Assignment 1

Anders L. Hurum

Development of Real-Time Systems

EIT Digital

Assignment 1

andershurum@gmail.com

September 11, 2017

Abstract

The first assignment for the course consisted of creating tasks in FreeRTOS, and getting those tasks to run with different delays and priorities. Further the tasks should have a seperate debug message each.

For more details on the assignment, see the `assignment_1.md` document in the repository at github.

`http://github.com/peakbreaker/tuts_FreeRTOS`

Code

The main function:

```
int main(void)
{
    /* Initializations for heap and trace recorder */
    prvInitialiseHeap();
    vTraceInitTraceData();
    xTickTraceUserEvent = xTraceOpenLabel("tick");

    /* Create the tasks */
    xTaskCreate(Task1, "Task1", 1000, 100, 3, NULL);
    xTaskCreate(Task2, "Task2", 100, 500, 1, NULL);

    // This starts the real-time scheduler
    vTaskStartScheduler();

    // Should not reach here
    for ( ;; );

    return 0;
}
```

One of the tasks:

```
void Task1(int msDelay) {
    // Block for the defined time
    const TickType_t xDelay = msDelay / portTICK_PERIOD_MS;

    for ( ;; ) {
        // Print the message and do the delay
        printf("This is task 1");
        fflush(stdout);
        vTaskDelay( xDelay );
    }
}
```

For the assignment I edited the main.c file. I created two tasks which takes an integer (delay time) as an argument.

Results

The resulting output from the program were as follows :

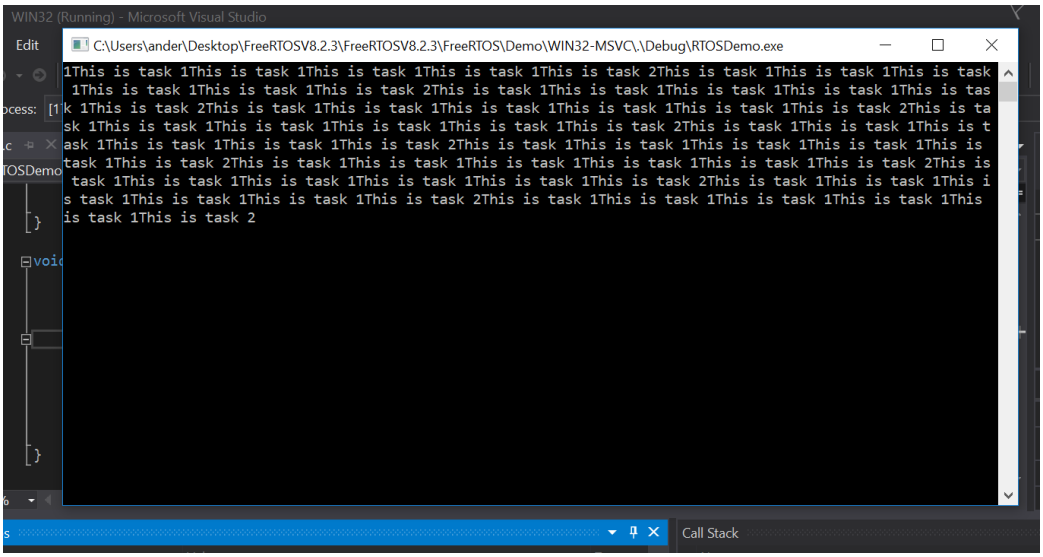

Figure 1: Debug output

As the output shows, Task1 debugs out five times as often as Task2. This meets the specified requirements for the assignment (Task 1 output every 100 ms, Task 2 output every 500 ms).

The repository for the entire assignment can be found at my github :

`http://github.com/peakbreaker/tuts_FreeRTOS`