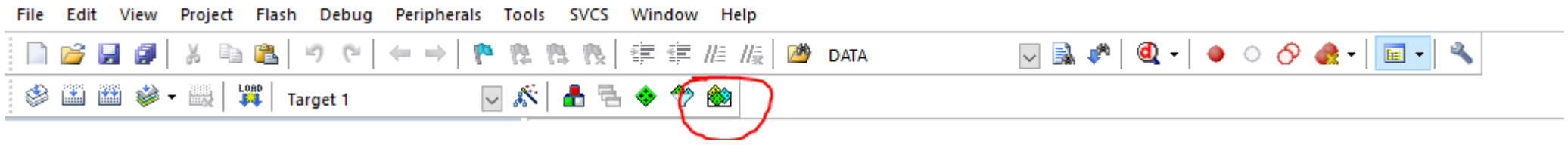
freeRTOS Setup on the Tiva C Launchpad

Target: TM4C123GHPM on the Tiva C Launchpad
IDE: Kiel uVision5 V:5.38.0.0

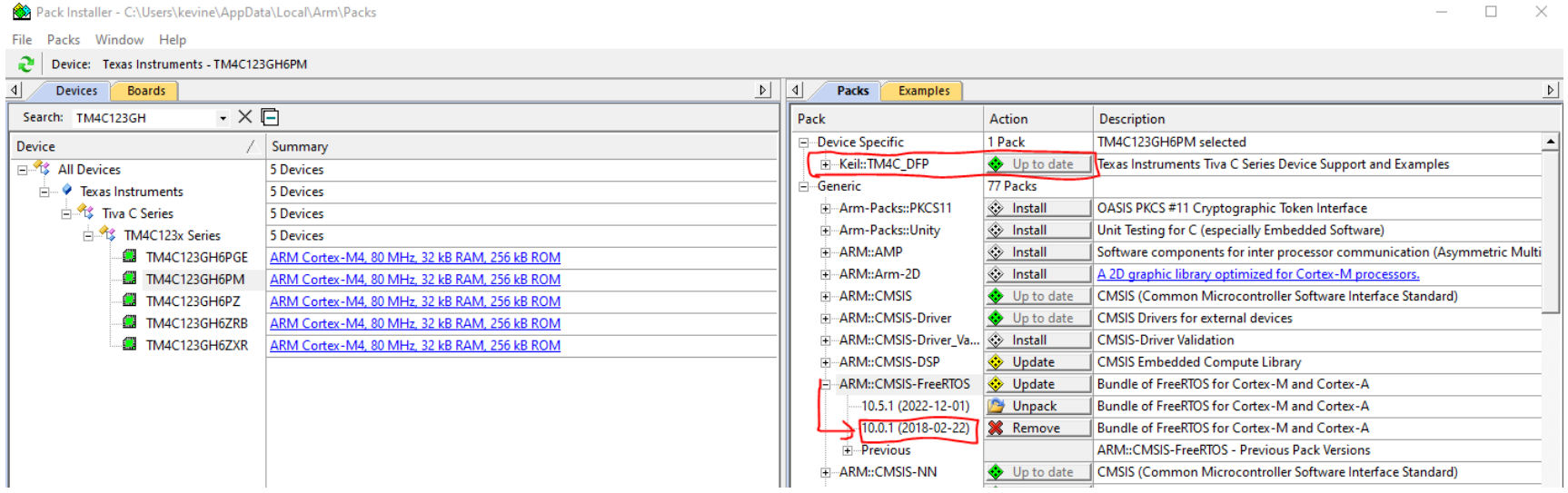
Kevin Ehrichs – CS467 – Team Unicorn

Install Packs

- Open Kiel uVision5 and click on the pack installer Icon

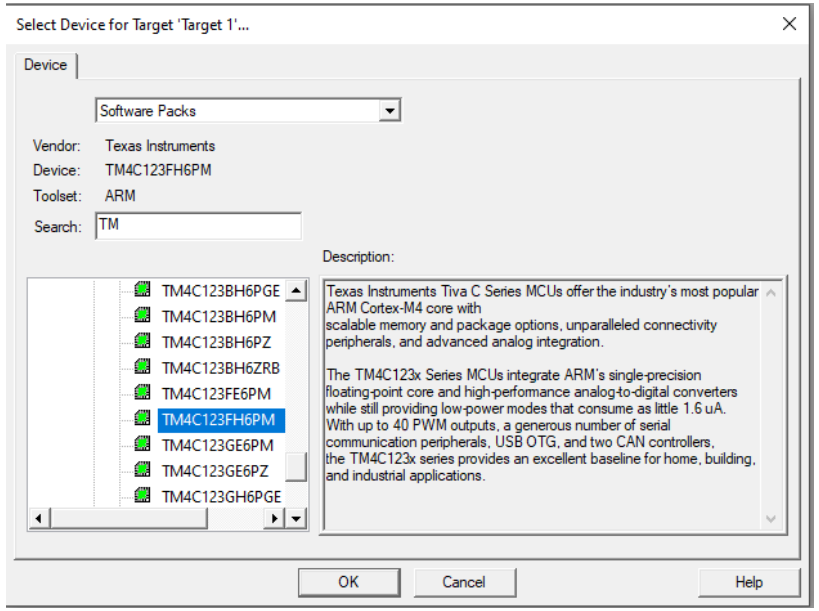


- Install the TM4C_DFP Package
- Install the CMSIS-FreeRTOS Package (V10.0.1 was used in this setup)



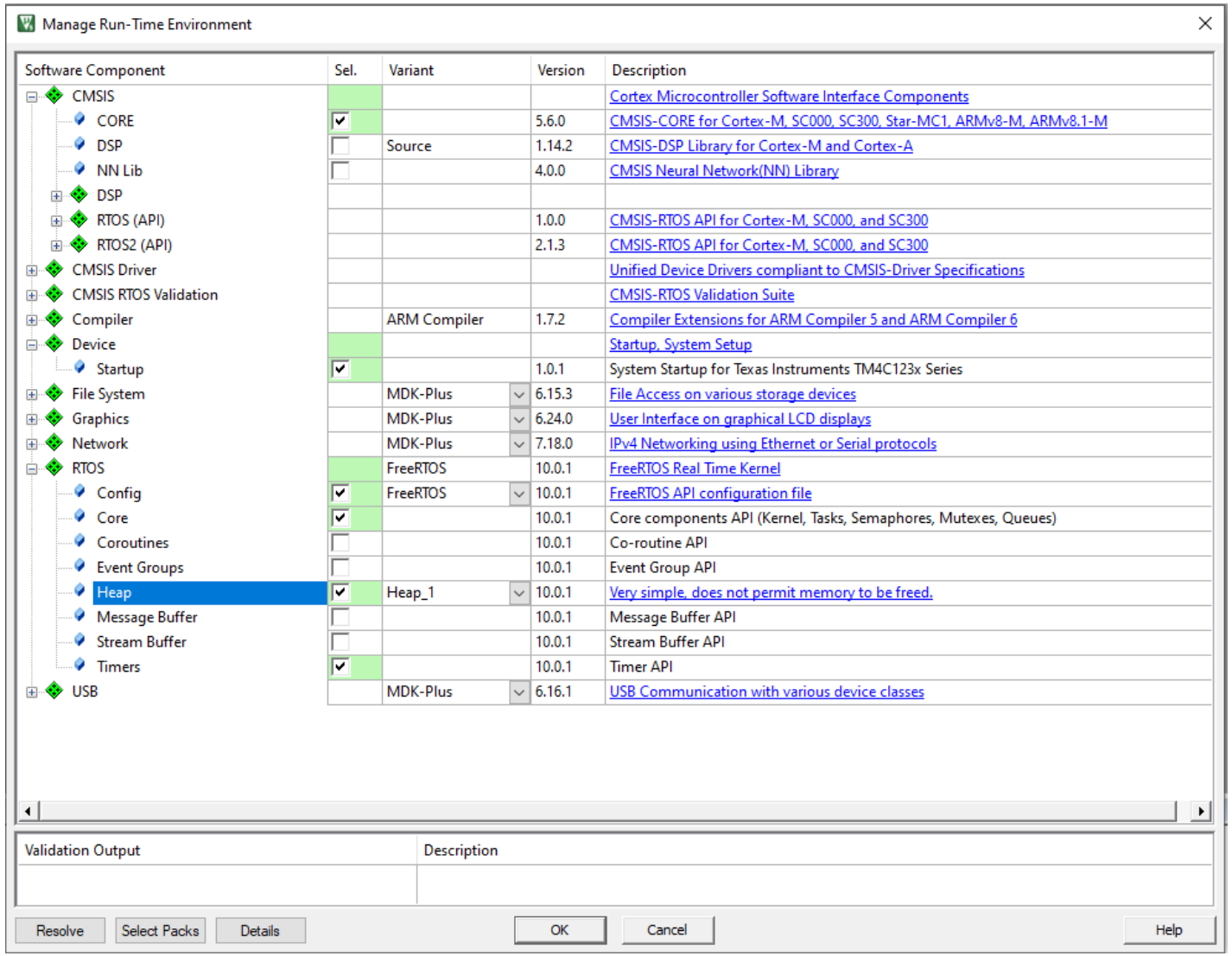
Create a new project

- Select Project-> New uVision Project
- Create a name for the project – save to a new folder
- Then select the correct target and click OK



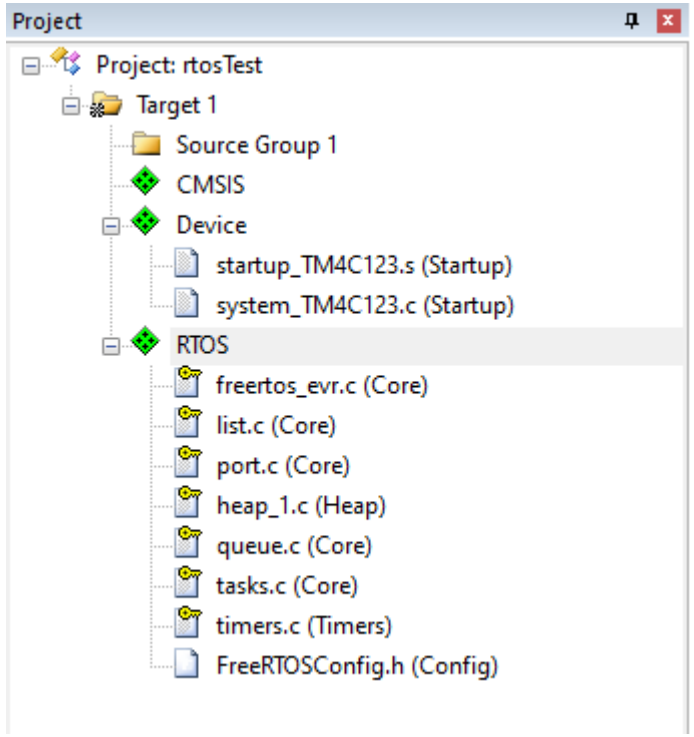
Setup Project Dependancies

- Select the following dependencies taking care of the following:
  - FreeRTOS is being used (V10.0.1)
  - Heap_1 selected * this may change as the project is more understood
  - Once Finished, Click OK



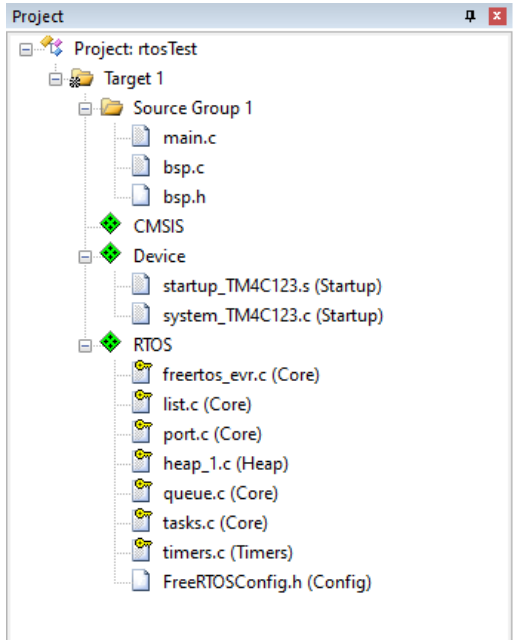
Check Project Explorer

- Your project template is now created, ensure the following structure
- Notice there is no main.c file…



Add main.c and BSP

- Head over to the Unicorn RTOS Github repo and collect a copy of the following files from the freeRTOS branch
- Main.c, bsp.c, bsp.h
- After adding the files, your project structure should now look like this:



Modify the Target Header File

- Modify the following file in the GPIOA Register map
- This is a read-only file - Modify under properties to allow editing
- This file also exists in the repo – can also simply replace the file

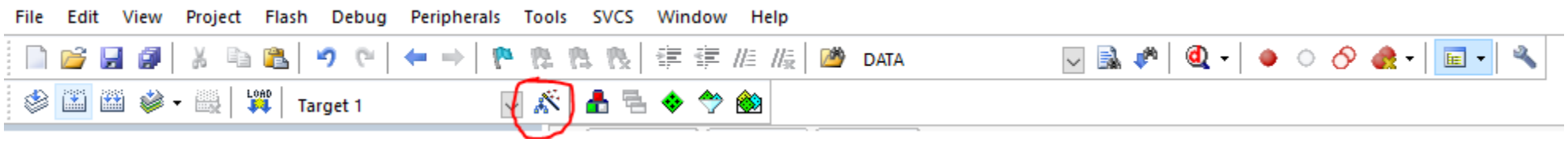
main.c | bsp.c | bsp.h | TM4C123GH6PM.h

```
} WATCHDOG0_Type;

/* ============================================================================ */
/* ================                      GPIOA                     ================ */
/* ============================================================================ */


/**
  * @brief Register map for GPIOA peripheral (GPIOA)
  */

typedef struct {                                  /*!< GPIOA Structure                                          */
  __IO  uint32_t  DATA_Bits[255];
  __IO uint32_t  DATA;                            /*!< GPIO Data                                                */
  __IO uint32_t  DIR;                             /*!< GPIO Direction                                           */
  __IO uint32_t  IS;                              /*!< GPIO Interrupt Sense                                     */
  __IO uint32_t  IBE;                             /*!< GPIO Interrupt Both Edges                                */
  __IO uint32_t  IEV;                             /*!< GPIO Interrupt Event                                     */
  __IO uint32_t  IM;                              /*!< GPIO Interrupt Mask                                      */
  __IO uint32_t  RIS;                             /*!< GPIO Raw Interrupt Status                                */
  __IO uint32_t  MIS;                             /*!< GPIO Masked Interrupt Status                             */
  __O  uint32_t  ICR;                             /*!< GPIO Interrupt Clear                                     */
  __IO uint32_t  AFSEL;                           /*!< GPIO Alternate Function Select                           */
```

Setup Compiler and Other Options

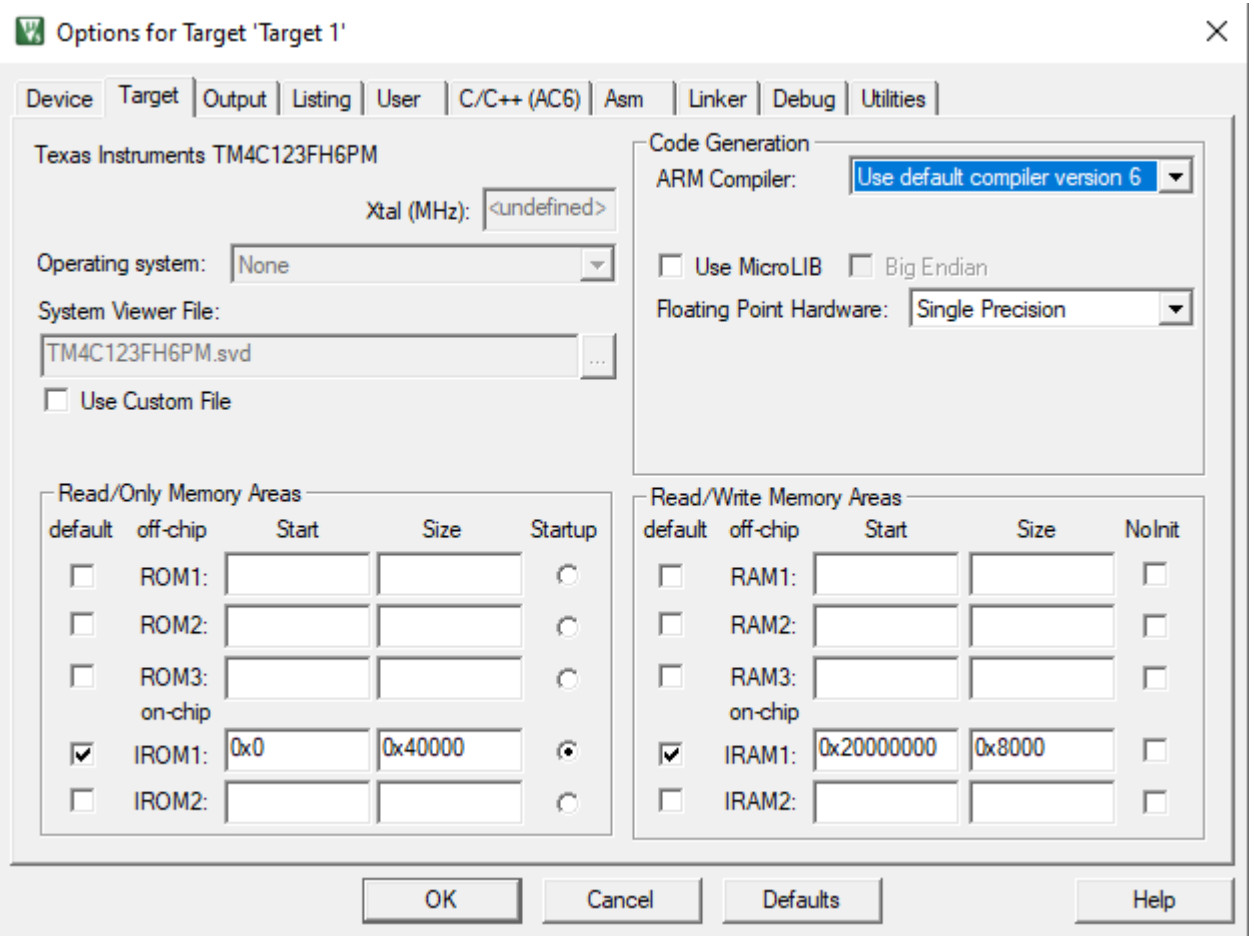
- If you try to compile the code at this point, you should receive a lot of errors and warnings.
- We still need to setup the project settings / Compiler

To do this, Open the following View (Options for Target)



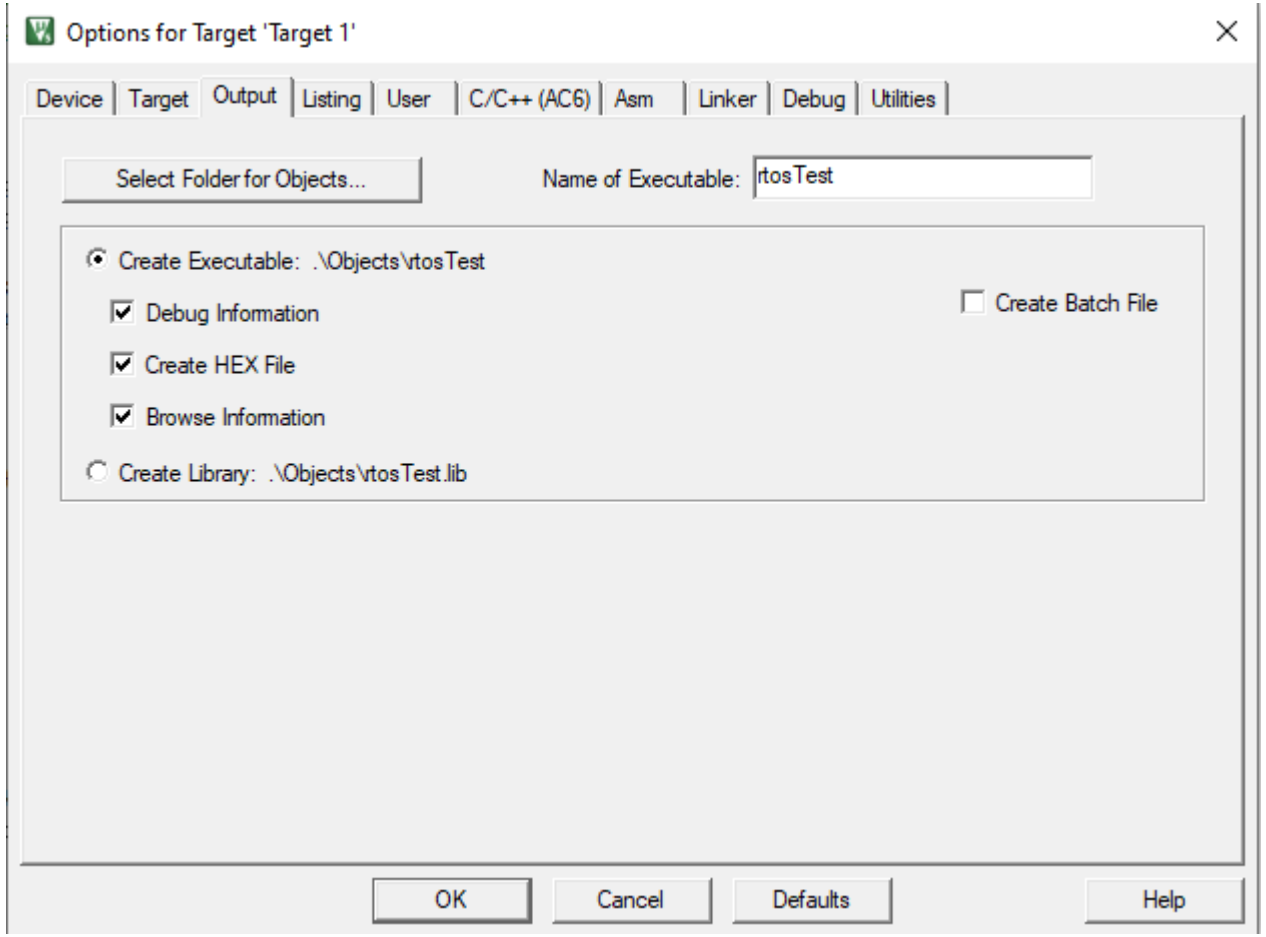
Options for Target: Target Tab

- Ensure the following options are set:



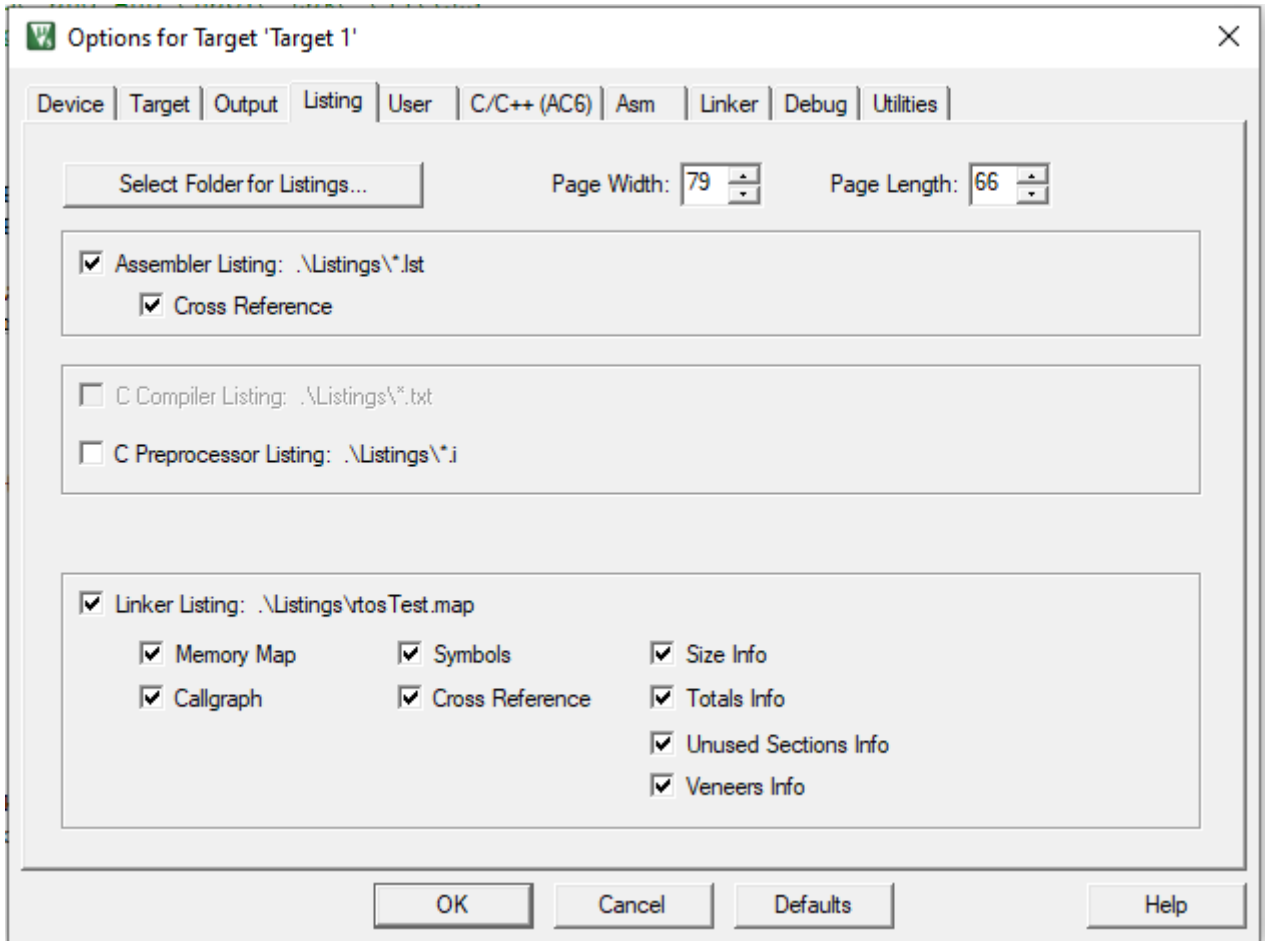
Options for Target: Output Tab

- Ensure the following options are set



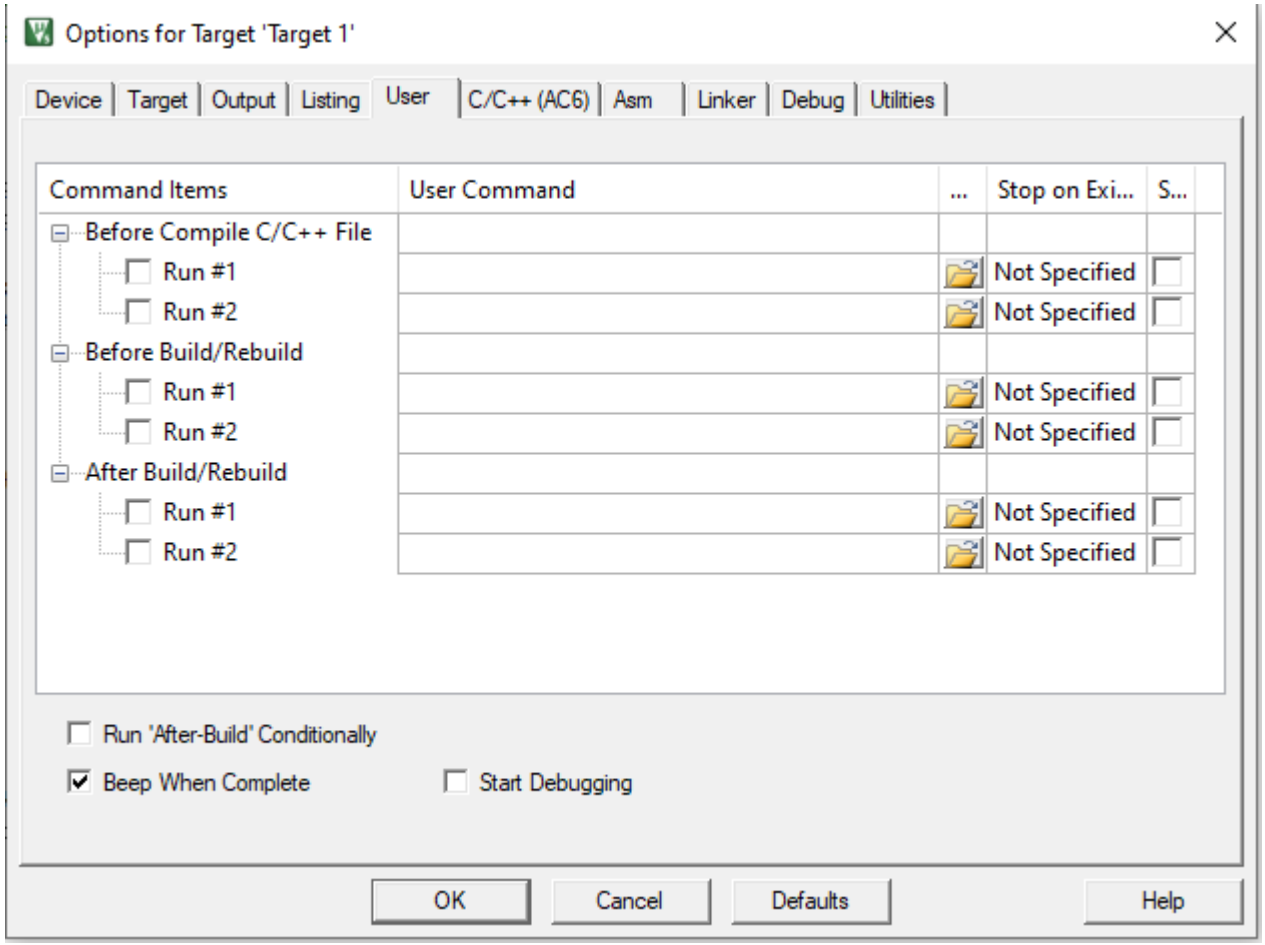
Options for Target: Listing Tab

- Ensure the following options are set



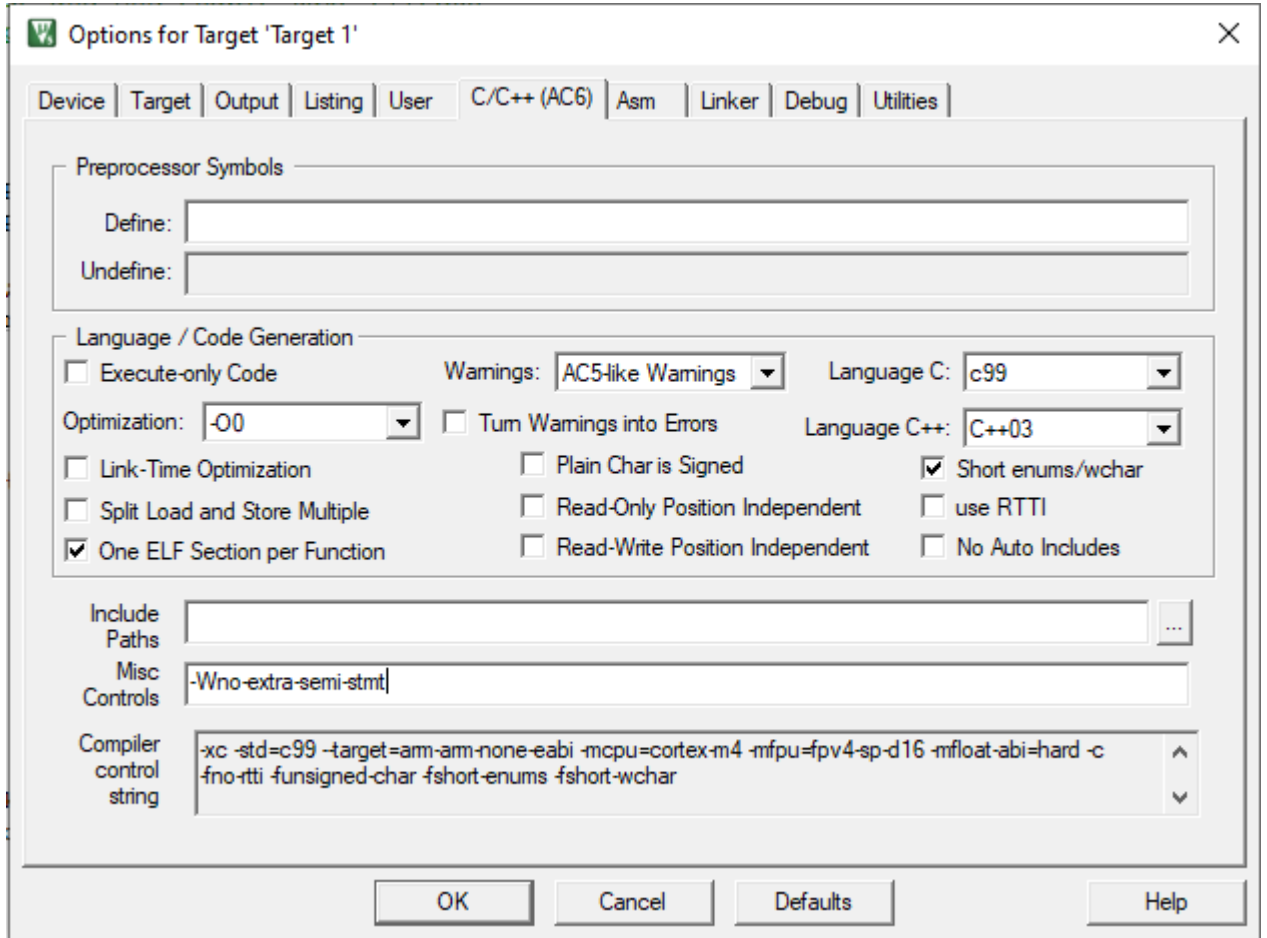
Options for Target: User Tab

- Ensure the following options are set



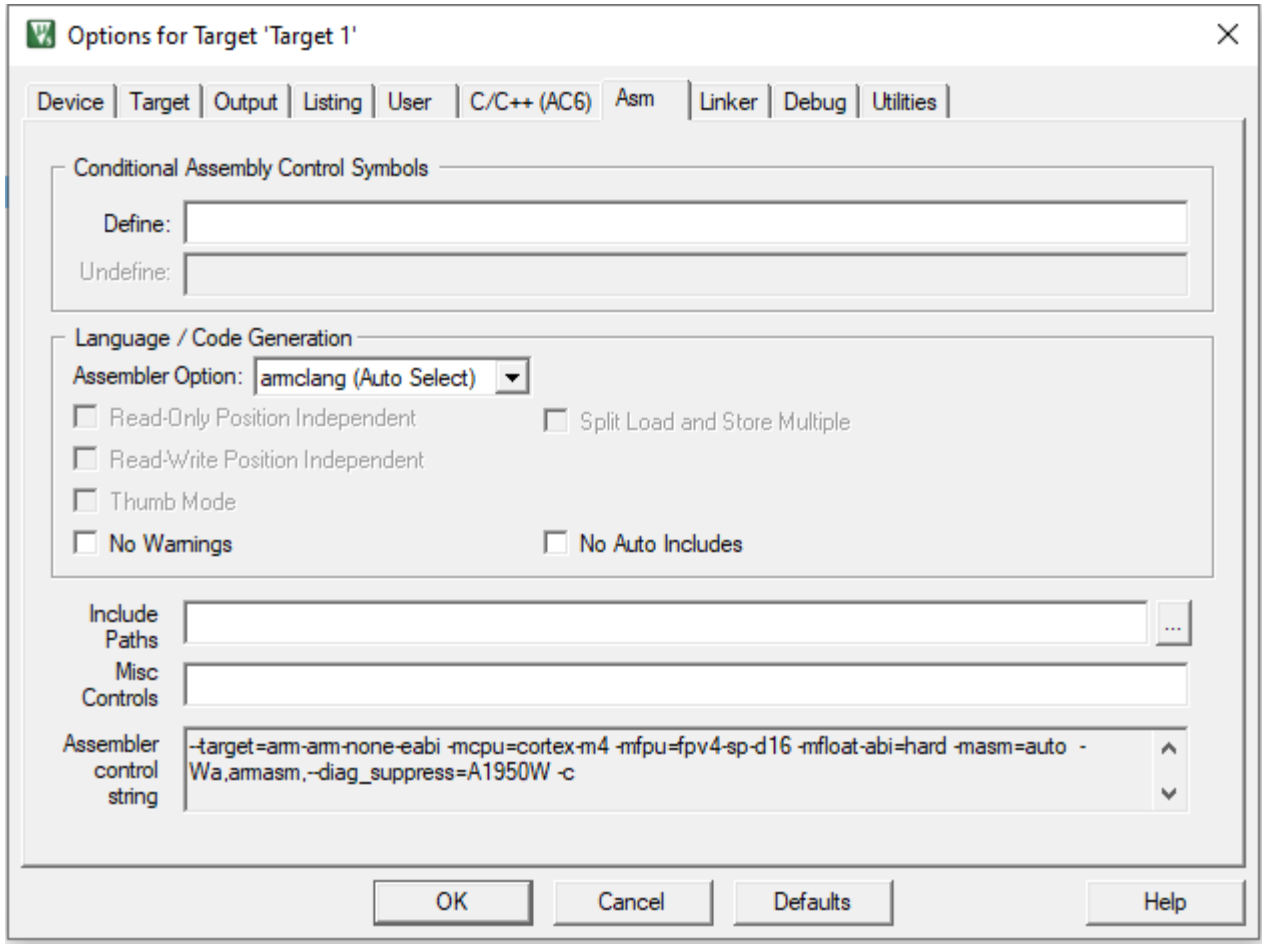
Options for Target: C/C++ (AC6) Tab

- Ensure the following options are set



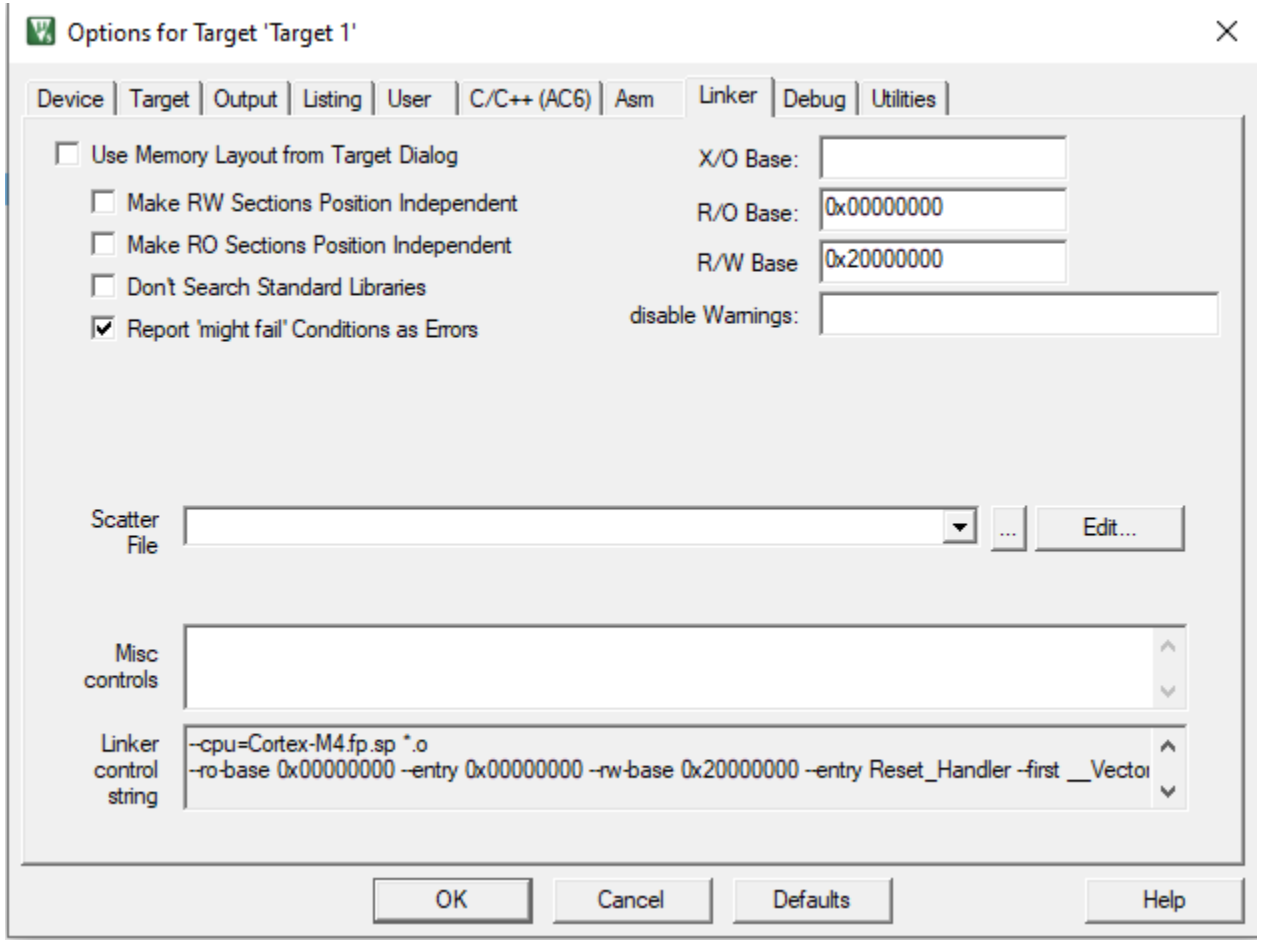
Options for Target: Asm Tab

- Ensure the following options are set



Options for Target: Linker Tab

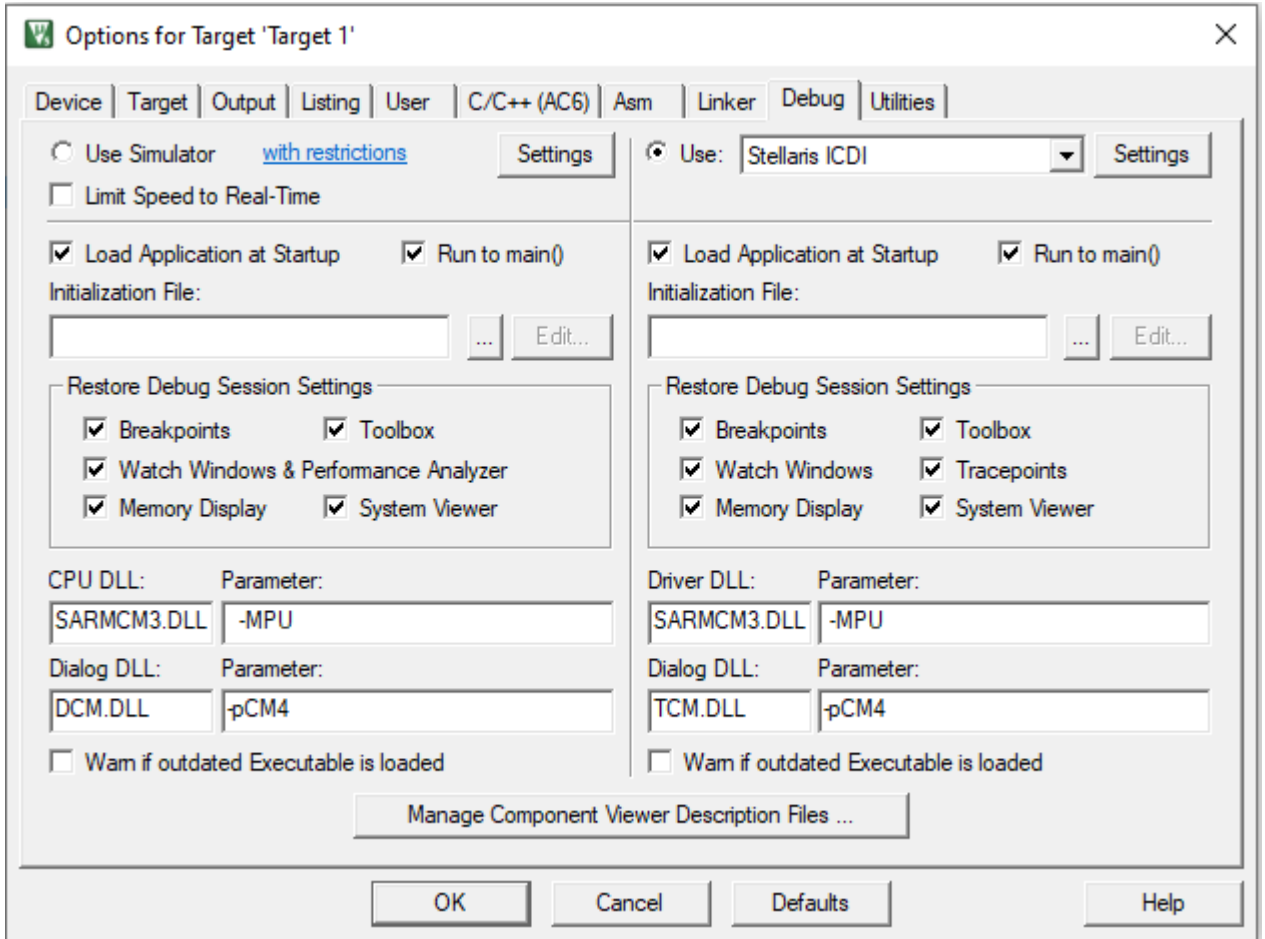
- Ensure the following options are set



Options for Target: Debug Tab

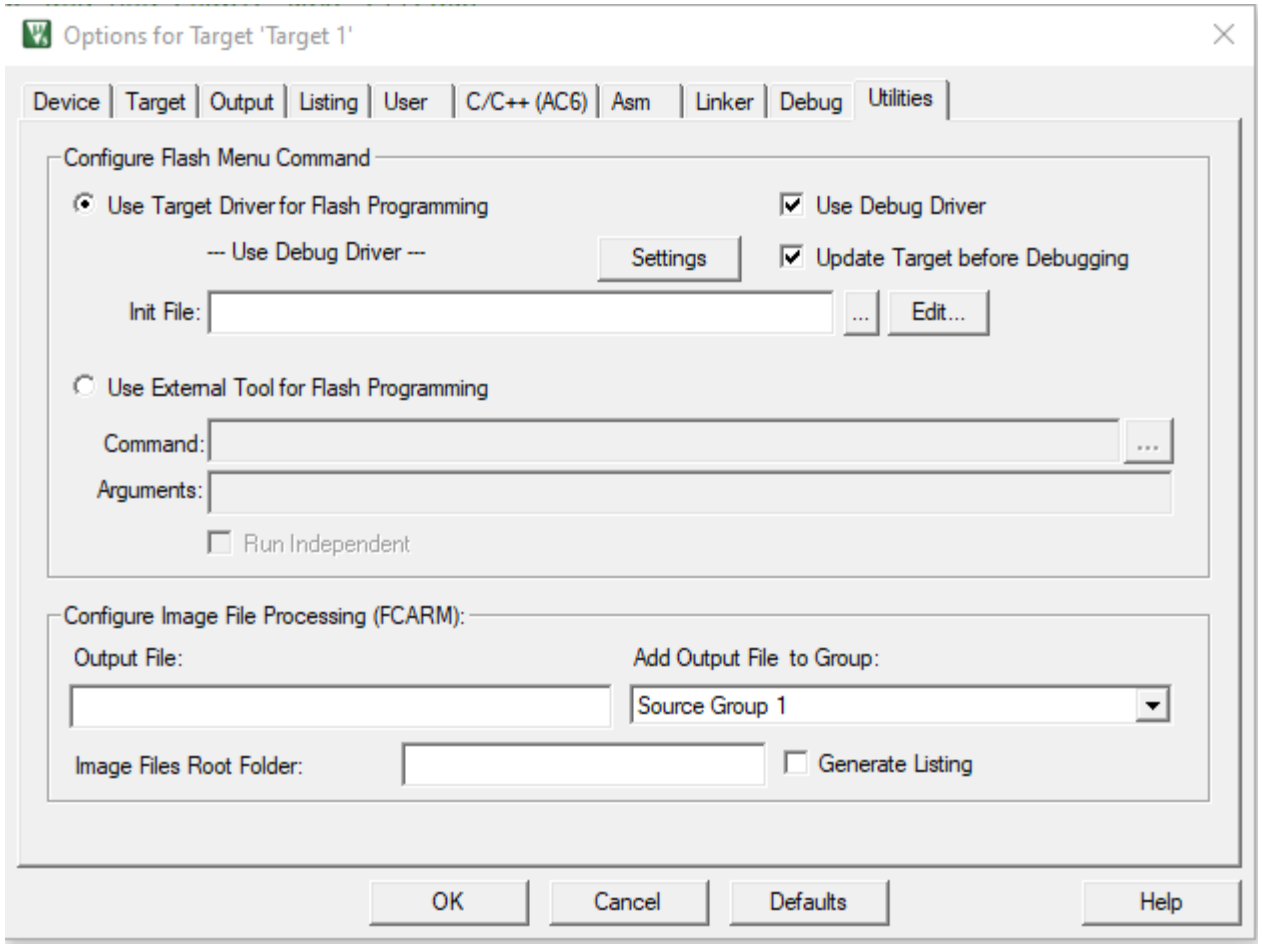
- Ensure the following options are set

Note: If you don't have a Stellaris ICDI listing please download the extension here: https://developer.arm.com/documentation/ka002280/latest



Options for Target: Utilities Tab

- Ensure the following options are set



Clean Project and Rebuild

- After verifying correct Project Options:

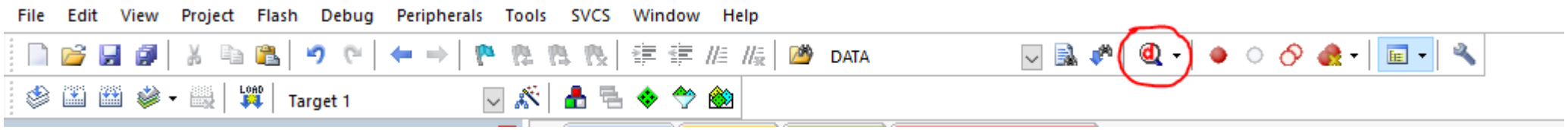
- Select Project->Clean Targets
- Select Project->Rebuild all Target Files
- The build output should now show 0 Errors / 0 Warnings

Build Output

```
compiling main.c...
compiling bsp.c...
compiling heap_1.c...
compiling port.c...
compiling timers.c...
compiling queue.c...
compiling tasks.c...
linking...
Program Size: Code=11850 RO-data=674 RW-data=8 ZI-data=5040
FromELF: creating hex file...
".\Objects\rtosTest.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

Running the code on-board

- Watch out, that LED is BRIGHT
- Connect your board to an available USB port
- Run the debugger by clicking on the debug icon



Running the code on-board

- Before running the code, place three breakpoints:

```c
6   void vPeriodicTask(void *pvParameters)
7   {
8
9     /* Establish the task's period.*/
10    const TickType_t xDelay = pdMS_TO_TICKS(1000);
11    TickType_t xLastWakeTime = xTaskGetTickCount();
12
13    for (;;) {
14      BSP_ledGreenOn();
15
16      /* Block until the next release time.*/
17      vTaskDelayUntil(&xLastWakeTime, xDelay);
18
19      BSP_ledGreenOff();
20
21      /* Block until the next release time.*/
22      vTaskDelayUntil(&xLastWakeTime, xDelay);
23    }
24
25  }
26
```

Green Blink Task

```c
48  void vvvPeriodicTask(void *pvParameters)
49  {
50
51    /* Establish the task's period.*/
52    const TickType_t xDelay = pdMS_TO_TICKS(1000);
53    TickType_t xLastWakeTime = xTaskGetTickCount();
54
55    for (;;) {
56      BSP_ledRedOn();
57
58      /* Block until the next release time.*/
59      vTaskDelayUntil(&xLastWakeTime, xDelay);
60
61      BSP_ledRedOff();
62
63      /* Block until the next release time.*/
64      vTaskDelayUntil(&xLastWakeTime, xDelay);
65    }
66
67  }
68
```

Red Blink Task

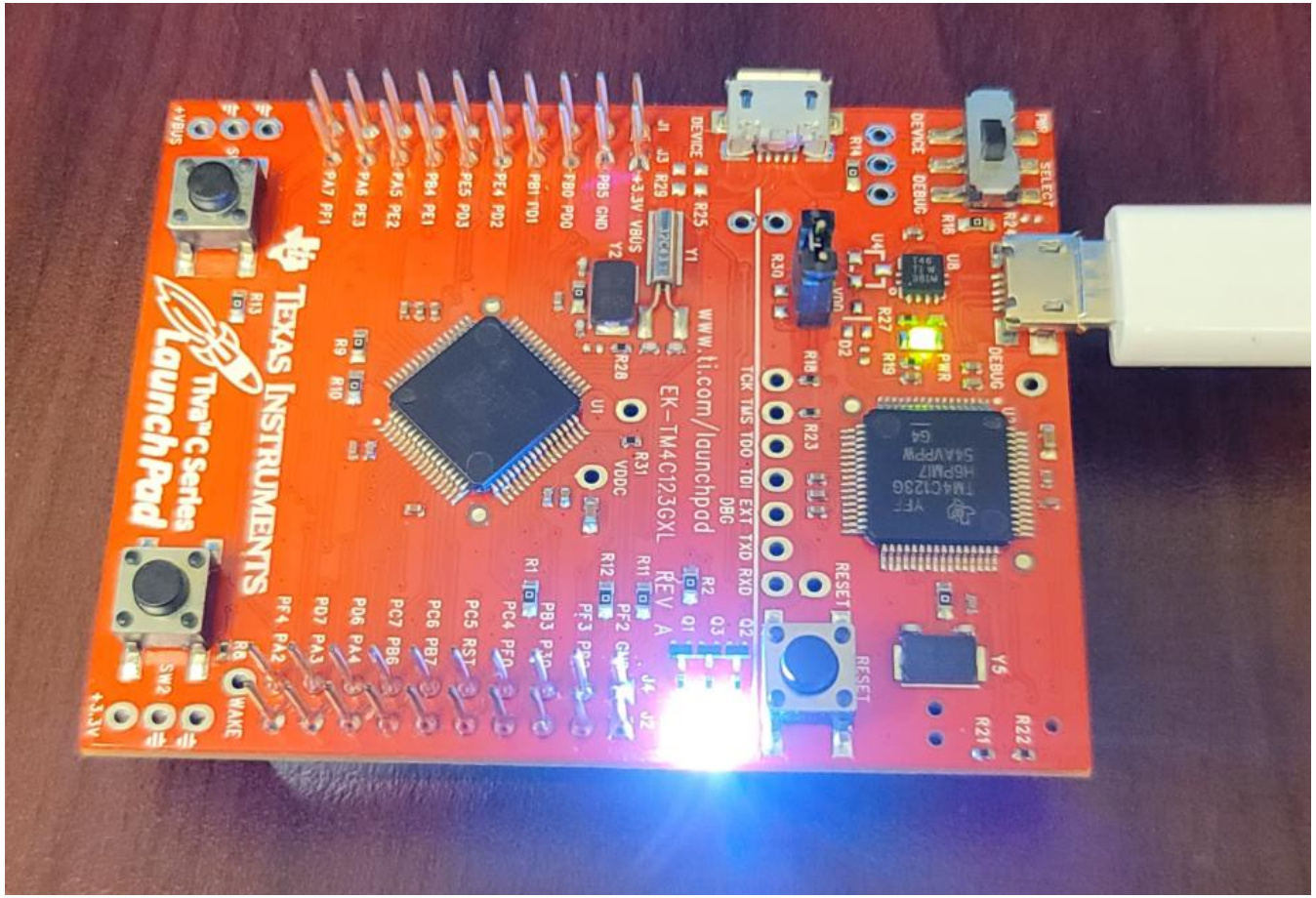
```c
27  void vvPeriodicTask(void *pvParameters)
28  {
29
30    /* Establish the task's period.*/
31    const TickType_t xDelay = pdMS_TO_TICKS(1000);
32    TickType_t xLastWakeTime = xTaskGetTickCount();
33
34    for (;;) {
35      BSP_ledBlueOn();
36
37      /* Block until the next release time.*/
38      vTaskDelayUntil(&xLastWakeTime, xDelay);
39
40      BSP_ledBlueOff();
41
42      /* Block until the next release time.*/
43      vTaskDelayUntil(&xLastWakeTime, xDelay);
44    }
45
46  }
47
```

Blue Blink Task

- Run the code and notice each time a breakpoint is hit, it's the next LED blinking task. All tasks are running.
- When free running the code without breakpoints, notice the color of the LED – It's white.

  This is because we are blinking all three colors at the "same time", producing a white color.

Congrats!


Texas Instruments
Tiva™ C Series
LaunchPad
www.ti.com/launchpad
EK-TM4C123GXL
REV A
TM4C123G
H6PM7
54AVPPW
G4
SELECT
DEVICE
DEBUG
PWR
RESET
WAKE
+3.3V
VBUS
GND
VDDC
TCK TMS TDO TDI EXT TXD RXD
DBG
PA7 PF1
PA6 PE3
PA5 PE2
PB4 PE1
PE5 PD3
PE4 PD2
PB1 TD1
PB0 PD0
PB5 GND
+3.3V VBUS
PF4 PA2
PD7 PA3
PD6 PA4
PC7 PB6
PC6 PB7
PC5 RST
PC4 PE0
PB3 PB2
PF3 PB1
PF2 GND