CG3002 FreeRTOS

Team 12

Our Solution:

- Using Timer1_OVF_Vect instead of Timer1_COMPA_Vect, since we are using the Timer 1 in mode 8, PWM, Phase and Frequency Correct mode.
- In this mode, TOP is ICR1 – the Input Capture Register.
- OCR1A is updated at BOTTOM instead, and TOV1 is also set at BOTTOM.
- Other than this, to share the printing resources (Serial print and read object) between the two tasks (one for UltraSonic sensors and motor feedback, the other to obtain the GY-87 readings – this one has higher priority), we are using a Mutex, from the FreeRTOS API (xSemaphoreCreateMutex).
- We are using TKJ Electronics library for the Kalman filtering (Kalman.h), and NewPing library (NewPing.h, NewPing.cpp) to read the UltraSonic sensors.

Difficulties Encountered & Solutions Implemented:

- Before the first prototype evaluation, we replaced the 16-bit Timer 1 (from the original FreeRTOS available for ATMega323) with the 8-bit Timer 0.
- This allowed us to try out a simple 'blinky' program, with two flashing LEDs, one per task, scheduled in a round robin.
- However, when we later added in UltraSonic sensors and made use of the NewPing library to get their readings, we got constant readings of either 0 cm, or 1149 cm.
- In a separate task, were able to make our LilyPad VibeMotors vibrate in regular intervals.
- Using the I2C Dev library we were able to read raw values from the GY-87 (MPU6050 for the accelerometer and gyroscope, and HMC5883L for the magnetometer) on the Arduino IDE.
    - https://github.com/jrowberg/i2cdevlib
    - http://www.i2cdevlib.com/
- However, the same library integrated with the rest of our FreeRTOS2560 setup, gave us all 0 readings for the sensor values.
- We would be getting more stable values if we used Kalman Filtering, so we switched to the TKJ Electronics library for I2C communication and GY-87 readings. Again, the readings when running the code from the Arduino IDE made sense.
    - http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/
    - https://github.com/TKJElectronics/Example-Sketch-for-IMU-including-Kalman-filter
    - https://github.com/tkjelectronics/kalmanfilter

- However, on our FreeRTOS2560 project, we either got all 0 readings, or the error messages: i2c write/read failed, error code: 4.
- Since it was later announced that Arduino functions like *millis()* and *delay()* make use of Timer 0, we realized that was probably why NewPing library and the UltraSonic sensors didn't produce accurate readings (since NewPing library makes use of these functions).
- We then switched to Timer 2, the other 8-bit timer, as recommended. It turned out, that Timer2_COMPA_Vect, was already being used by the NewPing library.
- We didn't want to make changes to the library, so instead we opted to try using the other 16-bit timers: Timers 3 & 5, in CTC mode. But we had the same results – 0 for sensor readings, in most cases.

- On closer inspection, we realized that VTaskDelay() did not seem to be working as expected, and decided it was a timer issue.
- We found a library which had adapted FreeRTOS for Arduino (but for the Arduino IDE), the reverse of what we needed to do. This gave us the idea to try using Timer 1 again, but in mode 8, and to use mutexes to safeguard printing resources.
    - https://github.com/pchickey/arduino-freertos-libs

Source Files Modified:

- Port.c – using Timer 1 in the mode mentioned above.
- FreeRTOSConfig.h: Added this line to be able to use mutexes.

```
#define configUSE_MUTEXES 1
```

- The I2C functions from the TKJ electronics library, in order to enable I2C pass-through and read from the magnetometer (HMC5883l). We did not use the I2C.h file directly; instead, the I2C read and write functions are pasted in our FreeRTOS2560.cpp file.
- Minor changes in *semphr.h, queue.h* and *queue.c* to get the project to build successfully.

Team Members:

Akshay Viswanathan

Larry Lee

Priyadharshini Tamilarasan

Raghav Ramesh

Spatika Narayanan

Supraja Bhavani Sekhar