

Lab 2:

Tasks, Priorities, Synchronization, Queues, Interrupts

Maria Alkeswani & Jamal Alaskari

A. Two tasks with priorities

A.1.

- First we create a setup function:

```
pinMode(5, OUTPUT); // analog number 5 will give the output
xTaskCreate(task,"task", 128, NULL, 1, NULL); // We create a task:
```

task, : The name of task's function.

"task": A descriptive name for the task.

128 : The size of the task's stack in words.

NULL: The pointer to void parameter passed to task's function, is null.

1 : The priority of task.

NULL: Will contain a handle for the created task, is null.

- In the task, every 3s the task turns on the LED for 2s.

```
void task()
{
    while(1)
    {
        digitalWrite(5, HIGH); // analog 5 , high priority
        Serial.println("On"); // LED is on
        delay(2000); // two seconds

        digitalWrite(5, LOW); // analog 5 , low priority
        Serial.println("Off"); // LED is off
        delay(3000); // one seconds
    }
}
```

A.2.

- In this part we create two tasks; task_a and task_b. The tasks are the same as the task as we created in A.1. Each task control its own LED. The tasks have a same priority.

```
pinMode(5, OUTPUT); // task_a, analog 5
pinMode(6, OUTPUT); // task_b, analog 5
xTaskCreate(task_a,"task_1", 128, NULL, 1, NULL); // priority 1
xTaskCreate(task_b,"task_2", 128, NULL, 1, NULL); // priority 1
```

- What kind of scheduling is used for scheduling the tasks?

If there are tasks with the same priority, they will run with time slicing; each of the tasks will run for one slice of time (for one tick). This is called Round-Robin scheduling

A.3.

- In this part we create two tasks. Each task control its own LED with different priority.

```
xTaskCreate(task_a,"task_1", 128, NULL, 1, NULL); // priority 1
```

```
xTaskCreate(task_b,"task_2", 128, NULL, 2, NULL); // priority 2
```

The highest priority task will be run first and then when finished, the second task will be run.

A.4.

In this part we use `vTaskDelay()`:

The execution of the calling task is delayed for duration of `xTicksToDelay` ticks

Use `pdMS_TO_TICKS()` macro to convert milliseconds to tick periods

- E.g., `pdMS_TO_TICKS(2000)` converts 2000 milliseconds into ticks

B. Three tasks with synchronization

B.1.

- In this part we create 3 tasks with periods of all 3 tasks to 100ms.

```
xTaskCreate(task_a,"task_1", 128, NULL, 2, NULL);
```

```
xTaskCreate(task_b,"task_2", 128, NULL, 2, NULL);
```

```
xTaskCreate(task_c,"task_3", 128, NULL, 1, NULL); // has a lower priority than other tasks.
```

- Task1 and task2 will control the LEDs using the PWM.

```
analogWrite(5, pwm); // for task_1
```

```
analogWrite(6, pwm); // for task_2
```

- The periodic task3 will read a command from the serial port.

```
f = Serial.parseFloat(); // to read the floating nuber
```

```
task = (int)f; //to read the integer nuber from the floating number,1.3 _here the int is 1
```

```
f = f-task; // 1.3 -1=0.3 the strong of LED
```

```
f = f*=255 ; // to multiply the 0.3 floating number with 255(the highest analong nubmber)
```

```
pwm =(int) f;
```

```
vTaskDelay(pdMS_TO_TICKS(100));
```

B.2.

- In this part we will change the program so that received commands can be queued by task1 and task2.

```
float f ; //float_number to put strong of led
```

```
int task;
```

```
int pwm; //to put Analog_n 0-255
```

```
QueueHandle_t firstQ = NULL; //to queue commands for task_1
```

```
QueueHandle_t secondQ = NULL; //to queue commands for task_2
```

- In setup function are length of the queues has to be limited to 5 commands.

```
firstQ = xQueueCreate (5, 4); //the first queue is 5 values, 4 is the size of integer number
```

```
secondQ = xQueueCreate (5, 4);//the queue (Q2)is 5 values, 4 is the size of integer number
```

- Task_1:

```
while(1)
{
    xQueueReceive(firstQ, &pwm, 100); //take the last value from Q to use the number (pwm)
    to turn on the LED
    analogWrite(5, pwm); // tern on Led 5 using pwm
    vTaskDelay(pdMS_TO_TICKS(5000)); // delay 5 sec, this function let to another task to
    run while this task is sleeping
}
```

- Task_2 is same Task_1 but the queue is secondQ and on led 6.

B.3.

we used a button to reset for queue1 and queue2, i.e., clear all the existing commands in the queues.