AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
Credit Hours Programs

SummerSemester, 2021

CSE 347: Embedded Systems

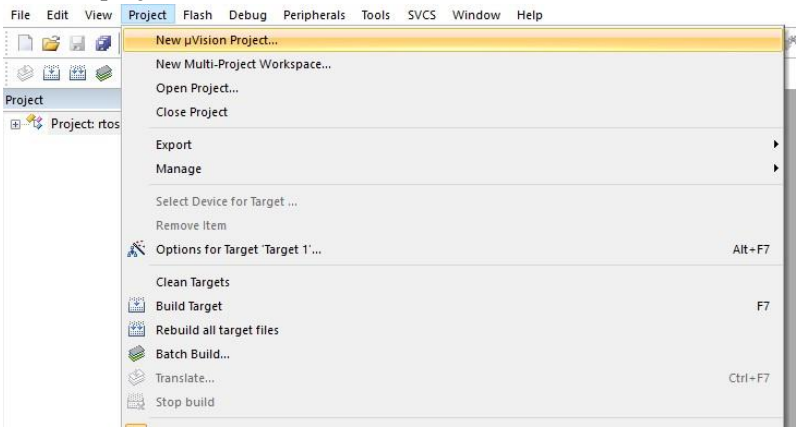
Lab. 3: Introduction to FreeRTOS

Goals of this Lab:

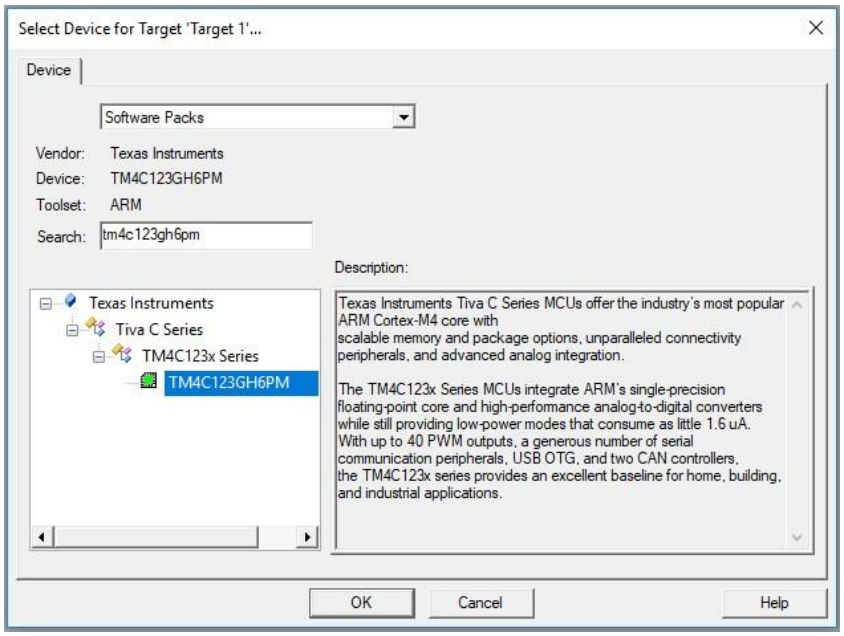
- Create a new project on kiel with FreeRTOS integrated in the project
- Adding tasks in FreeRTOS

Task 1: Initialize the Micro-controller:

1- Create a new project on Keil

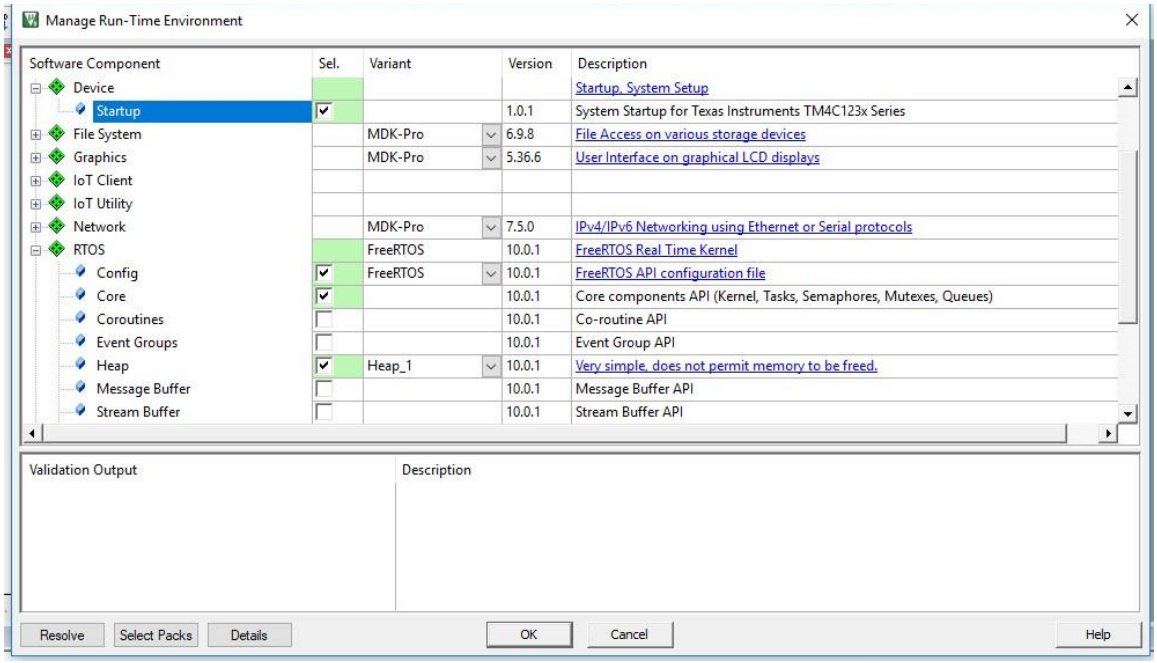


2- Save your project in the desired place, and choose the target

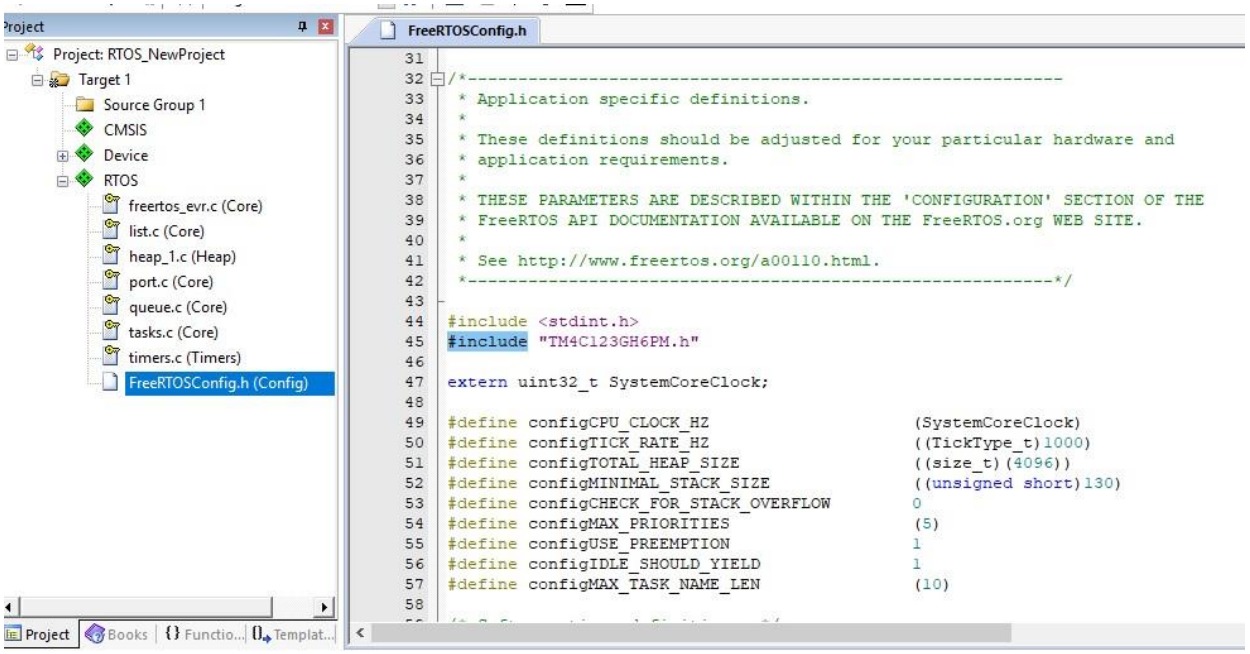


3- In the Manage Run Time Environment window, check the following boxes:
   - CMSIS -> Core
   - Device -> startup - RTOS -> Config Core

Heap (select Heap_1 from the drop down box)
Timers



4- In the Project menu on the left side Open the RTOS folder and open the file FreeRTOSConfig.h and include “TM4C123GH6PM.h” inside it.



5- Create a new main.c file, and start your code.

Task 2: Initialize Tasks

- Create a function that contains a for loop without conditions, ie: for(; ;), make sure that this function is implemented outside the main function.
- Inside the for Loop, implement a function that every time it get executed it toggles the RED LED
- Using the xTaskCreate API create a task and assign the function to it as per section 1.4 in the attached document

Richard Barry-Using the FreeRTOS Real Time Kernel - A Practical Guide - Cortex-M3 Edition.pdf

- After creating the task start the task by calling the scheduler function “vTaskStartScheduler();”
- Just before you start running the code, call the vTaskDelay() after calling the function that toggles the RED LED. Assign a 1000ms delay in the task