CMPE 443 PRINCIPLES OF EMBEDDED SYSTEMS DESIGN

PRE LAB #001 Part 1

"Setup Keil and LPC4088"

Motivation

IMPORTANT NOTE: This semester, we will be using three software packages for the lab and course coding submissions. Before starting the custom lab sessions, we will assign you the first prelab session in two parts so that you will have sufficient time to install and get used to working with these environments. This document explains the first part of the first prelab that is due 17/11/2020 16:59.

In this part of the first prelab, you will get introduced with the Keil development environment[1]. Within the scope of this experiment, you will learn:

- setting up and working in a typical embedded software development environment
- investigating simulation and debug environment
- generating output file that can be downloaded to the emulated embedded platform environment which will be described in PRE LAB #001 part 2 that will be announced on Nov. 19th.

1) Keil Environment

Keil is a software that you can program the embedded boards. This software works on Windows Operating systems so you can use virtual operating systems. You can use virtual free environments which are previously mentioned. You can download Windows Operating System from:

https://www.license.boun.edu.tr/

For this course, you need to run Windows and Ubuntu environments simultaneously. So, using virtual environments, you can instantly change the operating systems. The possible virtual free environments:

https://www.virtualbox.org/

[1] "Keil Embedded Development Tools for ARM, Cortex-M ..." <http://www.keil.com/>

https://my.vmware.com/en/web/vmware/downloads/details?downloadGroup=PLAYER-1600&productId=1039&rPId=51984

You can watch the installation videos:

https://www.youtube.com/watch?v=IBuuz9gTcR0

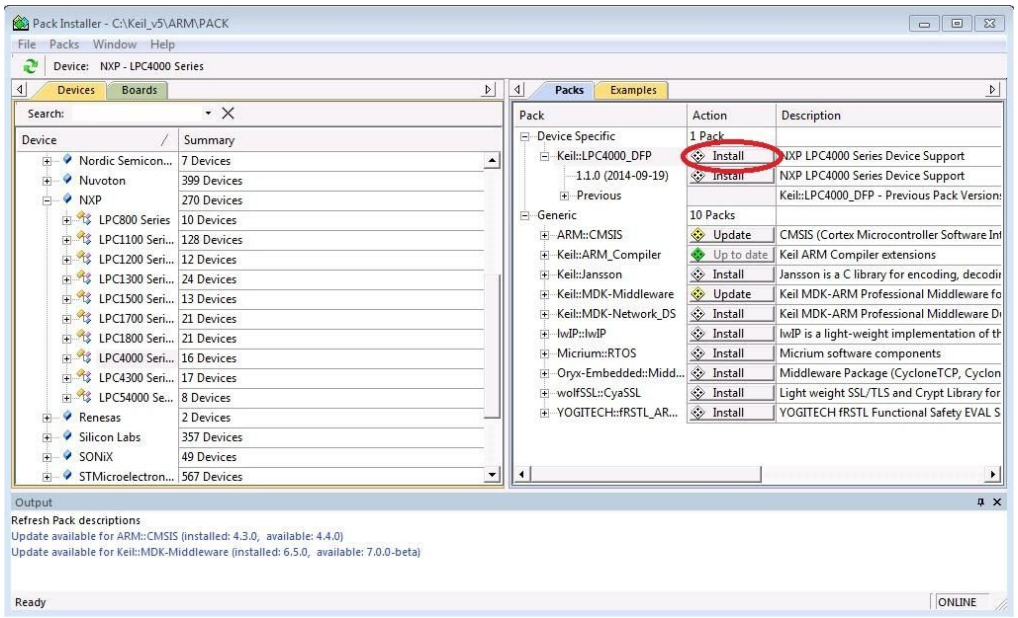
https://www.youtube.com/watch?v=QbmRXJJKsvs

You can download the Keil software from:

From https://www.keil.com/download/product/ download and install Keil uVision5 application (MDK-ARM) on Windows.

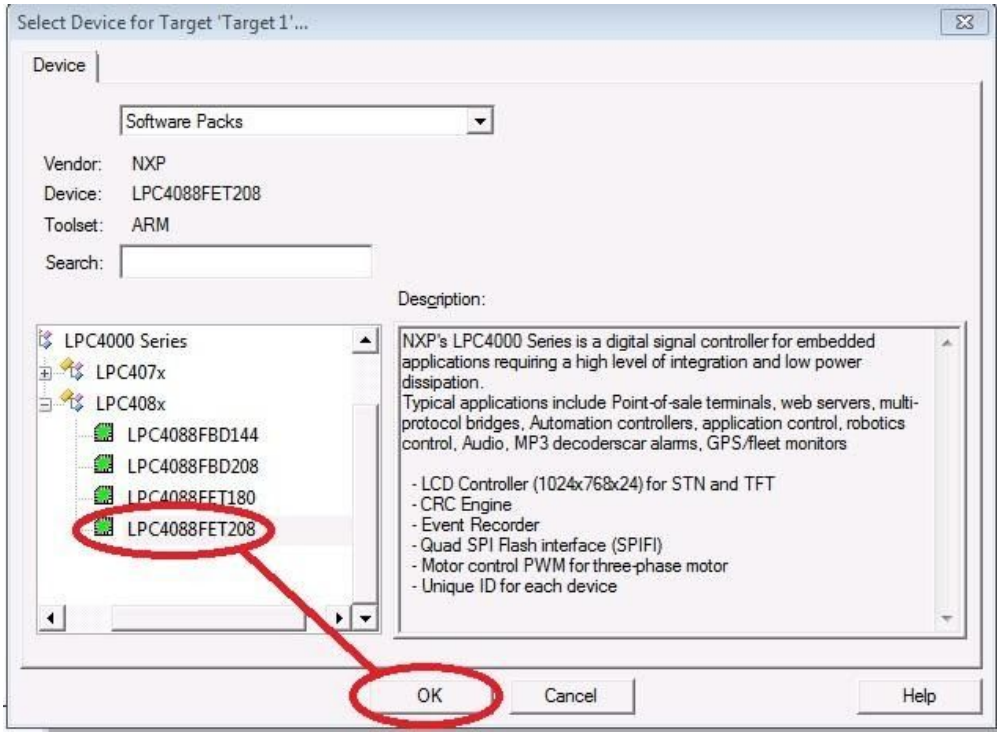
3) Setup Keil Project

Enter Keil uVision5 and open Pack Installer (). Select NXP Devices and select LPC4000_DFP and install that pack.



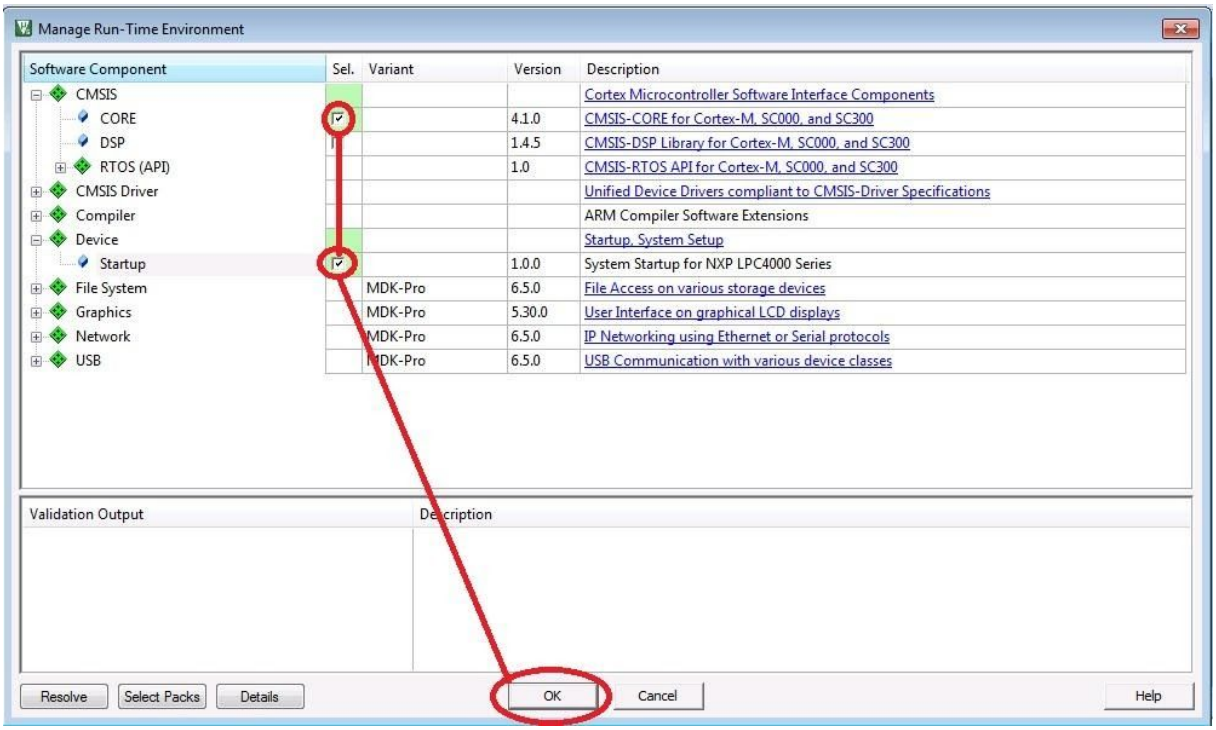
After installation, reload Software Packs using the menu: **Project - Manage - Reload Software Packs.**

For creating project select: **Project - New uVision Project.** (Choose the location of the project and save the project with any name such as LAB_001) After that:



Select **LPC4088FET208** for Device.

After clicking **OK** button, select **CMSIS - CORE** and **Device - Startup.**

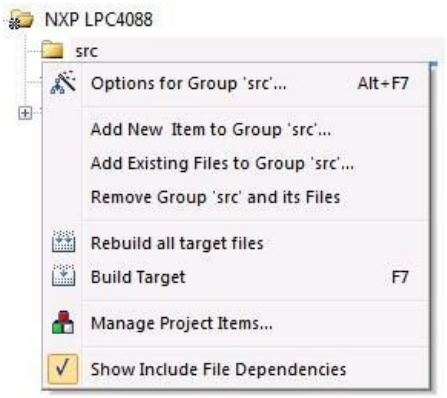


You can change the target name “Target 1” to any name such as “NXP_LPC4088”.

You can change the source group name “Source Group 1” to any name such as “src”.

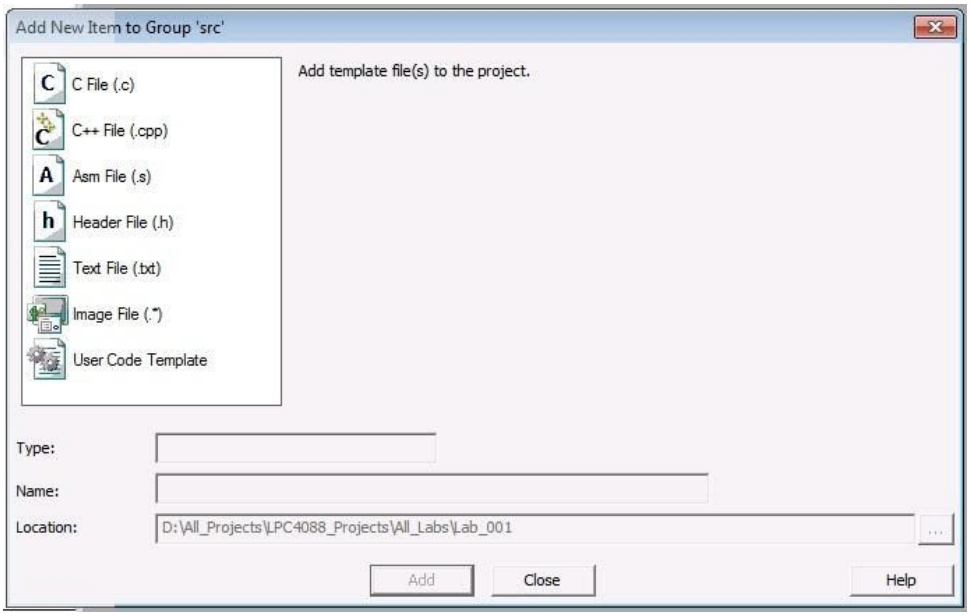
4) Add Source Files to Project

You can add files in several ways to a project. The most common way is to click on a file group in the window **Project** and use the context menu **Add New Item to Group** or **Add Existing Files to Group.**



Add New Item to Group

You can add new items: Right Click the **Source Group - Add New Item – Select File**.



For adding **main.c**: Right Click the **Source Group - Add New Item – Select C File** and name it as main.

In **main.c** file you can copy the following code:

```
#include "LPC407x_8x_177x_8x.h"

int millisecond = 1000;

void init() {

	LPC_GPIO1->CLR = 1 << 18;

	LPC_GPIO1->DIR |= 1 << 18;

	LPC_IOCON->P1_18 &= ~(0x3 << 3);

	LPC_IOCON->P1_18 |= (0 & 0x3) << 3;

}

void update() {

	int i;


	LPC_GPIO1->SET = 1 << 18;

	for(i=0;i<millisecond*2000;i++);
```

```
        LPC_GPIO1->CLR = 1 << 18;
        for(i=0;i<millisecond*2000;i++);
}
int main() {
        init();
        while(1) {
                update();
        }
}
```

Add Existing Files to Group

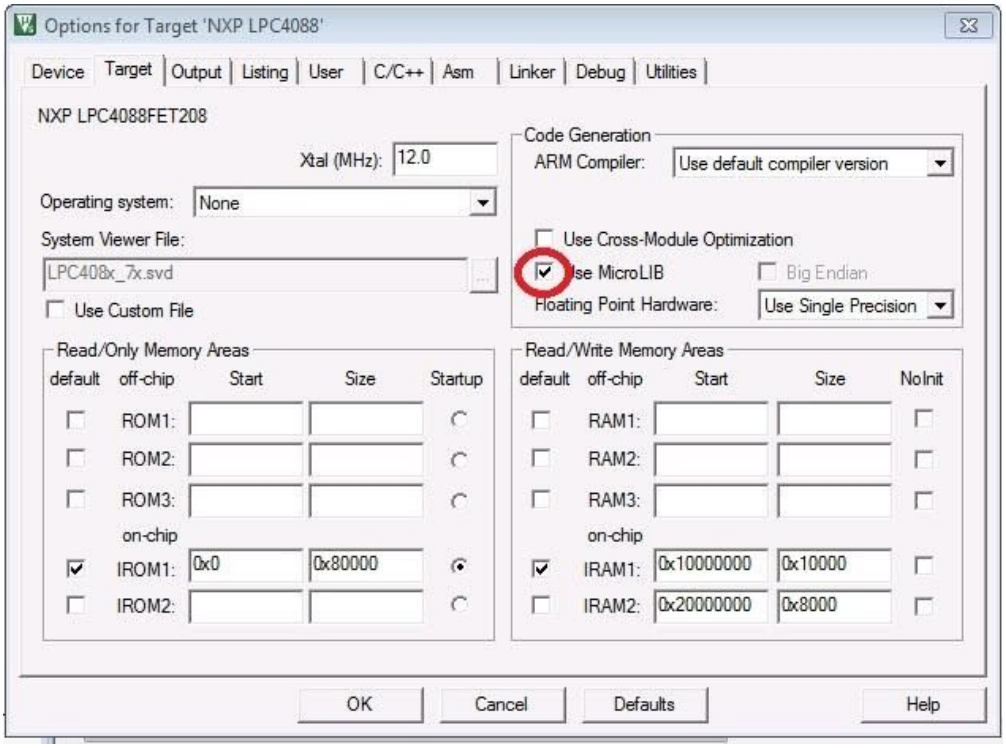
You can add existing items: Right Click the **Source Group - Add Existing Item – Select Files from File Dialog**. Yet, this item is not related with this week's experiment. So, you can skip.

5) Open Existing Project

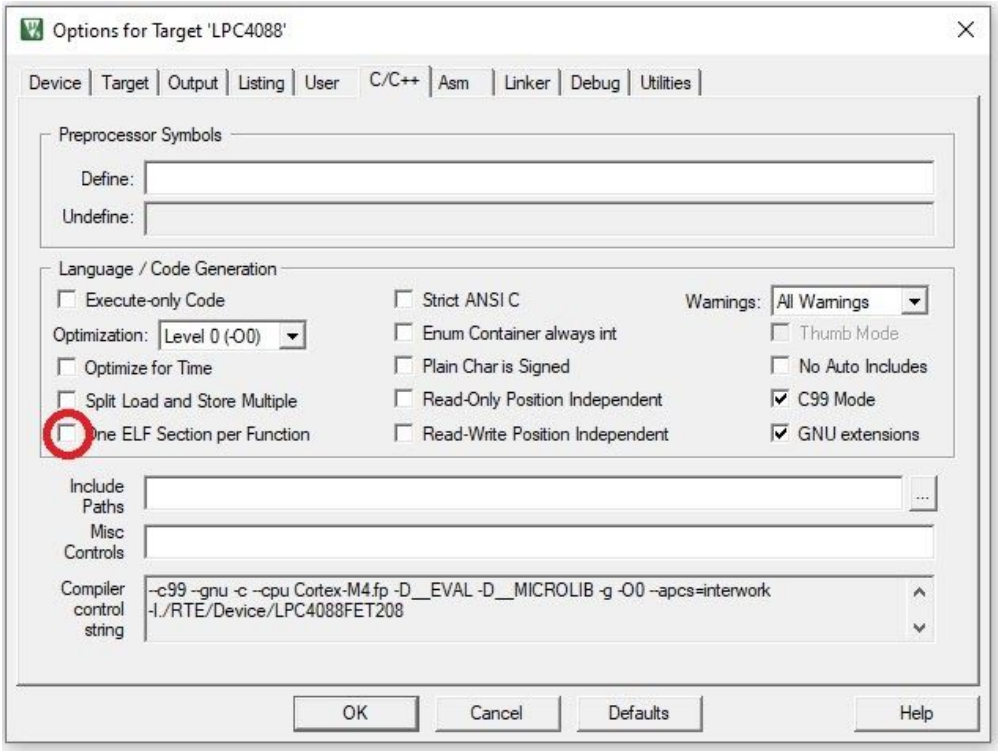
Select **Project - Open Project.** Yet, this item is not related with this week's experiment. So, you can skip.

6) Debug Settings

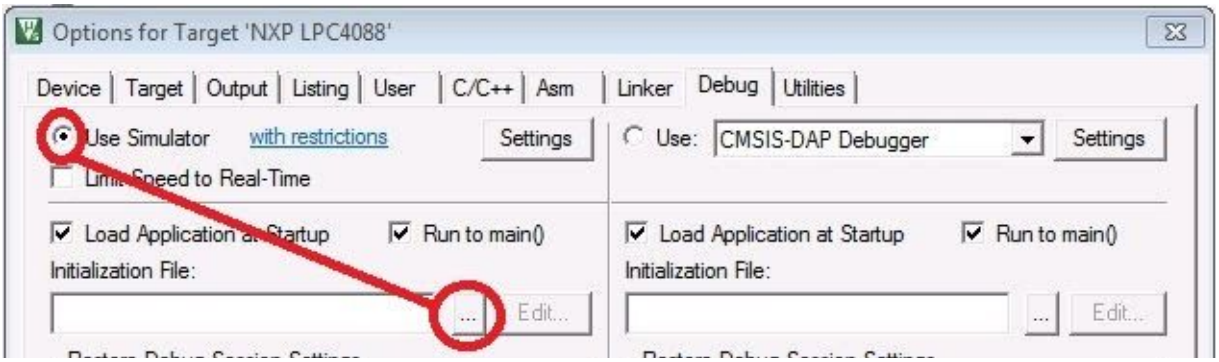
For debugging, select **Target Name - Options for target** 
In **Target** tab, click **use MicroLIB** (You do not need to change others)

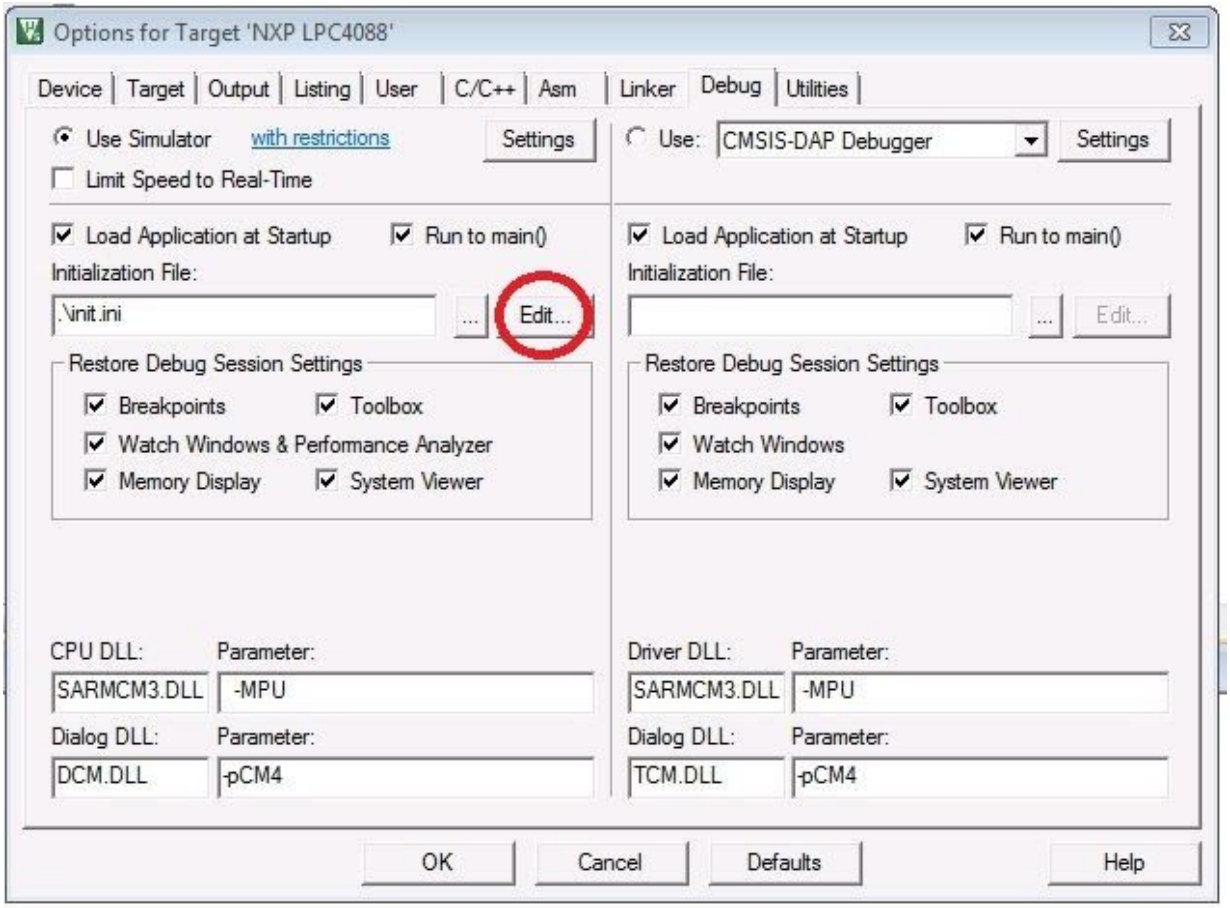


In the **C/C++** tab, uncheck One ELF Section per Function**.**



In **Debug** tab, select **Use Simulator** and click browse **Initialization File** and then write a name and click the Open button.





If the file does not exist, Keil will create it for you if you want. After that, click **Edit.** Then write in the file:

MAP 0x20000000, 0x27FFFFFF READ WRITE

MAP 0x40000000, 0x47FFFFFF READ WRITE

MAP 0xA0000000, 0xA7FFFFFF READ WRITE

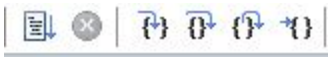
After that, open **system_lpc407x_8x_177x_8x.c** file (Under **Device**) and comment out ("//") 528th, 538th and 546th lines. (1700th, 1710th and 1718th lines in the new version)

```
//while ((LPC_SC->SCS & (1<<6)) == 0);/* Wait for Oscillator to be ready     */
//while (!(LPC_SC->PLL0STAT & (1<<10)));/* Wait for PLOCK0                  */
//while (!(LPC_SC->PLL1STAT & (1<<10)));/* Wait for PLOCK1                  */
```

7) Debug Program

In this section, you will debug your code. Open your **main.c** file from your Project Window. Now you can add **Breakpoints** by clicking next to the line.

Add breakpoint next to: LPC_GPIO1->SET = 1 << 18; and LPC_GPIO1->CLR = 1 << 18;

You can build your code by  . Whenever you change code, you should build the code. After that, you can start debugging by  . You can navigate by  in debug mode. (Every time build your code before loading) Use **Run to Cursor Line** in order to reach breakpoints.

8) Submission

When you build your code in Keil under the **Objects** folder **.axf** is made by Keil. You should submit this file to the PRE1 assignment on the moodle.

The file name for the submission should be:

PRE<exp num>_<StudentID1>_.axf (This will be generated **.axf** file)