Lab 1: Software Introduction and TivaC LaunchPad Programming

Name: Syed Muhammad Ghufran Alvi ID: sa08235 Section: T1

Objective

To download, install and setup the required software(s) and create C-Program for ARM Cortex M4F TivaC LaunchPad using Kiel and Energia.

Pre-Lab Tasks

- **Task 1:** Download the file "Pre Lab.rar" from LMS module of Lab 01. Extract the folder containing all the files.

In-Lab Tasks

Task 2: Install required software.
Task 3: C-Program for TivaC Launchpad Board with Keil uVision.
Task 4: C-Program on Keil for Push Button.
Task 5: C-Program for TivaC Launchpad Board with Energia IDE.

Task 2: Download and install required software

Introduction to Keil uVision

Keil μVision is a software which solves many of the pain points for an embedded program developer. This software is an integrated development environment (IDE), which integrates a text editor to write programs, a compiler and it will convert your source code to hex files too. Here is simple guide to start working with Keil μVision which can be used for:

- Writing programs in C/C++ or Assembly language
- Compiling and Assembling Programs
- Debugging program
- Creating Hex and Axf file
- Testing your program without available real Hardware (Simulator Mode)

Step By Step Guide to install Keil

You can skip to step 5 of this section if you have completed the pre-lab task of downloading the files.

1. To download the setup file, go to the following link: `https://www.keil.com/download/product/` Verify that your computer at least meets the minimum system requirements before you install any Keil software development product.

System Requirements

Minimum Hardware Requirements
1 GHz 32-bit or 64-bit processor, 1 GB of system memory (RAM), 2 GB of available disk space

Recommended Hardware Requirements
2+ GHz 64-bit processor, 4 GB of system memory (RAM), 5 GB of available disk space, 2 Mpbs or higher Internet connection for software activation, product updates and online services like the PackInstaller

Supported Operating Systems 32-bit and 64-bit variants of Microsoft Windows are supported:
Windows 10/11 Home, Windows 10/11 Pro, Windows 10/11 Enterprise, Windows 10/11 Education
Windows 8.1, Windows 8.1 Pro, Windows 8.1 Enterprise, Windows 8, Windows 8 Pro, Windows 8 Enterprise

2. Select MDK-ARM as shown in Figure 1.

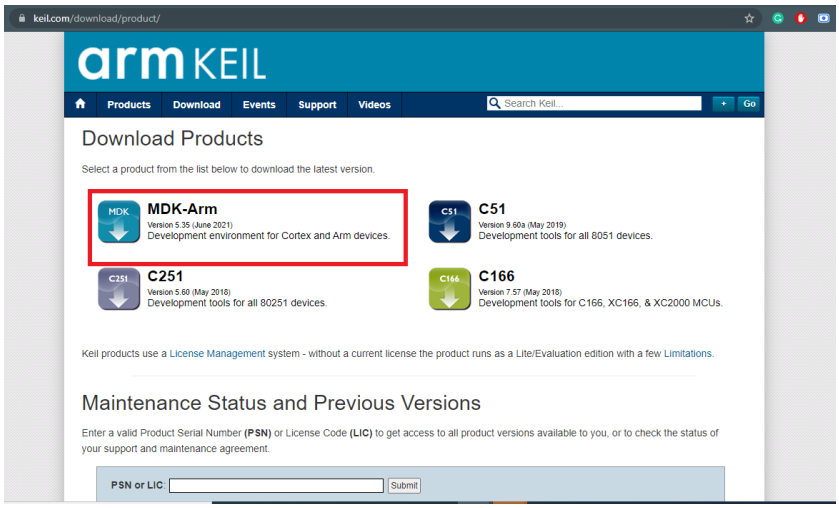


Figure 1

3. This will bring up a form as shown in Figure 2. Fill out the form and then click on the submit button, at the bottom of the page.

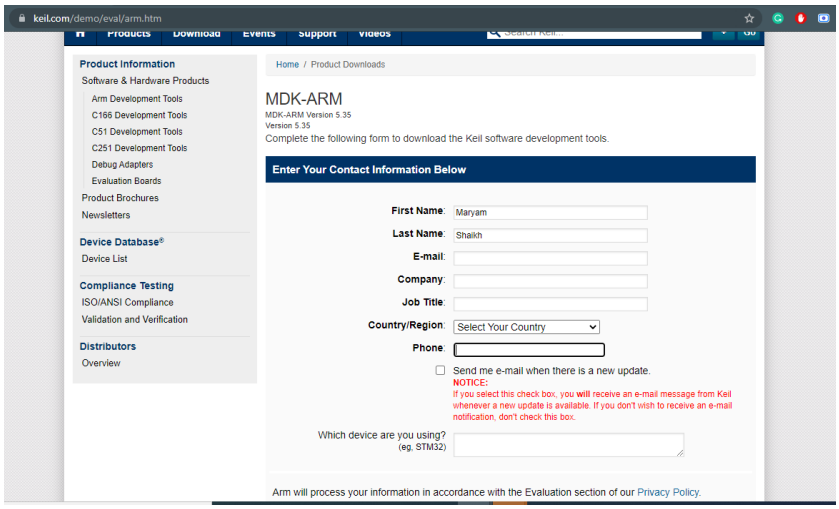


Figure 2

4. Next click on the MDKxxx.EXE link to download the setup executable file as shown in Figure 3.

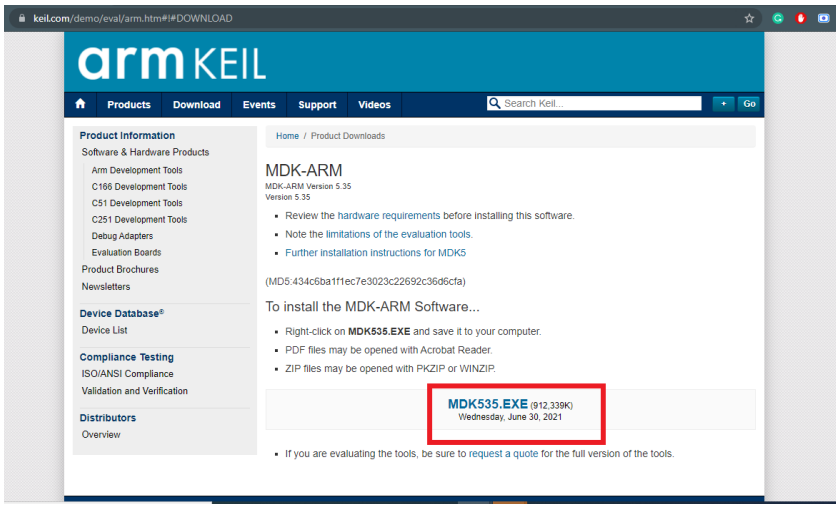


Figure 3

5. Execute the downloaded file named "MDK537.exe". A window pops up as shown in Figure 4. Installing the software is straight forward; choose the Next button in each step.

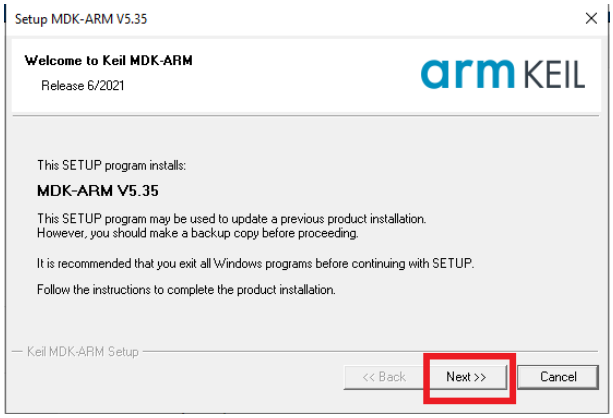


Figure 4

6. After successful installation, a window shown in Figure 5 pops up. Just click on Finish button.

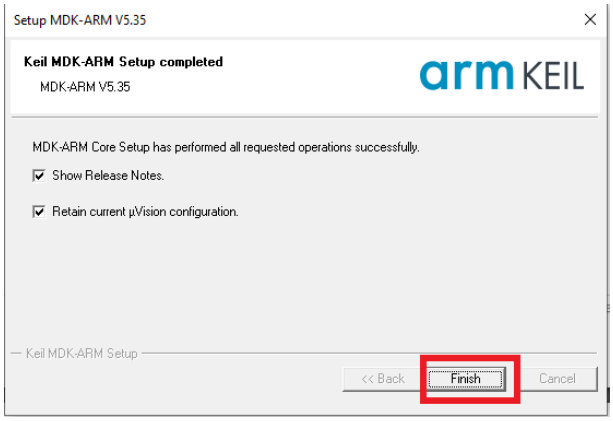


Figure 5

Next, we would install TM4C123 support packages for our board.

Installing the TM4C123 packages using Pack Installer

1. When the Keil installation finishes and you press the Finish button, the Pack Installer software opens automatically. In the pack installer, you can install the packages for different Arm microcontrollers. When the pack installer opens, it downloads some packages. Wait until the download of the packages finishes.

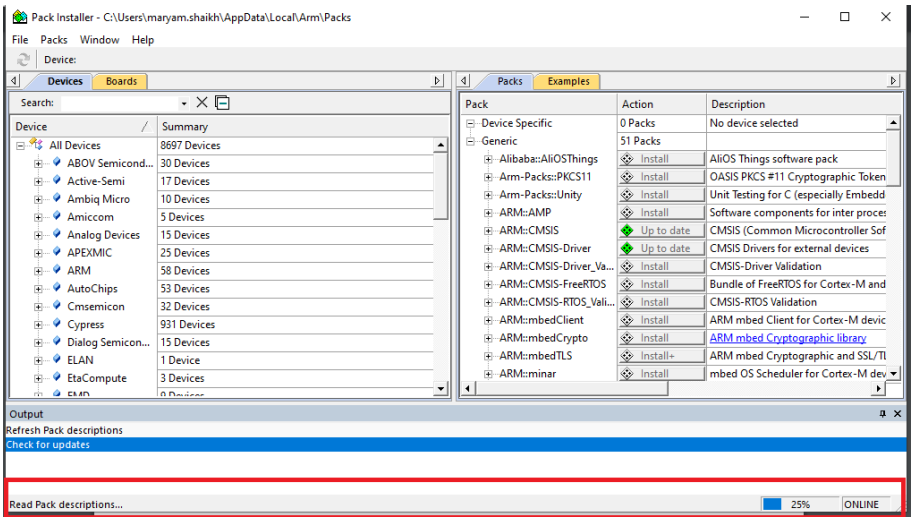


Figure 6

2. Now, In your search bar, write " TM4C123GH6PM " and follow the sequence showed in the picture. Wait until the installation finishes and the Install button changes to "Up to date". check if there is any package which need to be updated. If so, Update that aswell. Now, close the Pack Installer.

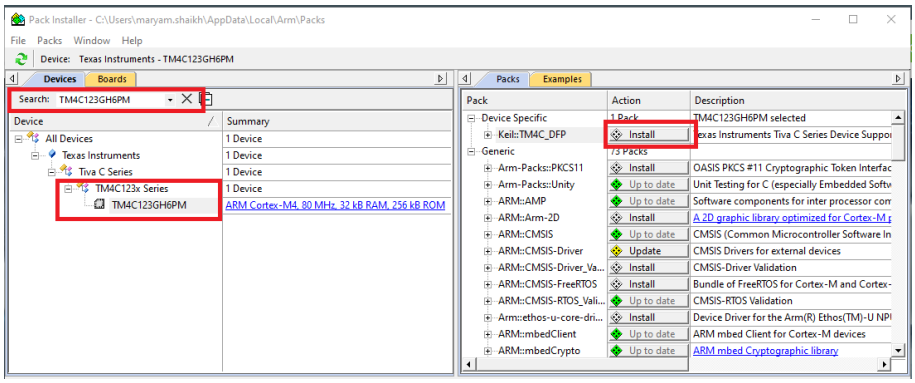


Figure 7

Installation of Tiva Ware for C-series "SW-TM4C"

You can skip to step 3 of this section if you have completed the pre-lab task of downloading the files.

1. Go the website `https://www.ti.com/tool/SW-TM4C` and click on the download option and download the full version. You may be required to Make your account on the website, Register yourself and make an account then continue with the download as shown in figure below.

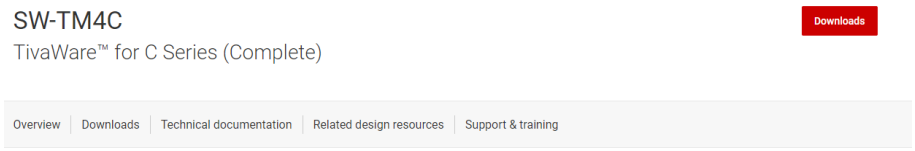


Figure 8

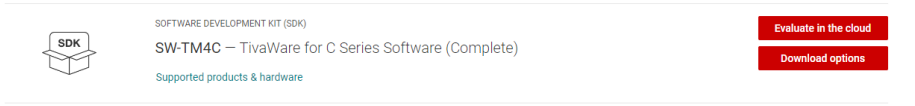


Figure 9

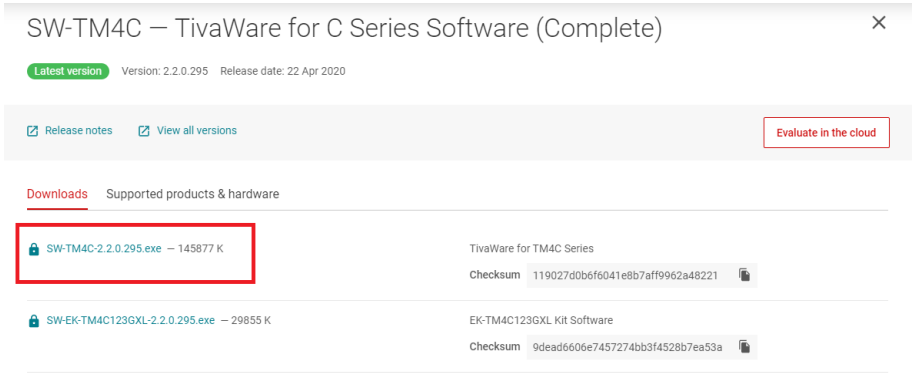


Figure 10

2. Fill out the required form with your information, once the request is approved, click on the download option.

3. Run the SW-TM4C-x.x.x.xxx.exe file and follow the steps by clicking next.
4. Click on Finish to complete the installation.

Figure 11

Installation of Stellaris ICDI Drivers

You can skip to step 2 of this section if you have completed the pre-lab task of downloading the files.

1. Go to the website: `https://www.ti.com/tool/STELLARIS_ICDI_DRIVERS` and click on the download option and download the full version.

Figure 12

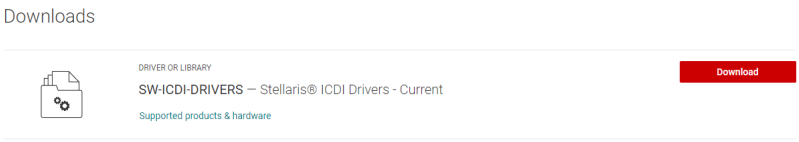


Figure 13

2. Now, extract the drivers file named ”spmc016a.zip”. Connect your TivaC LaunchPad to the computer. Go to device manager, go to “Others” and install the drivers by adding the path of the drivers folder that you extracted named ”stellaris_icdi_drivers”. Check the device manager you will be able to see it installed as shown below.

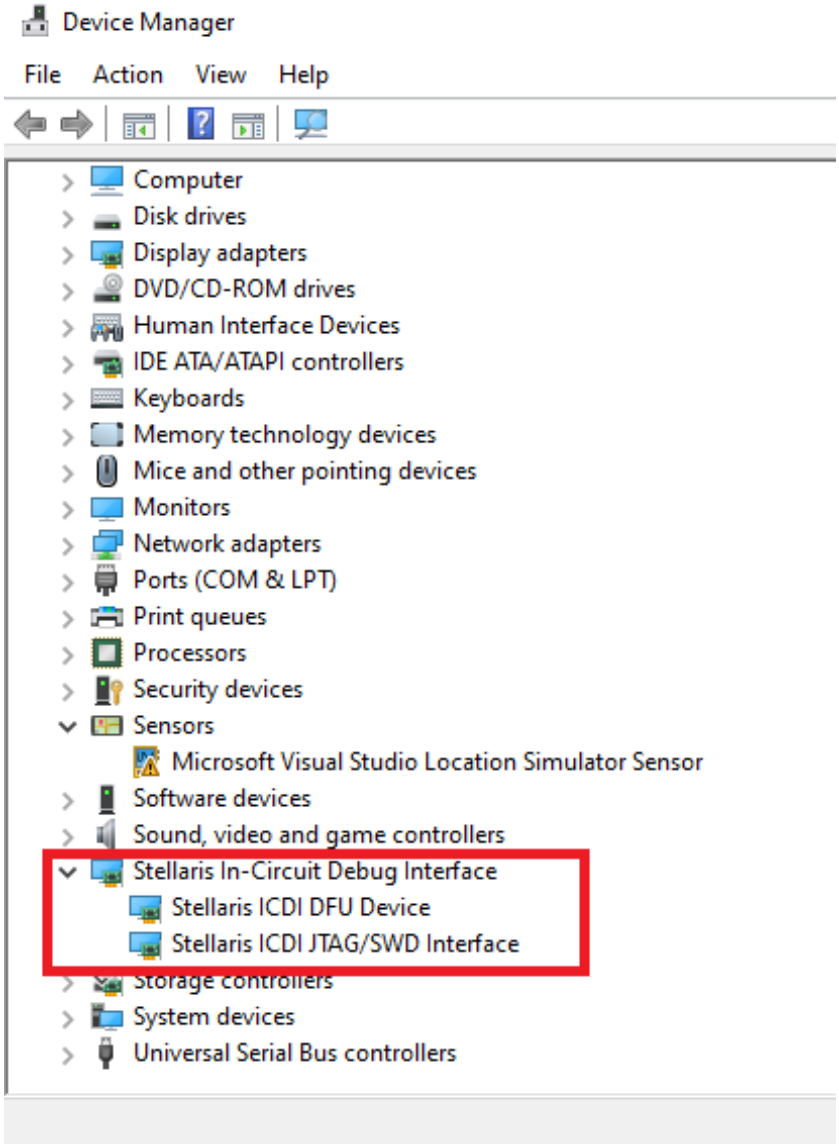


Figure 14

3. Go to “Ports“ in device manager and right-click and update your COM port with the same driver ”spmc016a.zip” file.

4. Once its done the name should be automatically updated as follow:

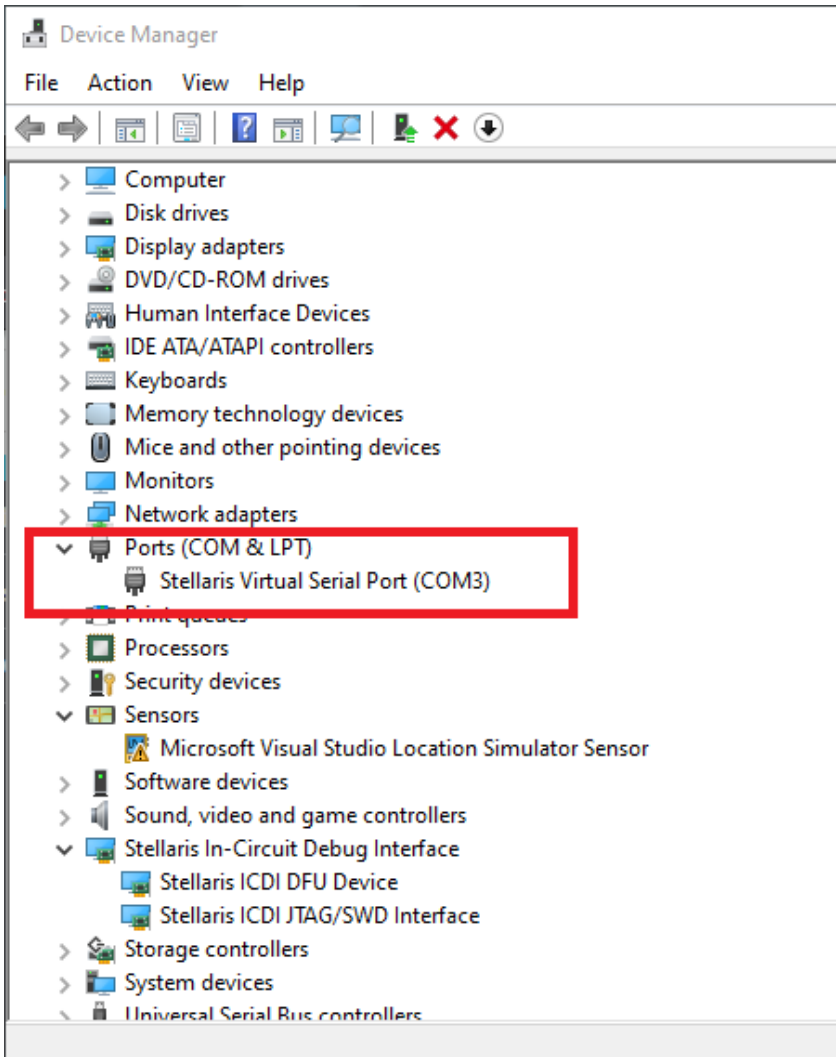


Figure 15

Installation of Stellaris ICDI Addon in Keil

1. Go to the website: `https://developer.arm.com/documentation/ka002280/latest`. If already downloaded in pre-lab, then run the file "MDK Stellaris ICDI AddOn.exe". Installing the software is straight forward; choose the Next button in each step.

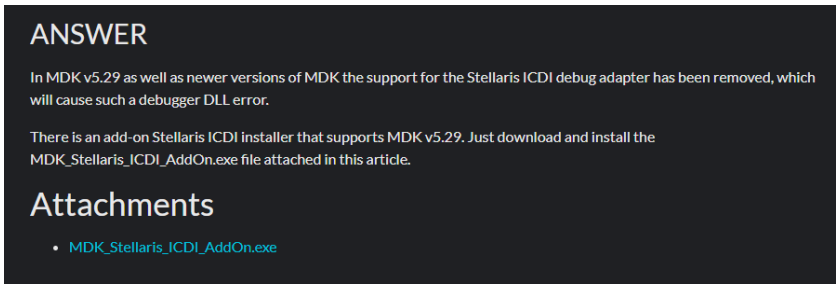


Figure 16

2. Click on Finish to complete the installation.

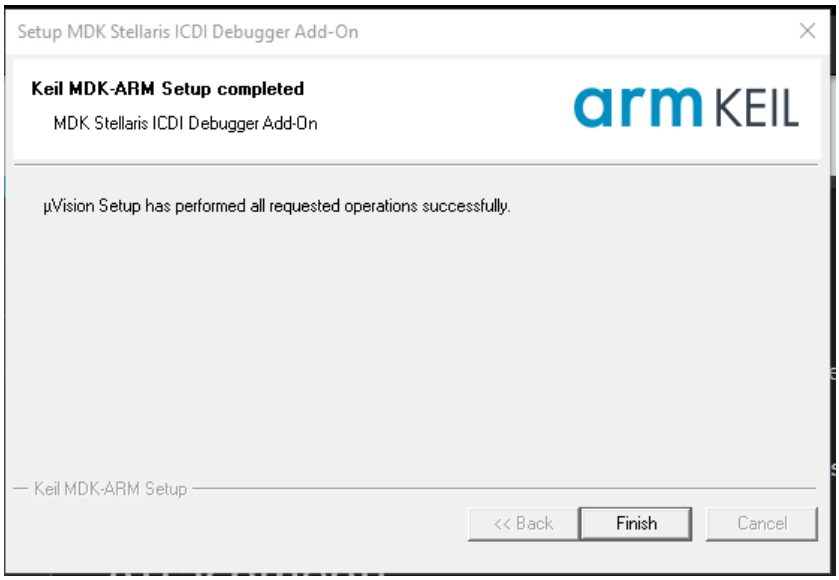


Figure 17

Installation of Energia IDE for LaunchPad

1. Go to the website `https://energia.nu/download/` to download the file or used the one already provided in pre-lab folder named "energia-1.8.10E23-windows".

2. Extract the file and Run the "energia.exe" to see if it opens correctly.

3. Follow this link to install relevant drivers: `https://energia.nu/guide/install/windows/` or directly extract "energia_drivers.zip" in pre lab on desktop.

4. Open the drivers folder extracted and install for your 32 or 64 bit system accordingly.

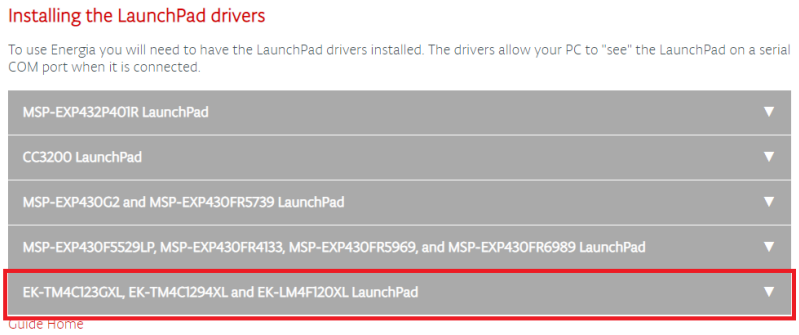


Figure 18

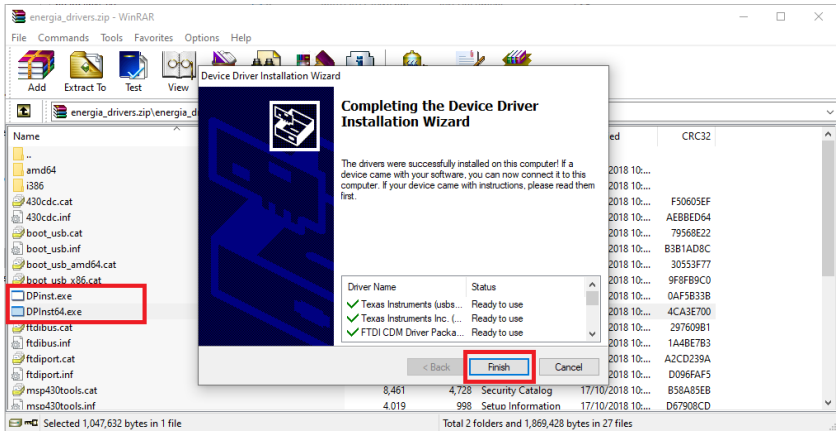


Figure 19

5. Open your Energia IDE, go to board manager in energia - Type " TivaC " and click on the install tab for mentioned result below.

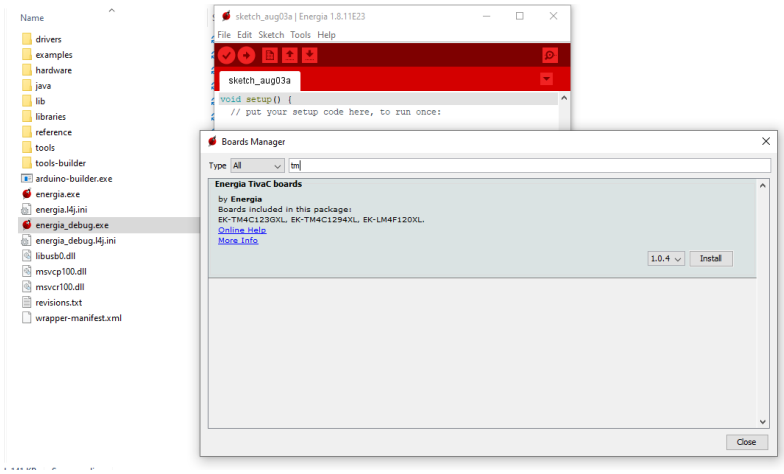


Figure 20

Installation of Terminal Program

1. Go to the website `http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe` to download the file or directly use the pre-lab file named "putty.exe".

2. You can use it directly via single click in the icon.

Introduction to the ARM Cortex M4(F) Processor

The ARM Cortex-M is a group of 32-bit RISC ARM processor cores licensed by ARM Holdings. The cores are intended for microcontroller use, and consist of the Cortex-M0, M0+, M1, M3, M4, and M7.

The Arm Cortex-M4 processor is a highly-efficient embedded processor, developed to address digital signal control markets that demand an efficient, easy-to-use blend of control and signal processing capabilities. The combination of high-efficiency signal processing functionality with the low-power, low cost and ease-of-use benefits of the Cortex-M family of processors satisfies many markets. These industries include motor control, automotive, power management, embedded audio and industrial automation markets.

Cortex-M4F microcontrollers are available from a number of semiconductor manufacturers such as Texas Instruments, Infineon, Atmel, NXP (formerly Philips), Analog Devices, Toshiba, STMicroelectronics, and more.

EK-TM4C123GXL - TivaC LaunchPad Development Board

The TM4C123G LaunchPad Evaluation Kit is a low-cost evaluation platform for Arm Cortex-M4F based microcontrollers. Featuring a 80-MHz Arm Cortex-M4F CPU, 256kB of flash, and 32kB of SRAM, the TM4C123GH6PM MCU provides integrated USB 2.0 support for USB Host/Device/OTG and two 12-bit ADC modules. The TM4C123GH6PM also includes a multitude of serial communication channels such as UART, SPI, I2C, and CAN. The design of the TM4C123G LaunchPad highlights the TM4C123GH6PM USB 2.0 device interface and additional device features such as the hibernation and PWM modules. Software libraries are available that allow users to program the chip using the energia IDE and Kiel uVision.

Powering your TivaC Launchpad

The Tiva C Series LaunchPad can be powered from one of two power sources:

1. On-board ICDI USB cable (Debug, Default)
2. USB device cable (Device)

The POWER SELECT switch (SW3) is used to select one of the two power sources. Select only one source at a time.

Input/Output Pins Configuration

TM4C123GH6PM is a 64-pin micro with more than 40 usable GPIO pins spreaded in six GPIO ports such as PORTA, PORTB, PORTC, PORTD, PORTE and PORTF. Except a few GPIOs (PB0, PB1, PD4 and PD5) all GPIO pins are 5V tolerant. All pins have internal input pull-up and pull-down resistors. Any pin can be used for external interrupt and ADC trigger. Pins can be configured as GPIOs or as AFIOs for other special purpose as follow:

1. Serial ports – receive and transmit data via the UART protocol
2. I2C ports – two-wire communication via the IIC protocol
3. SPI – serial communication
4. PWM for pulse generation
5. Built-in LED connected to PORTF in pins 1,2, and 3 for red ,blue and green LEDs.

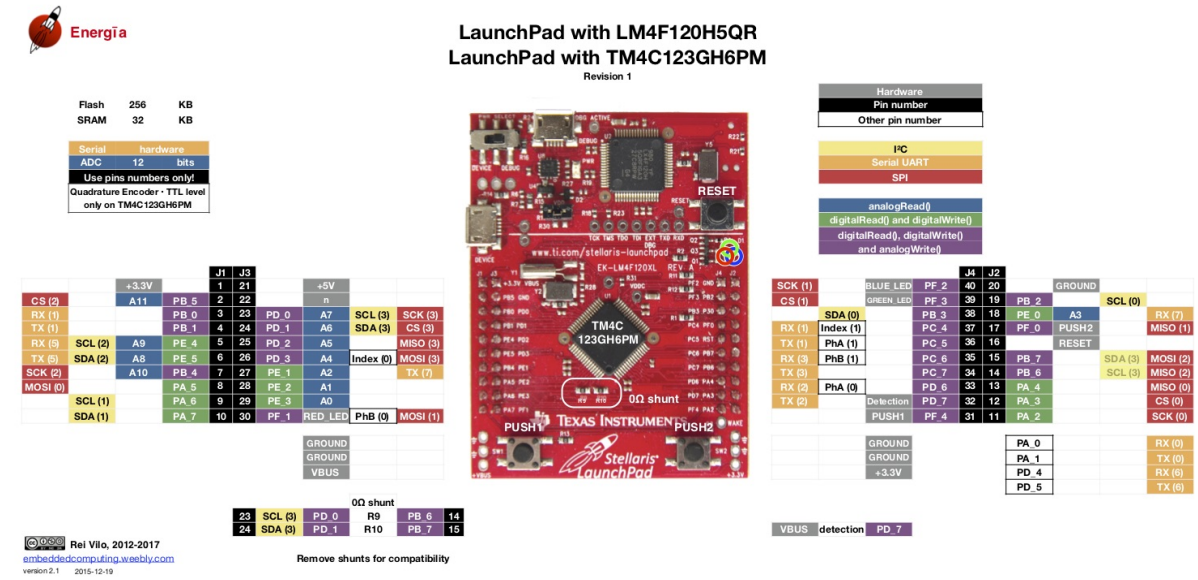


Figure 21: TivaC LaunchPad Pin Configuration

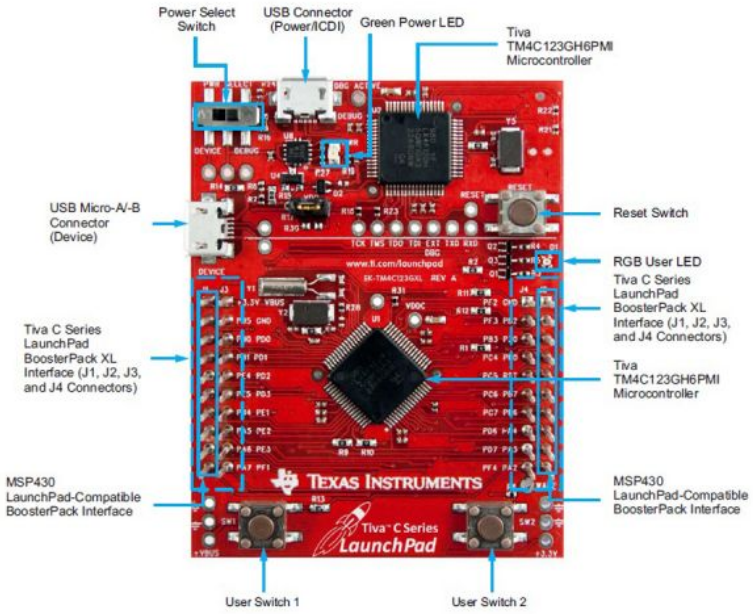


Figure 22: TivaC LaunchPad Board Features

Point to remember

1. Never exceed input logic high voltage of any pin beyond VDD limit unless you are sure that the source will not exceed the 5V limit.
2. Don't use negative input voltages with any pin. Be sure of polarity.
3. Don't stress any GPIO pin beyond 10 – 15mA, although the max limit is 25mA. Use external switching devices like opto-isolators, FETs and BJTs to drive high power loads.

Programming the Board

The LaunchPad provides an easy way to test the capabilities and features of the microcontroller as well as a means to easily program and debug the microcontroller with JTAG through a simple USB connection. This allows for quick prototype development and testing, which is a common use of evaluation boards of this nature.

Point to remember

Before programming, make sure the "Power Select Switch" is in "DEBUG" mode and micro USB is connected with USB Connector(Power/ICDI) port as shown in Fig. 22.

Potential Board Applications

1. Rapid Prototyping and CRC Calculations
2. Robotics and Drone Controllers
3. Consumer Products

Task 3: Creating a C-language Program in Keil uVision

Open the Keil IDE by clicking on its icon on the desktop. Choose New uVision Project from the Project menu as shown in Fig. 23.

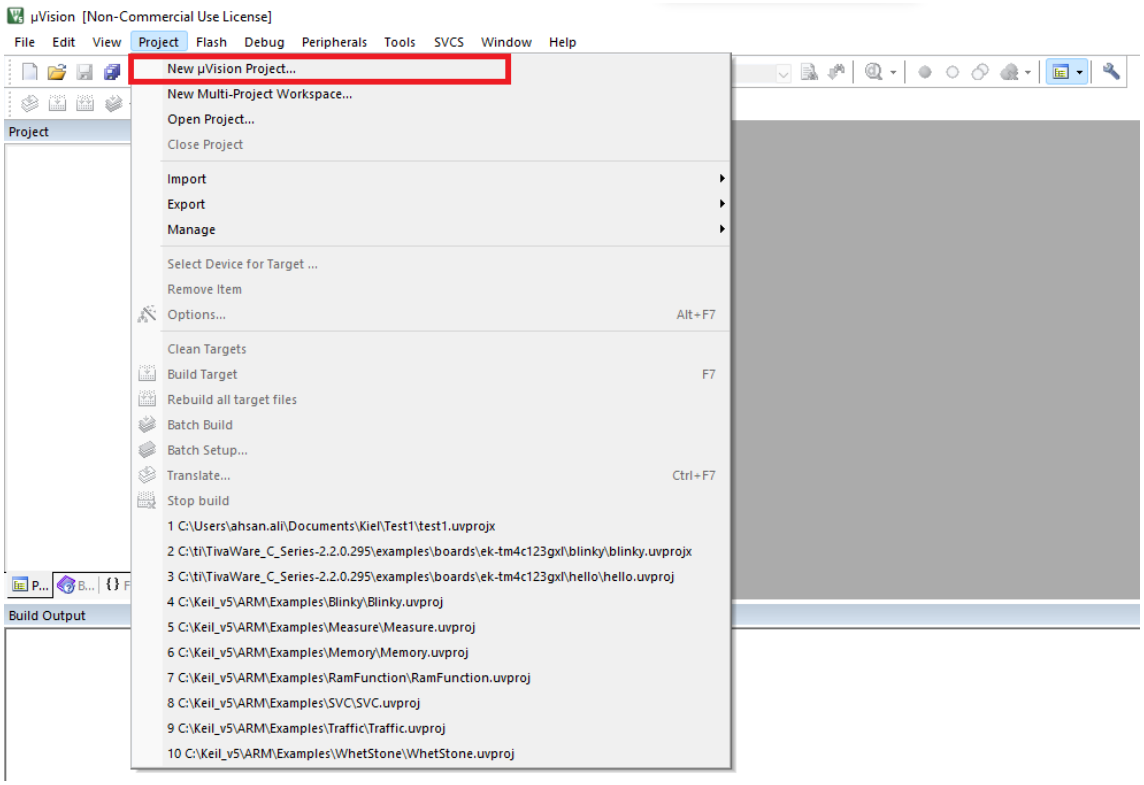


Figure 23

Create a new folder and Name it ”newProject”. Type the name ”newProject” for the project name and click Save.

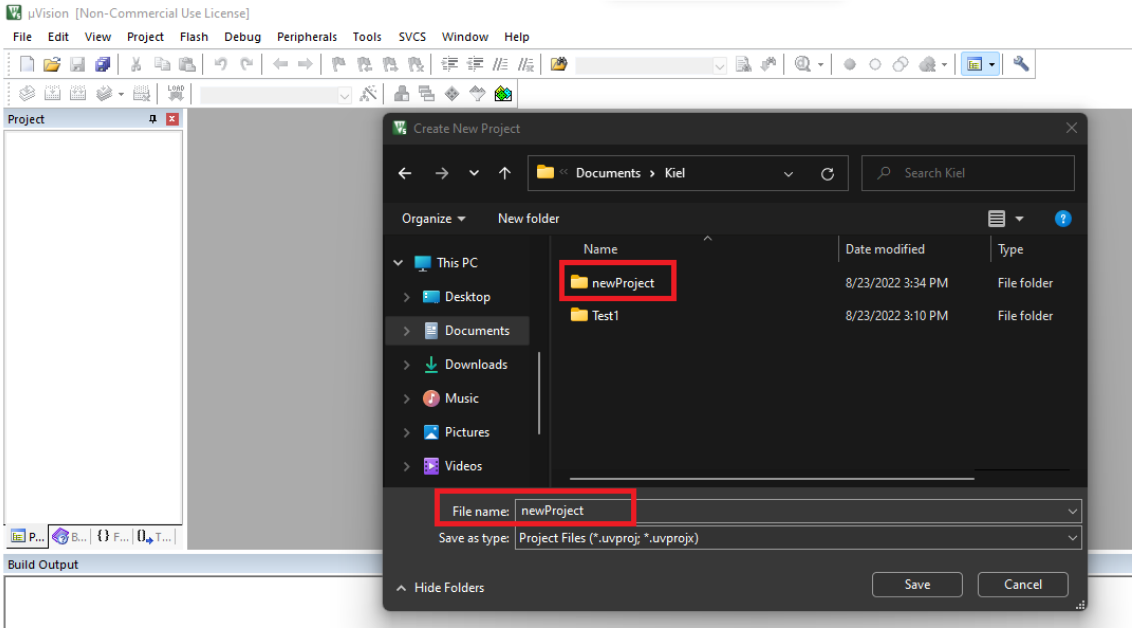


Figure 24

In the Device menu, Search for ”TM4C123GH6PM”, select it and click OK.

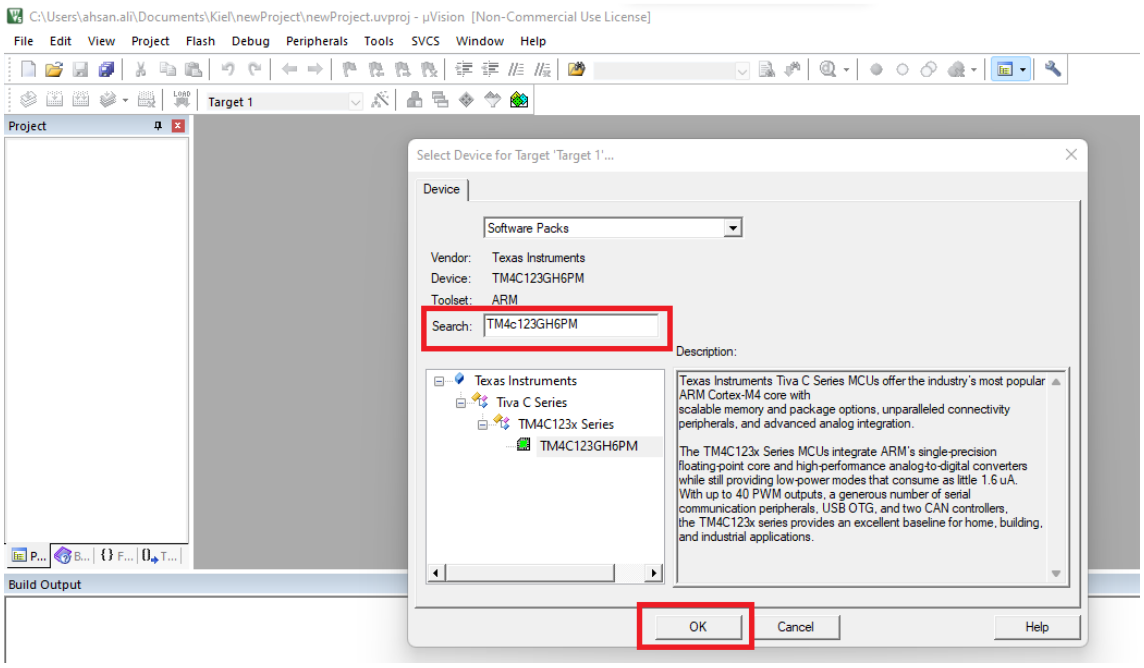


Figure 25

From the software component tree click on Device and add the Startup file by clicking the checkbox next to Startup. Similarly, for CMSIS, check the "CORE" as shown in Fig. 26. Then, click on the OK button.

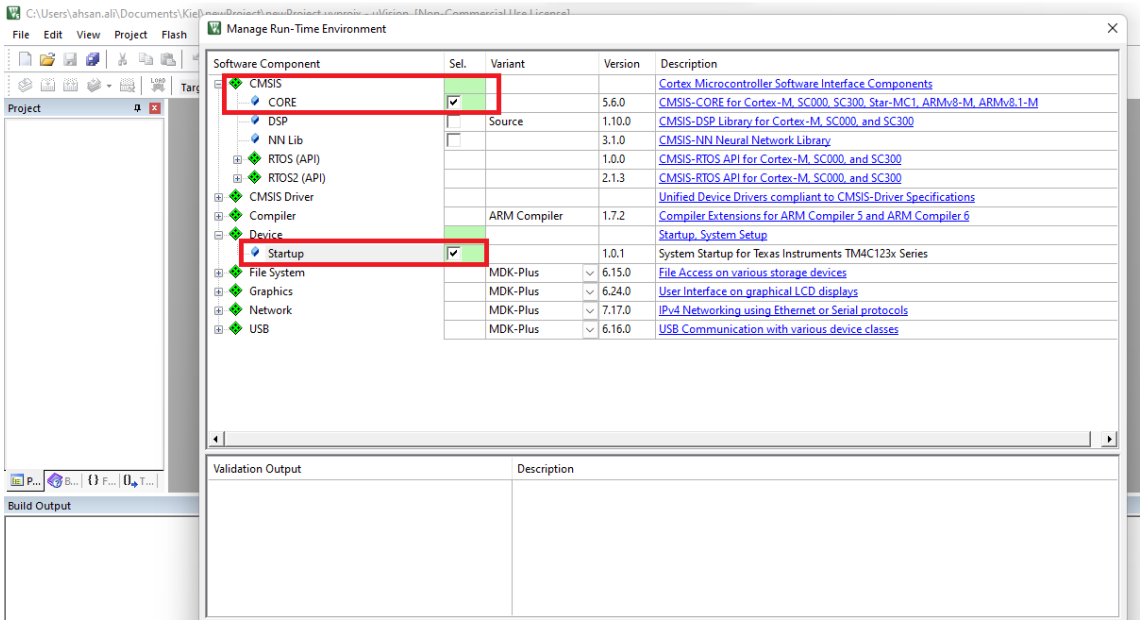


Figure 26

Right click on **Source Group 1** and choose **Add New Item to Group**. This makes a new file and adds it to the project.

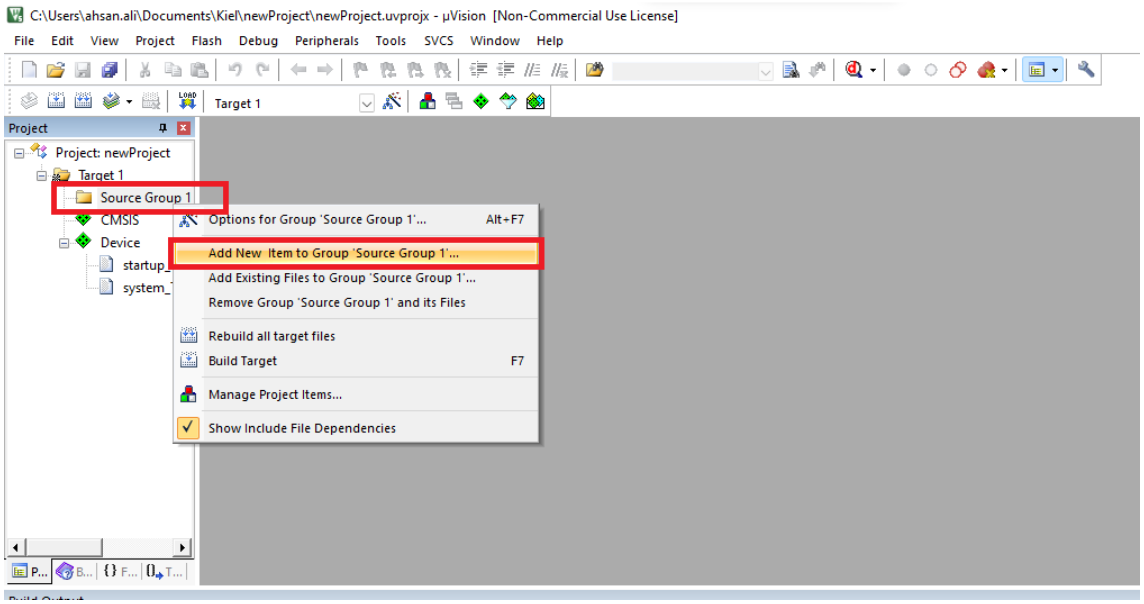


Figure 27

Choose the type of file as C File(.c) and name it as "myProject". Click on the Add button and then click on Close.

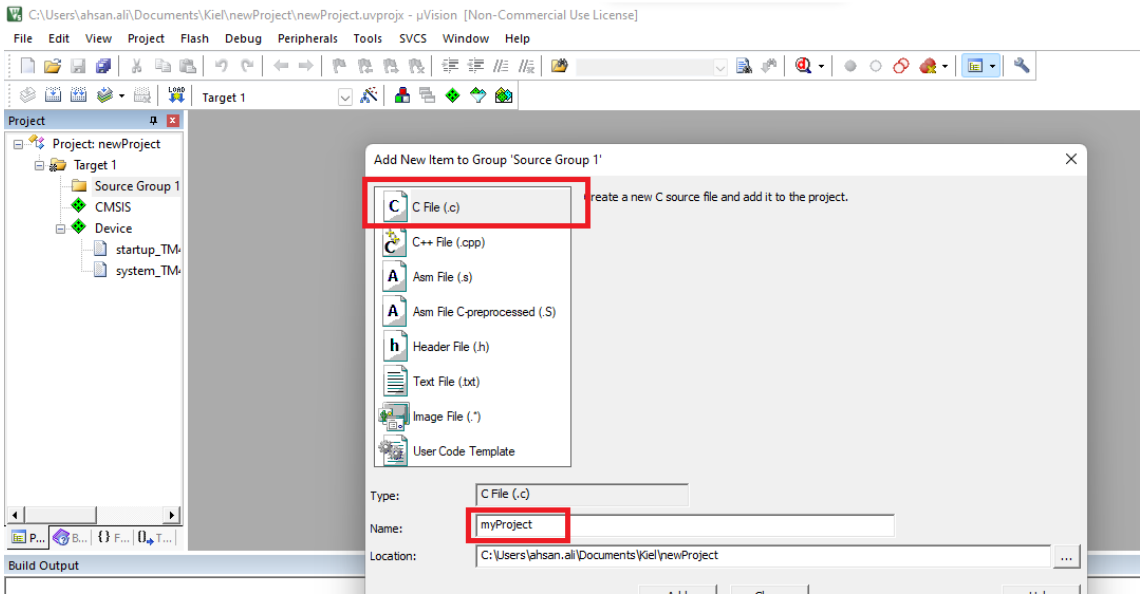


Figure 28

Find the "LED_ BLINK.txt" file on LMS Resources under Lab2 Module and copy its code into editor. Press Ctrl+S to save the file. To compile click on the Build icon or press F7. Connect the TivaC LaunchPad with your machine using UART USB cable. Once done, set the "Options for Target" as shown in Fig. 29 and Fig. 30 and press OK.

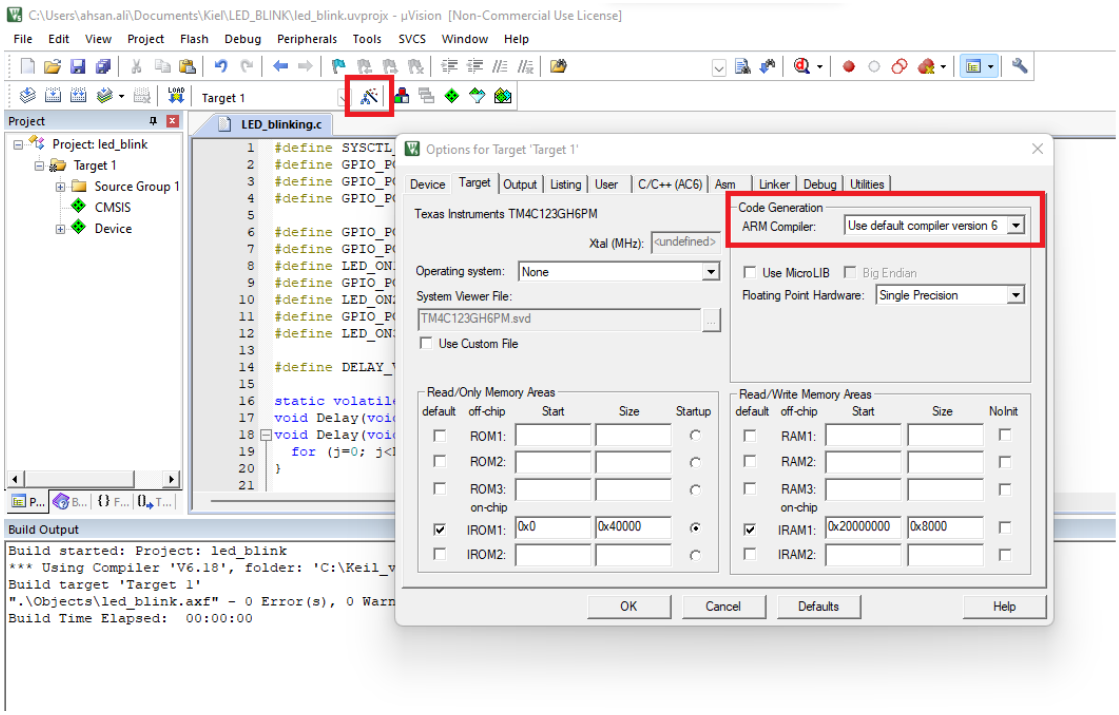


Figure 29

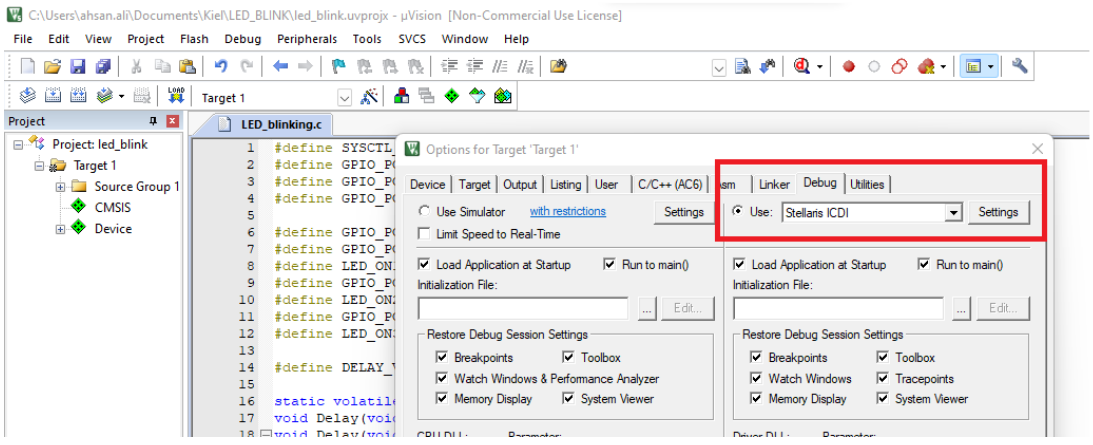


Figure 30

Once the "Options for Target" is set and the Board is connected, press the "Download" button as shown in Fig. 31 to deploy the code on the TivaC LaunchPad. If successfully downloaded, the message will be displayed as shown in Fig. 31.

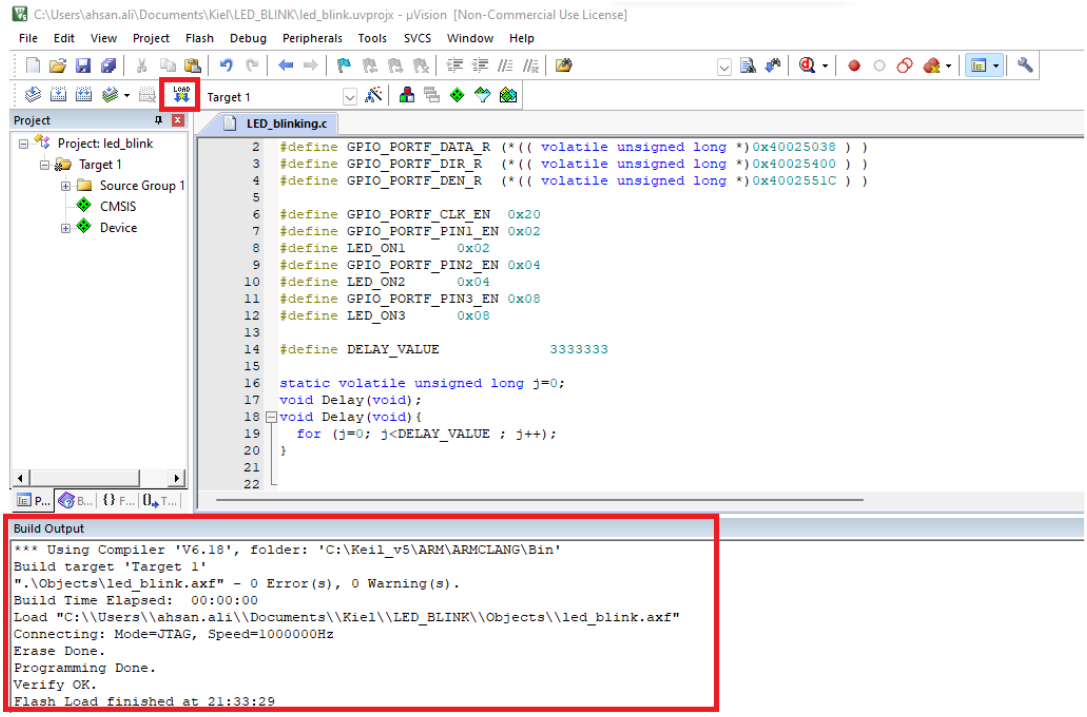
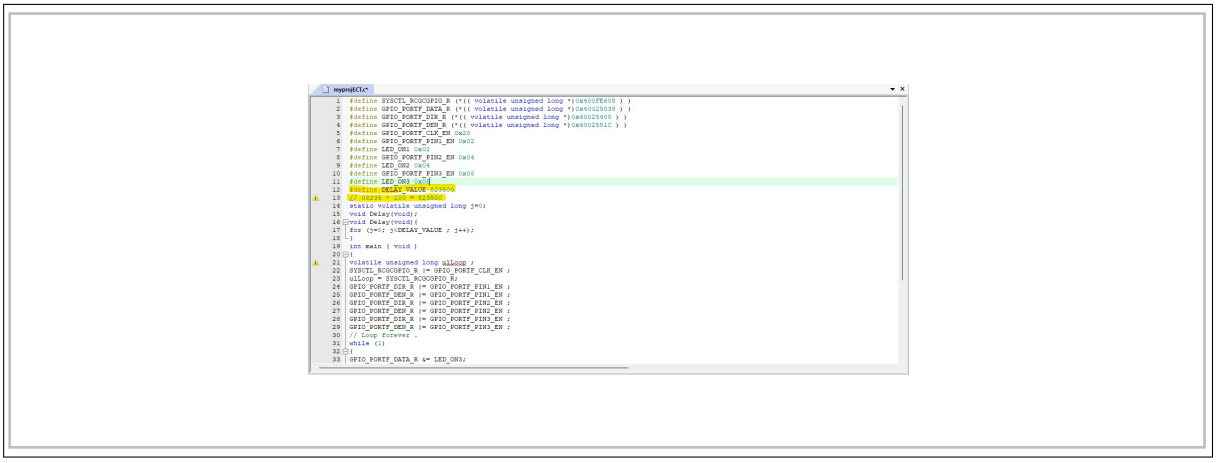


Figure 31

Now, in order to observe the RGB Built-in LED on the board, press the ”RESET” button on the board. Once RGB starts blinking, perform the following steps:

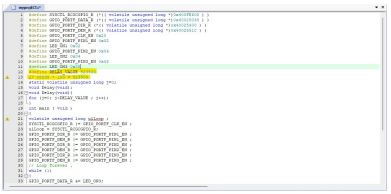
1 Go through the code and try changing the ”DELAY_ VALUE” with factor of 100 multiplied with your ID digits i.e. 100*01234 = 123400.

2 Repeat the steps of Building and Downloading the code and Resetting the Board. Attach the screenshot of the Code with you ID clearly visible and program downloaded in Build Output window in below window:



3 Try increasing and decreasing the ”DELAY_ VALUE” and repeating the steps of build, download and reset. State your observation for ”DELAY_ VALUE” below:

4 Attach the image of the RGB LEDs turning on after pressing ”RESET” button on board below:



Task 4: C-Program on Keil for Push Button

For this task, you are required to create your own new project in similar manner you did for previous task and use the file ”PUSH_ BUTTON.txt” to test the functionality of Push Button SW1 on TivaC board located at bottom-left corner of board.

1 State your observation for usage of push button SW1 on TivaC board, what does it achieve:

A push button on the LaunchPad is a simple mechanical switch that closes the switch as far as the push button is pressed.

2 Research and provide an explanation for ”Debounce in Push Buttons”. Go through the code provided in file ”PUSH_ BUTTON.txt” and explain how the function ”Delay()” is used to cater debouncing:

To understand what the function DELAY DE-BOUNCE, we need to understand that the input parameter to the function. The input is nothing but system's Clock Frequency divided by 1000. The frequency is defined through a variable in the start of the code only, which is 16000000. Now the function, itself, is simply counting from 0 to the input passed to the function as input from the main. Creating a delay as the system is not doing anything, practically, yet giving us a delay in real life.

Task 5: Program the TivaC Launchpad Board with Energia IDE

Open the Energia IDE and and select example code "Fade" as shown in Fig. 32. Make sure the Board is selected to be "LaunchPad (TivaC) w/tm4c123 (80MHz)" as shown in Fig. 33. Compile the code as shown in Fig. 34. And finally upload the code to board as shown in Fig. 35. You would now be able to observe the fading GREEN LED on the board.

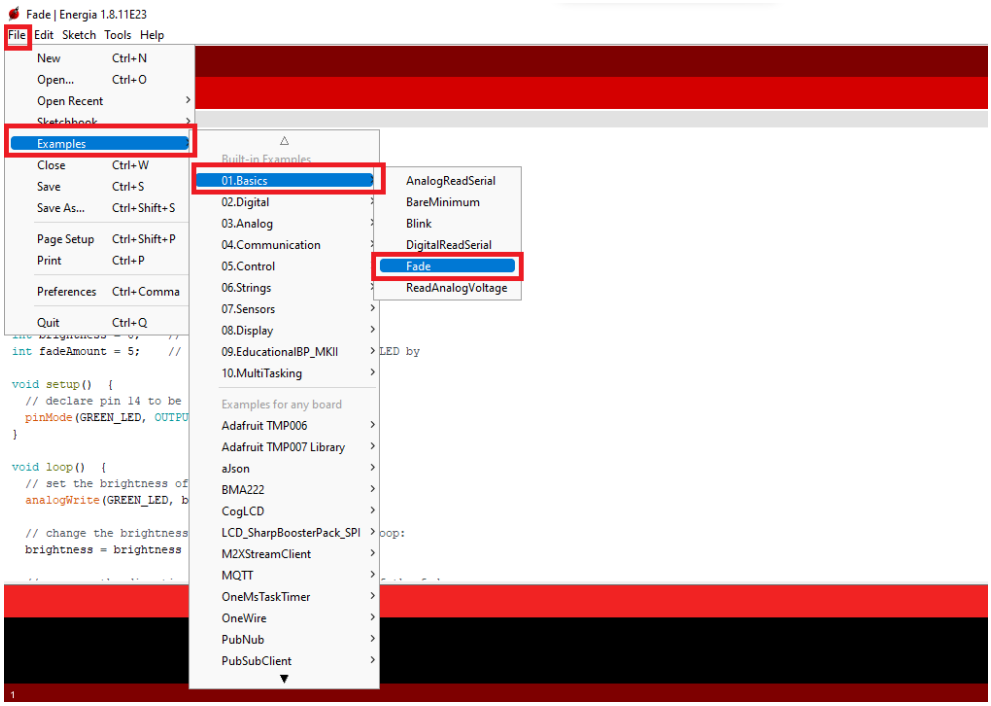


Figure 32

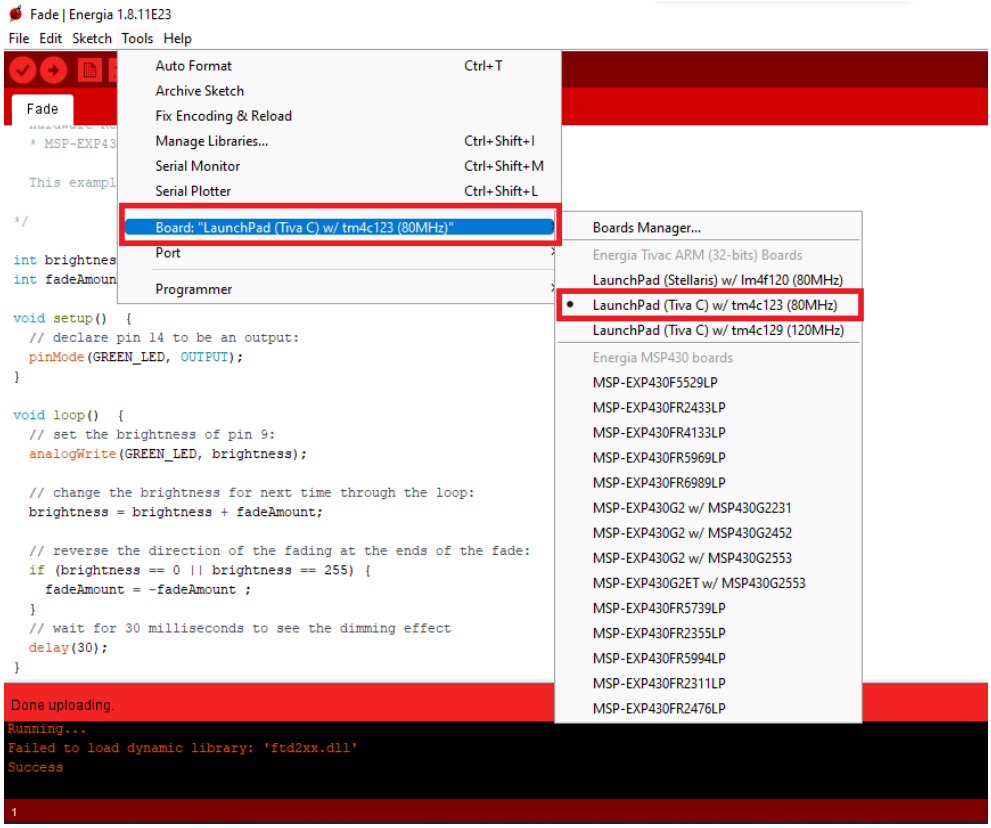
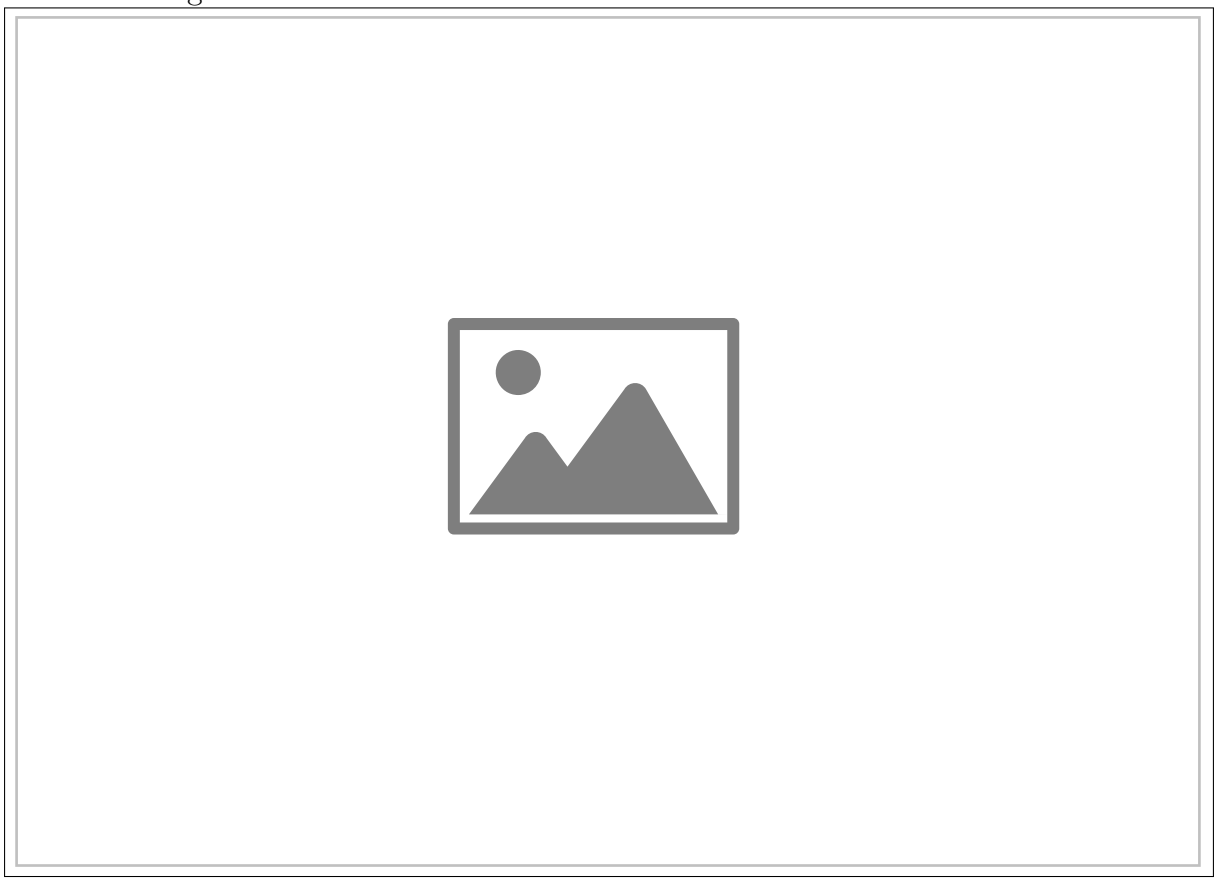


Figure 33

Figure 34

Figure 35

Now repeat the above task for "BLUE_ LED". Attach the screenshot of changes in code and image of LED on the board below:



Explain each variable, function and logic used in the code from your previous knowledge of C-programming to analyze how we are achieving the desired functionality of fading LED light:

Do enable the fade function we first need to get the LED switched on. And for that we write the function of type void named setup. This should declare the any hardware input or output that there should be, in our case we give inputs to the function 'pinmode' parameters like BLUE_LED, and set it as the OUTPUT.

Now to give switch our BLUE_LED on and giving it a fading effect. We make a continous loop for our LED to constantly lit up. Function analogueWrite is used to control the brightness of the led, and so the inputs are BLUE_LED, and it's brigthness value. We declare our variables brightness and fade_amount, and associate suitable values to them as 0, and 5 respectively.

Now we have written a statement that just increases the brightness all the time. And to counter the maximum value, everytime we increment the brightness we check if the brightness has reached the maximum allowable brightness (which is in this case is 255). If that happens reset the value of brigthness to zero. Considering that the system has a fast clock speed a suitable delay is added for the human eye to observe the fade effect.

Assessment Rubrics

Marks Distribution

		LR4	LR5	LR7	LR9
Pre-lab	Task 1	-	-	-	-
In-lab	Task 2	-	-	-	-
	Task 3	10 points	10 points	-	10 points
	Task 4	-	10 points	10 points	10 points
	Task 5	-	10 points	10 points	20 points
Total Marks 100					

Marks Obtained

		LR4	LR5	LR7	LR9
Pre-lab	Task 1	-	-	-	-
In-lab	Task 2	-	-	-	-
	Task 3			-	
	Task 4	-			
	Task 5	-			
Total Marks 100					