**Lab Report**

Major: Embedded Systems Design

Student number: 192050194

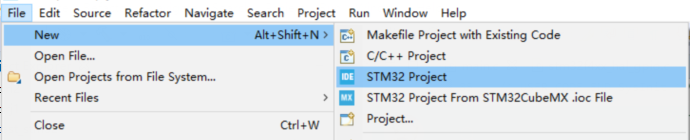
Student name: Yang Kang

Teacher: Arkady Kluchev and Sergei Bykovskii

Date: 23 June 20

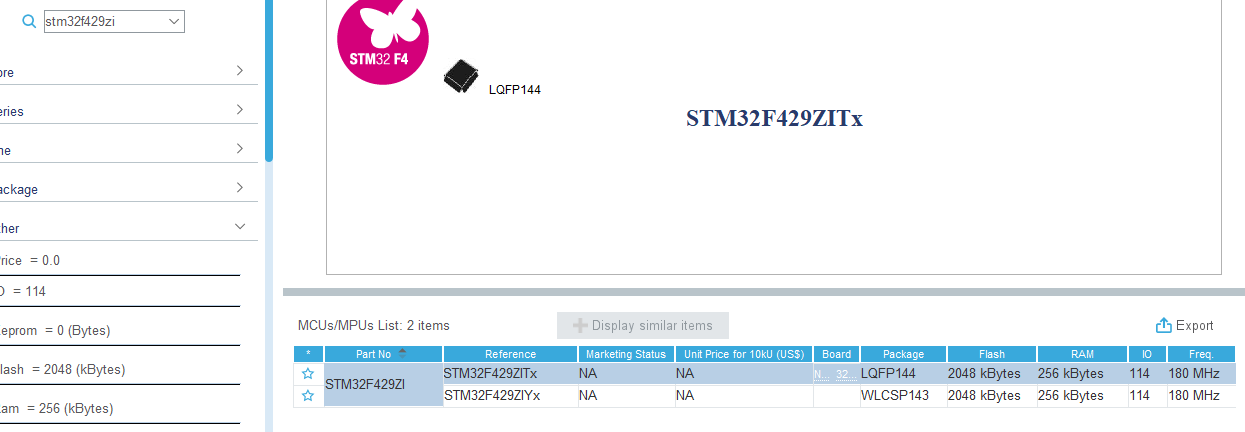
Step1:

Create an stm32 project



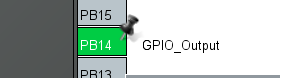
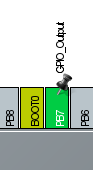
Step2:

Select the corresponding model, and create

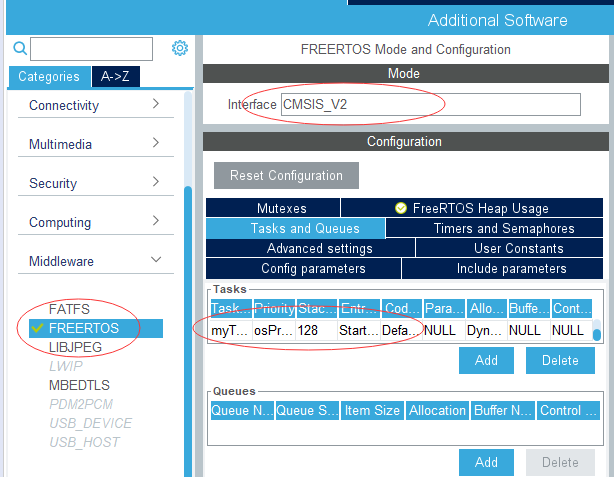


Step3:

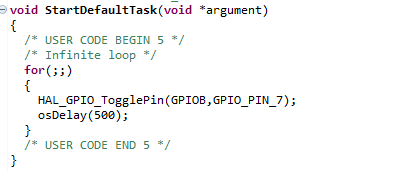
Find pb7 and pb14, select GPIO\_Output, and save.



Step4:



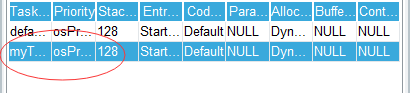
Choose freeRTOS select CMSIS\_V2 and create code.



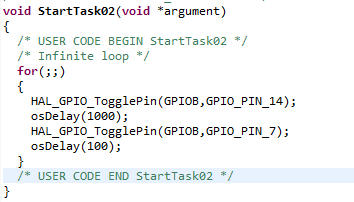
Than write code.

Step5:

Create another task.



Write code and debug it.



1. What features of FreeRTOS do you remember?

(1) Both preemption and collaboration priorities are supported;

(2) Very flexible task priority assignment;

(3) Flexible, fast and lightweight task notification mechanism;

(4)Very portable source code structure, predominantly written in C.

(5) [Mutexes](http://www.freertos.org/Real-time-embedded-RTOS-mutexes.html) with priority inheritance.

(6) Free embedded software source code.

1. What are the differences between FreeRTOS, OpenRTOS and SafeRTOS?

1.FreeRTOS and OpenRTOS share the same source code and OpenRTOS is used for commercial purposes.

2.SafeRTOS is also based on FreeRTOS, but unlike FreeRTOS, it has been redesigned by security experts.

1. Why do we need the vTaskStartScheduler() function?

When running the program, we need a basic task to get into the running state.A free task is created when vTaskStartScheduler() is called to ensure that no errors occur in the program.It is mainly a protection for the program.

1. Why do we need the xTaskCreate() function?

Our task requires RAM to hold the state of the task and is used as a stack.At this point, we need to use xTaskCreate() to dynamically request memory to hold the various running states.