

## Assignment 2

### Report

#### Code

Since the code for the communication and matrix tasks are already given so its modifications are added in the main file attached and commented on the code but not shown here. The priorityset\_task() task is used to change the priority of the communicationtask based on the execution time of the task.

#### Declarations

```
/* Global declarations*/  
  
volatile TickType_t tickCount           = SET_TO_ZERO;  
volatile TickType_t commTickCount       = SET_TO_ZERO;  
volatile TickType_t matrixTickCount     = SET_TO_ZERO;  
volatile TickType_t prevCommTickCount   = SET_TO_ZERO;  
volatile TickType_t prevMatrixTickCount = SET_TO_ZERO;  
TickType_t comm_time                    = SET_TO_ZERO;  
TickType_t matrix_time                  = SET_TO_ZERO;  
xTaskHandle matrix_handle;  
xTaskHandle communication_handle;  
xTaskHandle priorityset_handle;
```

#### Priorityset task function

```
static void priorityset_task() {  
    static uint8_t commFlag = 0;  
    printf("Priority task started\r\n");  
    while (1) {  
        /* In this condition the priority of the communication task is set to  
        * 4 if the execution time if the greater than 1000 millisecond  
        */  
        if (comm_time > 1000 && commFlag != 4) {  
            printf("Communication task priority increased to 4\r\n");  
            commFlag = 4;  
            vTaskPrioritySet(communication_handle, commFlag);  
            fflush(stdout);  
            vTaskDelay(100);  
        }  
        /* In this condition the priority of the communication task is set to  
        * 2 if the execution time if the less than 200 millisecond  
        */  
        else if (comm_time < 200 && commFlag != 2) {  
            printf("Communication task priority decreased to 2\r\n");  
            commFlag = 2;  
            vTaskPrioritySet(communication_handle, commFlag);  
            fflush(stdout);  
            vTaskDelay(100);  
        }  
        vTaskDelay(1000);  
    }  
}
```

## Main code additions

```
/* Three tasks are created and each task are initially assigned a priority statically.*/
xTaskCreate((pdTASK_CODE)matrix_task, (signed char *)"Matrix", 1000, NULL, 3,
&matrix_handle);
xTaskCreate((pdTASK_CODE)communication_task, (signed char *)"Communication",
configMINIMAL_STACK_SIZE, NULL, 5, &communication_handle);
xTaskCreate((pdTASK_CODE)priorityset_task, (signed char *)"Priority",
configMINIMAL_STACK_SIZE, NULL, 6, &priorityset_handle);
```

## Screenshots

```
Priority task started
Communication task priority decreased to 2
Sending data...
Matrix task count          : 958
Data sent!
Communication task count    : 1887
Communication task priority increased to 4
Sending data...
Matrix task count          : 1029
Data sent!
Communication task count    : 201
Sending data...
Data sent!
Communication task count    : 200
Sending data...
Data sent!
Communication task count    : 200
Sending data...
```

Figure 1: Scheduler execution output focusing on the priority change of communication task

```
Data sent!
Matrix task count          : 1165
Communication task count    : 201
Sending data...
Data sent!
Communication task count    : 200
Sending data...
Data sent!
Communication task count    : 202
Sending data...
Data sent!
Communication task count    : 201
Sending data...
Data sent!
Communication task count    : 201
Sending data...
Data sent!
Communication task count    : 201
Sending data...
Data sent!
Matrix task count          : 1170
Communication task count    : 200
Sending data...
```

Figure 2: Scheduler execution output focusing on the execution time for the matrix task

## Question and answers

- *Why is "matrixtask" using most of the CPU utilization?*

The matrix task takes more computation than the other tasks thus having a higher CPU utilization.

- ***Why must the priority of "communicationtask" increase in order for it to work properly***

The communication task blocks the matrix task from executing so the priority of the communication task must be increased else the matrix task will take a longer time to execute. Lowering the priority of the matrix task makes it pre-empt the communication task.

- ***What happens to the completion time of "matrixtask" when the priority of "communicationtask" is increased?***

Communication task priority increase doesn't have much effect on the matrix task due to the low execution time of the communication task.

- ***How many seconds is the period of "matrixtask"? (Hint: look at vApplicationTickHook() to measure it)***

```
/* The block of code below this is used to calculate the time needed to execute
 * the matrix task and set the matrix_time variable based on the calculated
 * time for further analysis.
 */
    matrixTickCount = tickCount;
    matrix_time = matrixTickCount - prevMatrixTickCount;
    printf("Matrix task count : %d\r\n",
matrix_time);
    fflush(stdout);
    vTaskDelay(100);
    prevMatrixTickCount = matrixTickCount;
```

This part of the code calculates the time period the matrix task takes and prints the output. It is estimated that the period of the matrix task is 1165ms from my execution over a long time execution. Changes in the priority of the task will affect the period.