested and again starts listening for new requests.

Startup Test:

- Temp Task: this task writes to the tlow temperature register and reads the same register again to check the written data. Thus ensuring I2c is working properly.
- Light Task: this task reads the id register to check functionality of the i2c.
- Logger task and socket task: this task responds to the main task when requested for the startup test thus ensuring they are running properly.

Main tasks logs message into the file after completion of each task's startup test indicating success or failure.

Unit Test:

Light task: for light task unit test checks for below mentioned functions

- Day_or_night
- Low_interrupt_threshold_register
- high _interrupt_threshold_register
- lux_conversion

Temp Task: for this task unit test checks for below mentioned functions

- binary_to_decimal _temp conversion
- Config_register
- Convert_to_farenheit
- Convert_to_kelvin
- Thigh_register
- Tlow_register

All functions and their descriptions are on the GitHub repo in the Doxygen Report.