

Intro to IOT

Demo 8: RTOS

In this session, you will learn more about using OS on arduino, and its capabilities. We will work with FreeRTOS (<http://www.freertos.org/>). Its very simple, light, and supports Arduino uno. Find the instruction to install FreeRTOS here:

<https://create.arduino.cc/projecthub/feilipu/using-freertos-multi-tasking-in-arduino-ebc3cc>

You will need to include Arduino_FreeRTOS.h in your projects.

The template for using RTOS looks like this:

```
#include <Arduino_FreeRTOS.h>
#include <semphr.h> // add the FreeRTOS functions for Semaphores

// define two Tasks for DigitalRead & AnalogRead
void task1( void *pvParameters );
void task2( void *pvParameters );

// the setup function runs once when you press reset or power the board
void setup() {
    // Now set up two Tasks to run independently.
    xTaskCreate( task1, "name1",..... , Priority,... );
    xTaskCreate( task2, "name2",..... , Priority,...);
}

void loop()
{
    // Empty. Things are done in Tasks.
}

/*----- Tasks -----*/
void Task1( void *pvParameters __attribute__((unused)) ) // This is a Task.
{ //some functions, and maybe using Semaphores
}

void Task2( void *pvParameters __attribute__((unused)) ) // This is a Task.
{ //some functions, and maybe using Semaphores
}
```

As you can see, there is nothing in `loop()`, and the task can be done in *Task*, and it's possible to set priority for them. Note that `loop()` cannot be removed from code.

NOTE: You might read <http://www.freertos.org/implementation/a00005.html> for more information regarding FreeRTOS scheduler and task priority.

Here are some basic functions of RTOS

(source: <http://www.instructables.com/id/FreeRTOS-With-Arduino-09-Read-Task-Info-VTaskList/>):

xTaskCreate(): create a new Task

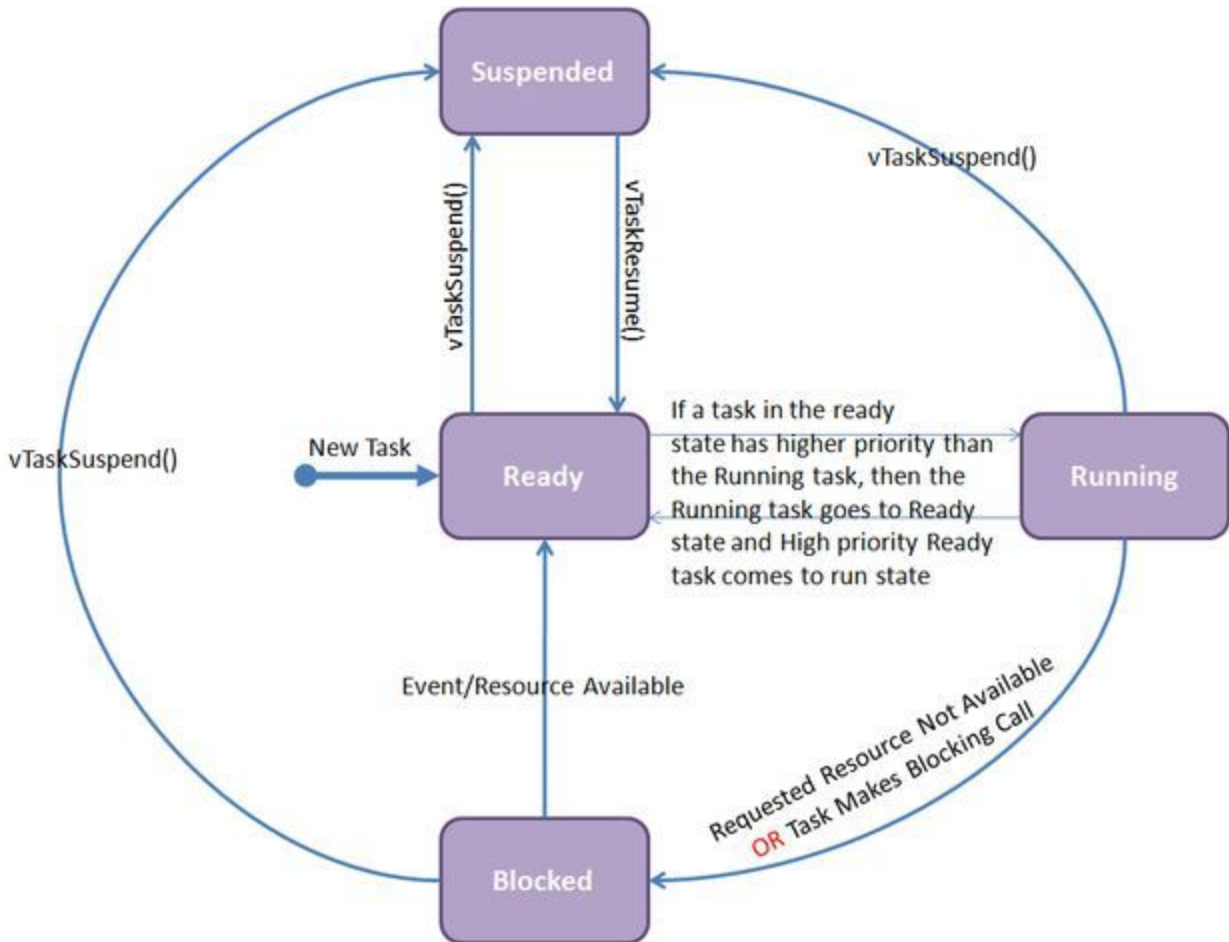
vTaskDelay(): delay/block the task for specified delay time(ticks). You can use it to block the task.

vTaskSuspend(): Suspend a task, the suspended remains in the same state until it is resumed. For this, we need to pass the handle of the tasks that needs to be suspended.

vTaskResume(): resume a suspended task. If the Resumed task has higher priority than the running task then it will preempt the running task or else stays in ready state. For this, we need to pass the handle of the task to be resumed.

vTaskList(): read the task details(name, state, priority, num). We need to pass a string pointer(buffer) to which it copies the above task details.

The following figure shows the life cycle of tasks:



[source:<http://www.instructables.com/id/FreeRTOS-With-Arduino-02-Task-Switching/>]

Your job is to deliver a design with three tasks. You have to know how to create new task, use semaphore, and prioritise the tasks.

task 1: read data from temperature sensor each 100 ms if the LED is not being used by other tasks. this task has highest priority(3). When it finished reading data, it has to toggle the LED by 4 times ,and before each toggle wait for 10 clock tick(using vTaskDelay). for this purpose, it has to take the semaphore. At give it back when it finished.

task 2: it has to toggle the LED by 2 times, and before each toggle wait for 60 clock tick(using vTaskDelay). This task has lower priority(1). When it turns the LED on, it has to take the semaphore. And release it when it turns it off.

task 3: read data from light sensor each 100 ms if the LED is not being used by other tasks. this task has medium priority(2). When it finished reading data, it has to toggle the LED by 2 times, and before each toggle wait for 60 clock tick(using vTaskDelay). For this purpose, it has to take the semaphore. At give it back when it finished.

When each task is running, print its name in the serial port. It will help you trace the execution of tasks. Set the delay between toggling 2 tick delays, i.e. `vTaskDelay(2)`. Note that each tick is around 15ms.

- a) After giving back the semaphore, don't set any delay. Explain how these three tasks runs, and why.
- b) After giving back the semaphore, wait for few millisecond using `vTaskDelay(1)` for all tasks. Explain how these three tasks runs, and why.
- c) Now set the priority of task 2 to 2. (the same as task 3), what happened? Explain how these three tasks runs, and why.

Upload the answer to these questions to EEE dropbox.