**GPS and CO-Sensor Integration using FreeRTOS and Google IOT**

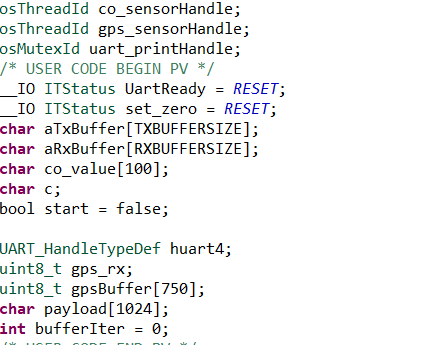
Dates worked: 06/25 (8 hours), 06/26 (8 hours), 06/27 (8 hours), 06/28 (8 hours)

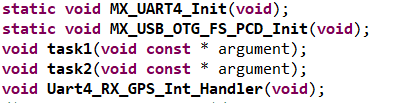
Summary of work:

1. Now that CO-Sensor started working fine over UART2, it was necessary to integrate the GPS with CO-Sensor in the same discovery kit.
2. First approach was to make both the sensors work using FreeRTOS using 2 separate tasks. Within the code of UART2, we need to add dependencies of UART4 for proper functioning of GPS. Following modifications were made in the code,

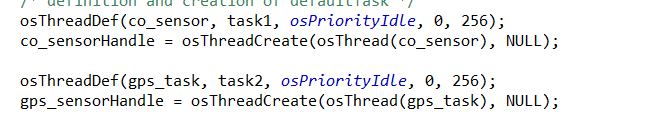
In main.c

1. Added the variables and function definitions specific to GPS and UART4

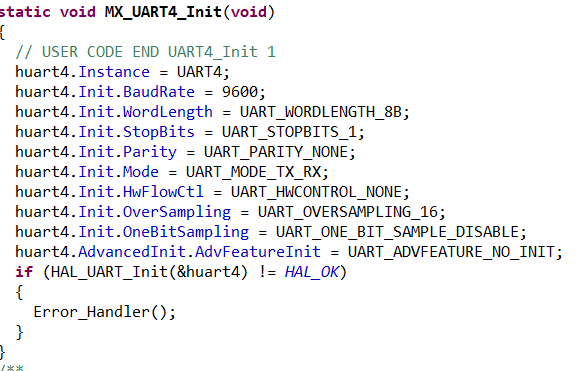




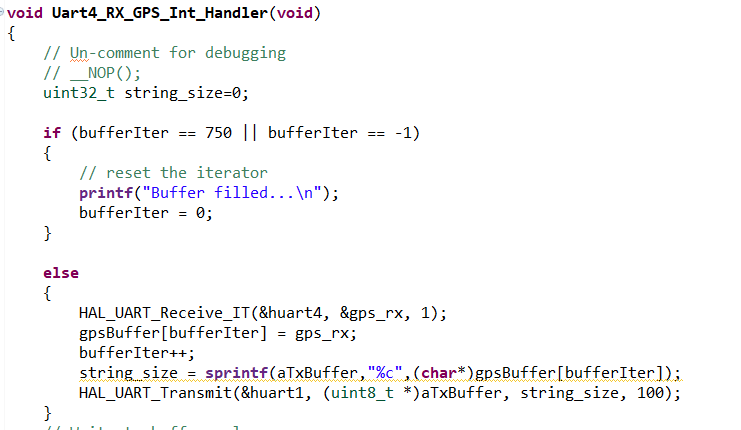
2) We have defined 2 tasks with the same priority named gps\_task and co\_task. With the same priority, these tasks will be executed in round robin fashion. Below is the thread definition of these 2 tasks



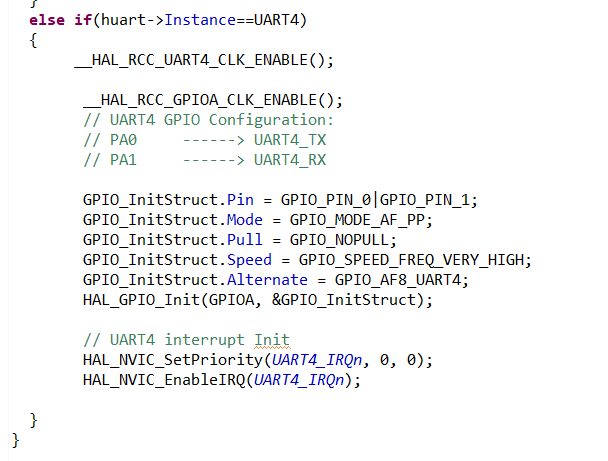
3) To initialize UART4, we need to include MX\_UART4\_Init() in main.c



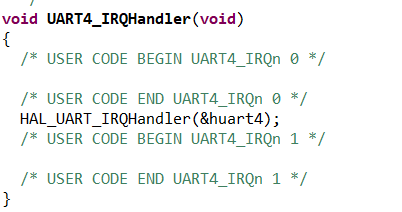
4) Although task1 will be same as CO-Sensor logic, task2 will call **HAL\_Recceive\_IT(uart4)** indicating that when data comes at UART4, it will execute the interrupt routine where it calls UART4\_Rx\_GPS\_Int\_handler() to receive data and store in buffer (DMA approach)



In stm32xx\_hal\_msp.c, we add the GPIO definitions for UART4 and configure the GPIO pins as UART4\_Tx and UART4\_Rx



In stm32xx\_hal\_it.c, add the below function for handling interrupt at UART4



Summary..Contd…

3. After following changes, we compile and run the code in debug mode and observe that Co-Sensor data is printed when task 1 is invoked and GPS data is printed when task 2 is invoked. The delay for the Co-Sensor is kept at 2000ms and GPS at 1ms.

**Thus, GPS and Co-Sensor is successfully integrated together using FREERTOS**

4. Next approach was to combine FreeRTOS with Google IoT project to send both the data over google cloud. I thought of 2 approaches

1. Combining Google IoT into FreeRTOS project (Integrated sensor project)

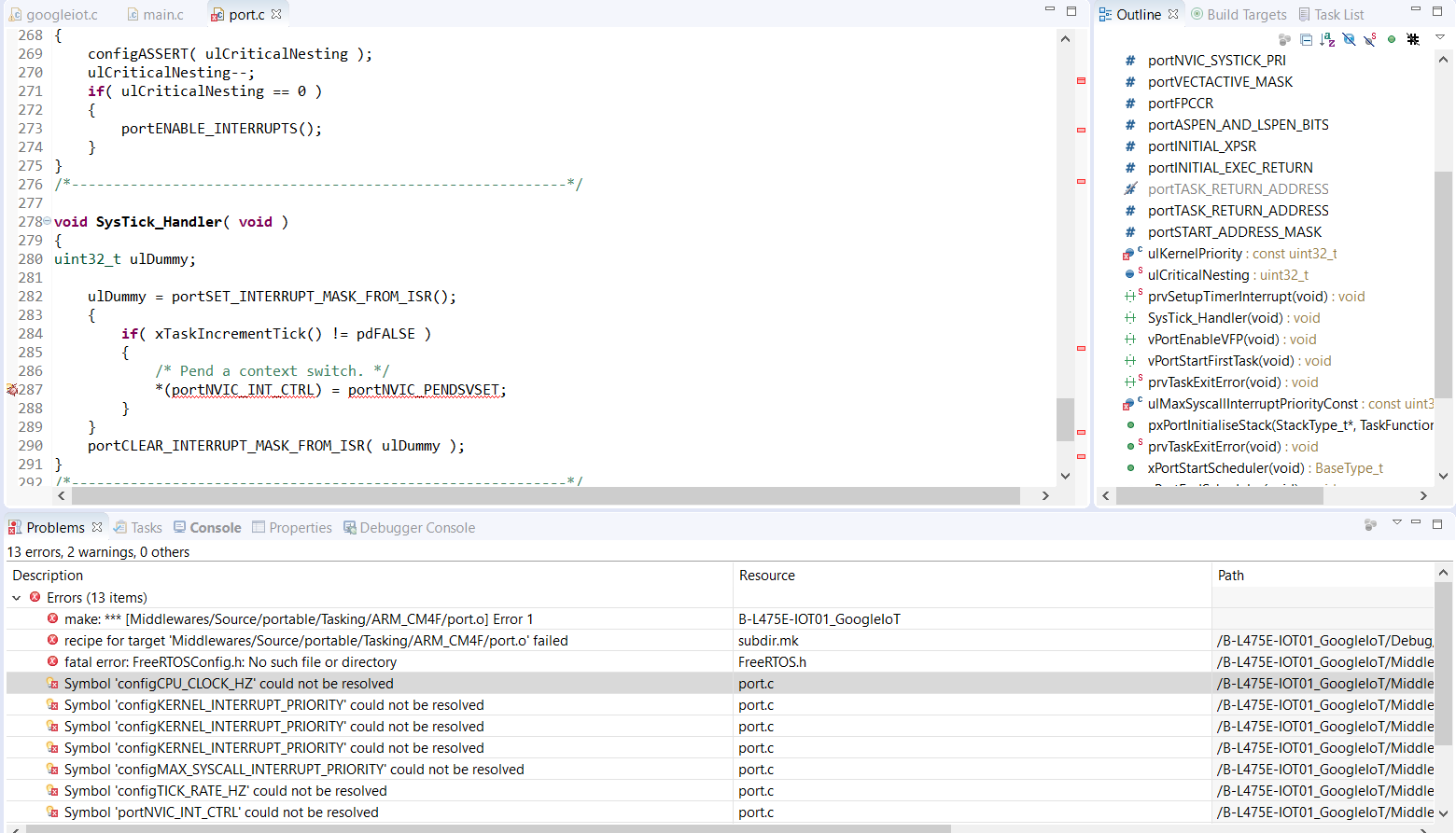
(OR)

1. Importing FreeRTOS package and sensor integration logic into sample Google IoT project

**Approach 1**

I imported the google IOT project from the workspace(B-L475E-IOTA\_GoogleIoT) and added into FreeRTOS project. I added the driver files, middleware files, application files directly. I faced the following problems

1. The files stm32xx\_hal\_msp.c, stm32xx\_hal\_it.c and FreeRTOS files were duplicate so i had to manually compare file to file contents and make changes wherever necessary. **(It took almost a day)**
2. After comparison, I started facing error in the FreeRTOS port.c core files which we are not supposed to edit. Below is the screenshot of few errors



This was taking more time to solve the error. I came up with approach 2 and started integrating FreeRTOS in GoogleIoT project

**Approach 2**

I followed these simple steps to successfully integrate freertos with Google IoT

1. Right click Google IoT project-->import-->file system--->go to STM32CubeExpansion\_Cloud\_GCP\_V1.0.0-->Middleware-->FreeRTOS-->Select OK
2. Go to Sensor Integrated project, copy freeconfig.h (Middleware-->FreeRTOS-->Include folder). Go to eclipse-->Google IoT project-->paste inside FreeRTOS--Include folder
3. In Google Iot project, go to portable folder--GCC--Remove all the folders except ARM\_CM4F
4. In Google Iot project, go to portable folder--MemMang--just keep heap\_4.c
5. In Google Iot project, go to portable folder--Remove all folders except GCC and MemMang
6. Refresh Google Iot project in eclipse
7. In Google Iot project, go to stm32xxx.it.c and comment the functions: PendSV\_Handler(), Sys\_tick\_handler(), SVC\_handler()

**Thus, we are able to integrate FREERTOS with Google IoT**

**Challenges:**

1. **During GPS-CoSensor integration, the data of co-sensor was displaying in teraterm when task1 was invoked, but GPS data was not displayed even when task2 was invoked**

Solution: There was a delay of 2000ms in GPS Task. Due to this delay, GPS was not able to synchronize its UART speed. Reducing the delay helped to solve the problem.

**2. During FreeRTOS integration, port.c file in ARM\_CM4F (FreeRTOS portable folder)**

**was throwing “variable unresolved error”.\**

Solution: The problem was port.c file was also present in other folders like ARM\_CM0, ARM\_CM4 which was clashing with the required file. Removing these extra folder solved the issue.

**Milestones:**

1. **GPS and Co-Sensor was integrated successfully together**
2. **FreeRTOS and Google IOT integration was successful which costed me 3-4 days to work on the errors and fix them**

**Work in Progress: I have created a 3rd task for sending GPS and Co-Sensor over the Google Cloud. There is a problem of task synchronization and os delay which i am currently working on.**