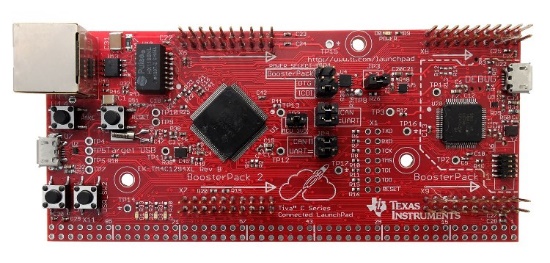
**TM4C1249 BOARD PROGRAMMING ON CCS(CODE COMPOSER STUDIO)**

**1.Tiva™ TM4C123G Series Overview**



**Specification:**

1. High Performance TM4C1294NCPDT MCU:
2. 120MHz 32-bit ARM Cortex-M4 CPU
3. 1MB Flash, 256KB SRAM, 6KB EEPROM
4. Integrated 10/100 Ethernet MAC+PHY, data protection hardware, 8x 32-bit timers
5. Dual 12-bit 2MSPS ADCs, motion control PWMs
6. USB H/D/O, and many additional serial communication interfaces
7. Dual, stackable [Booster Pack](https://www.ti.com/ww/en/launchpad/boosterpacks.html) XL connection sites
8. Cloud-based, Exosite QuickStart Application
9. TivaWare 2.1
10. On-board, in-circuit debug interface (ICDI)
11. Multiple development tool chain support: [CCS](https://www.ti.com/tool/ccstudio), Keil, IAR, Mentor & GCC

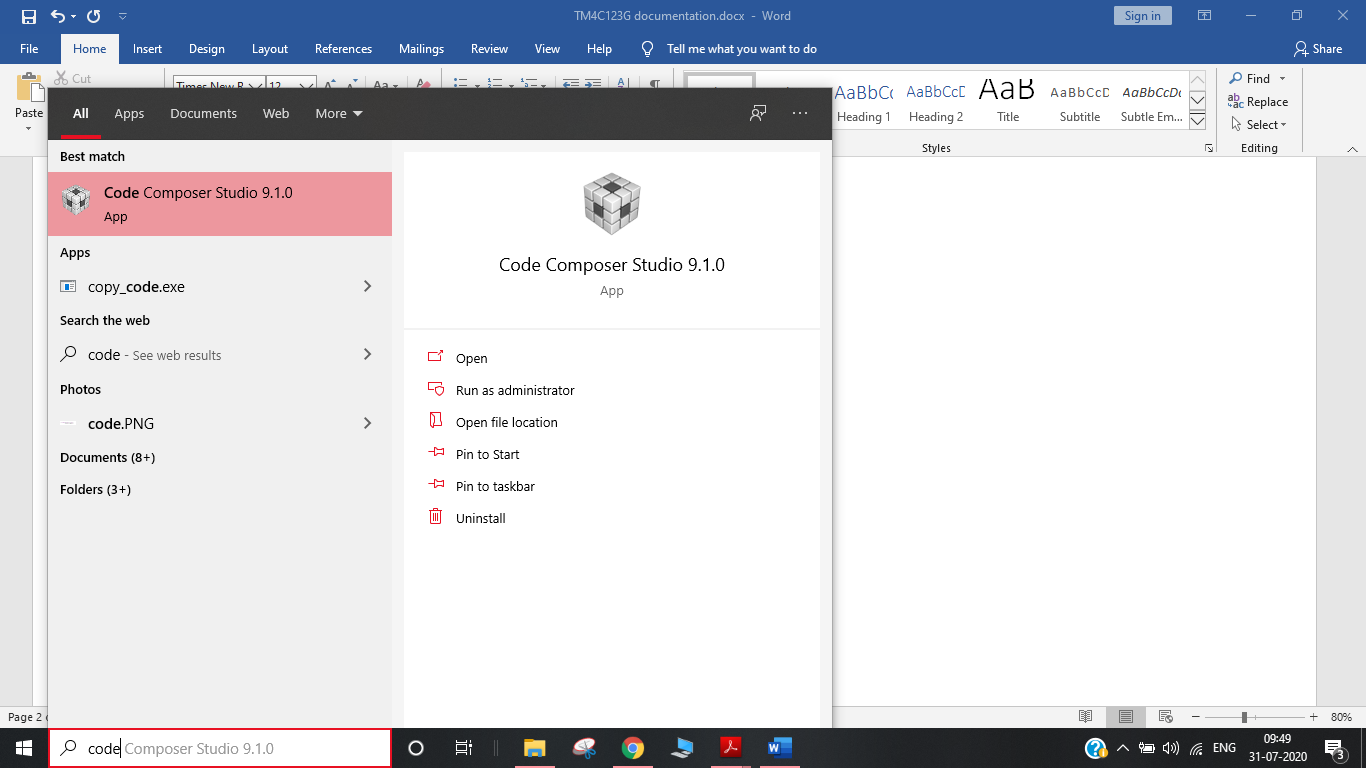
**Code Composer Studio with TivaWare!**

**You will need several things**. (Important to get it first)

1. Tiva Microcontroller (Such as a TM4C1294XL)
2. Code Composer Studio
3. Texas Instruments TivaWare Library (<http://www.ti.com/tool/sw-tm4c>(You'll want the complete library)

Get this and go to following steps:

**Step 1**: **Open CCS from computer**

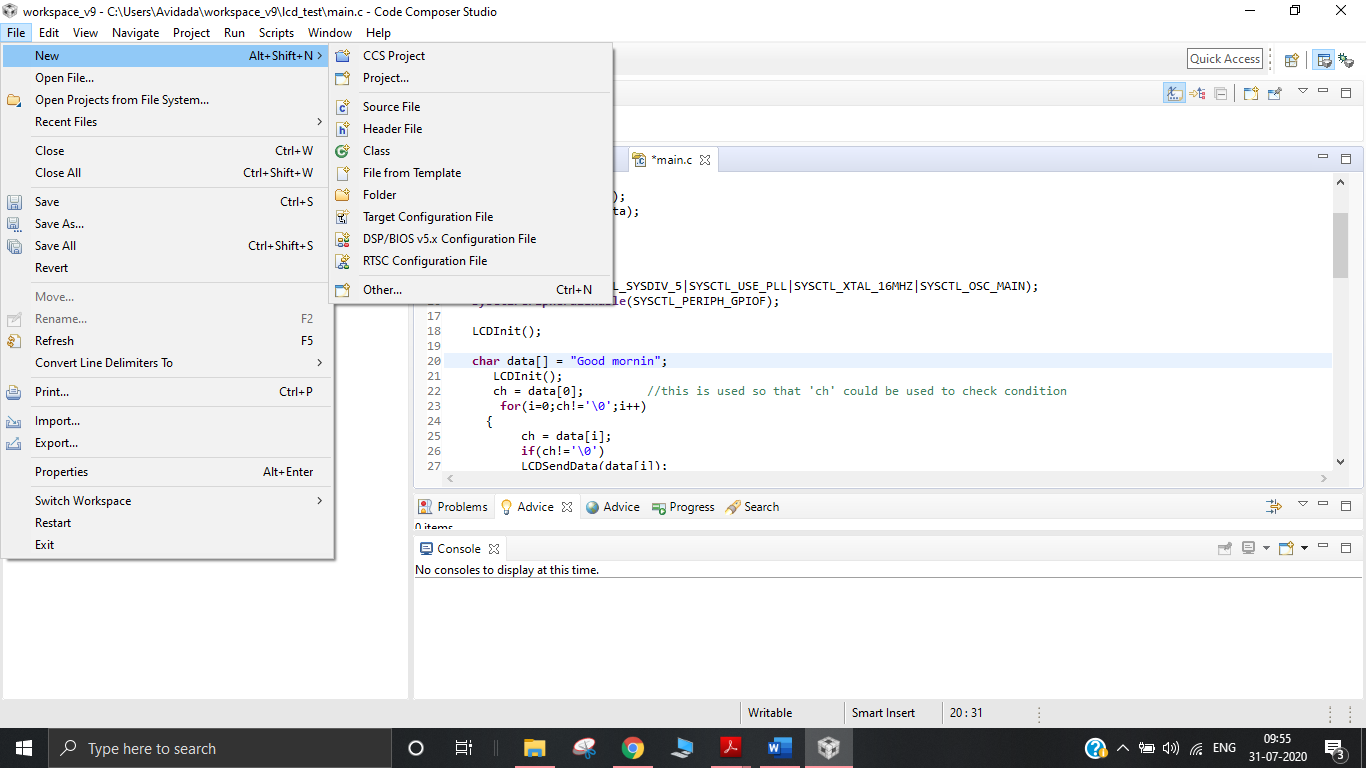


**Step2:**

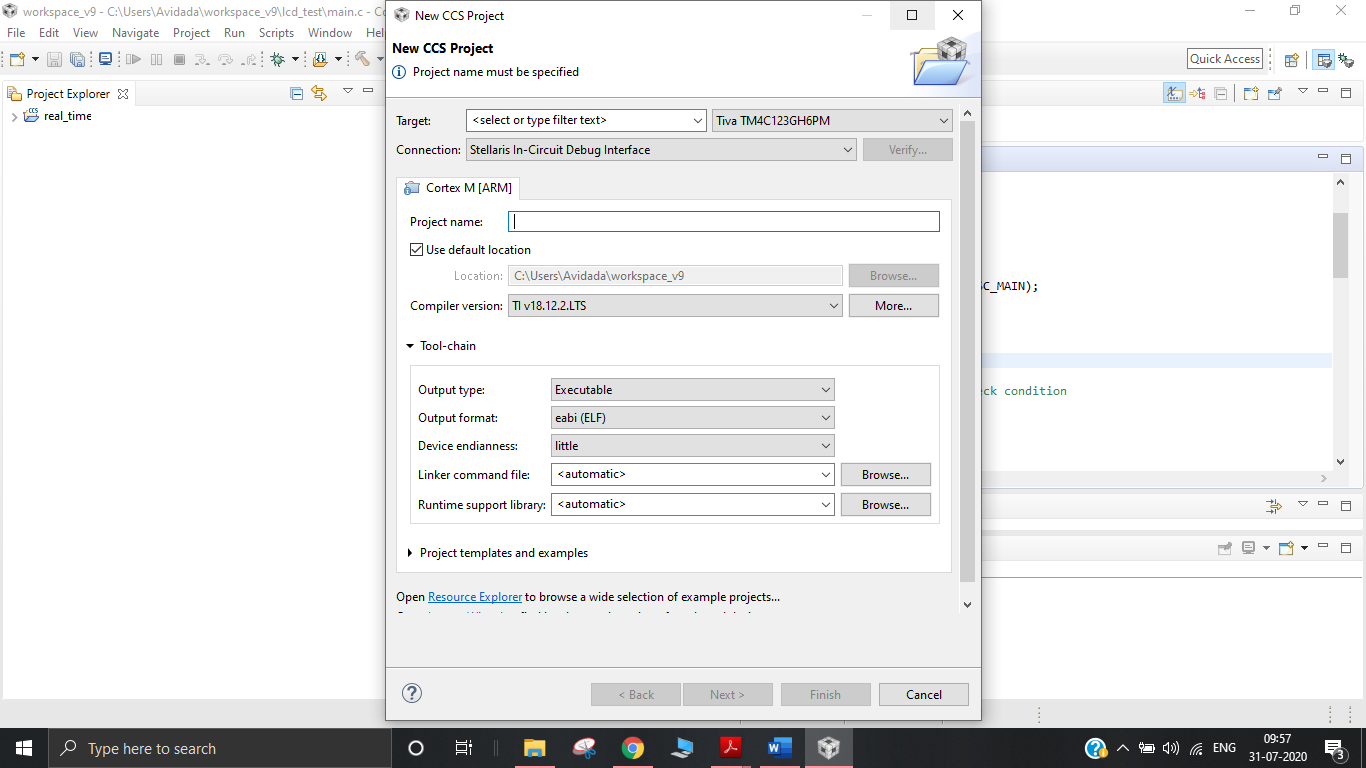
After opening go **to the file**

Then select **new**

And select **CCS project** as shown

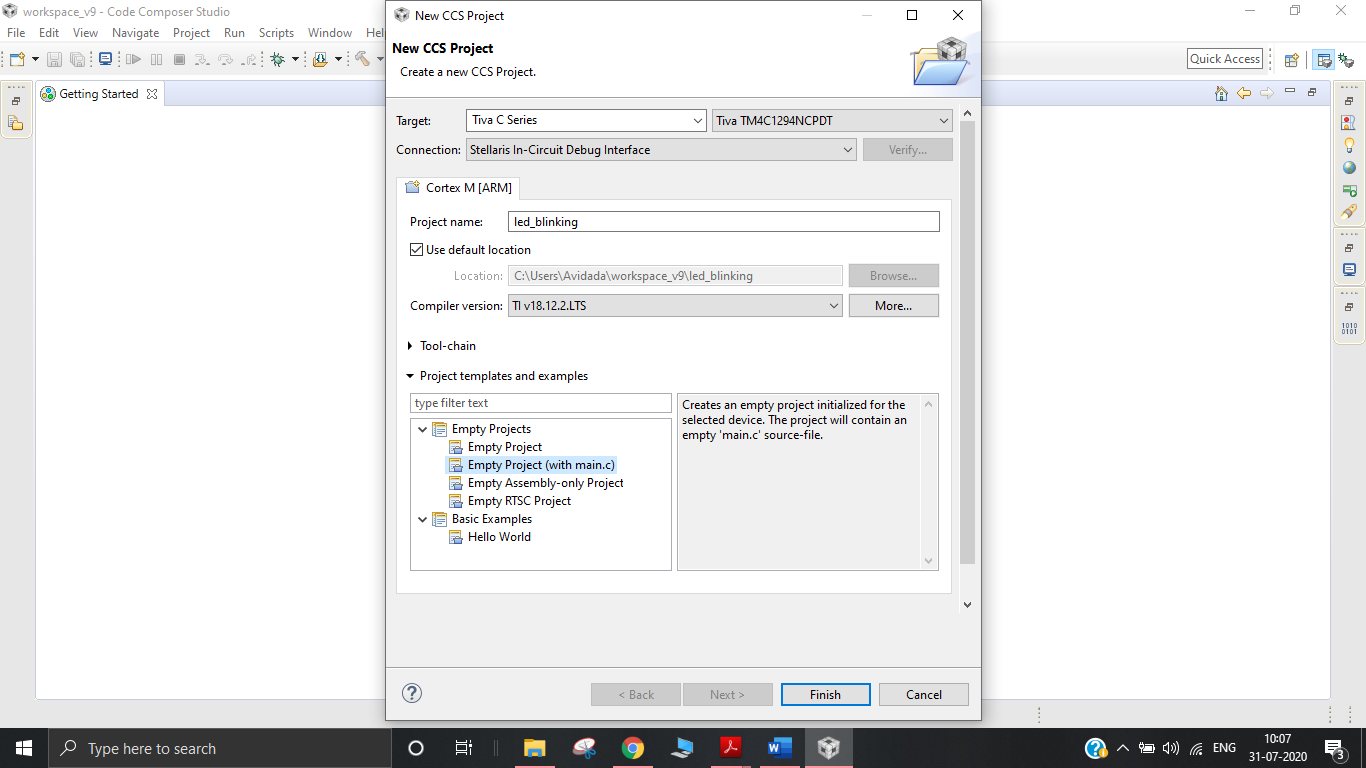


**Step3:**

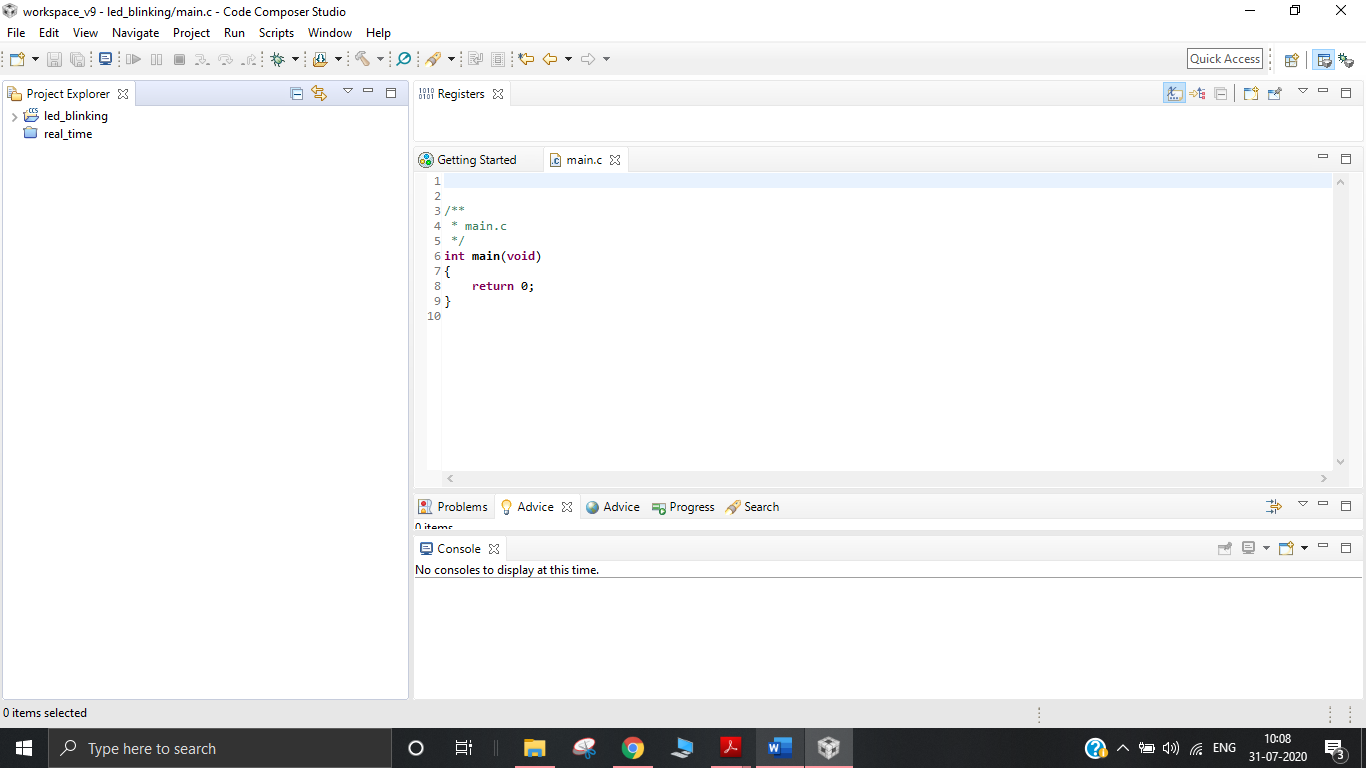


After this this window will be opened.

1. Select target as **Tiva C series** and model **TM4C1294NCPDT**
2. Connection: **Stellaris in-Circuit debug interface.**
3. Give project name anything you want.
4. In project template and example click **on Empty project (in main.c)**
5. Click on **finish**

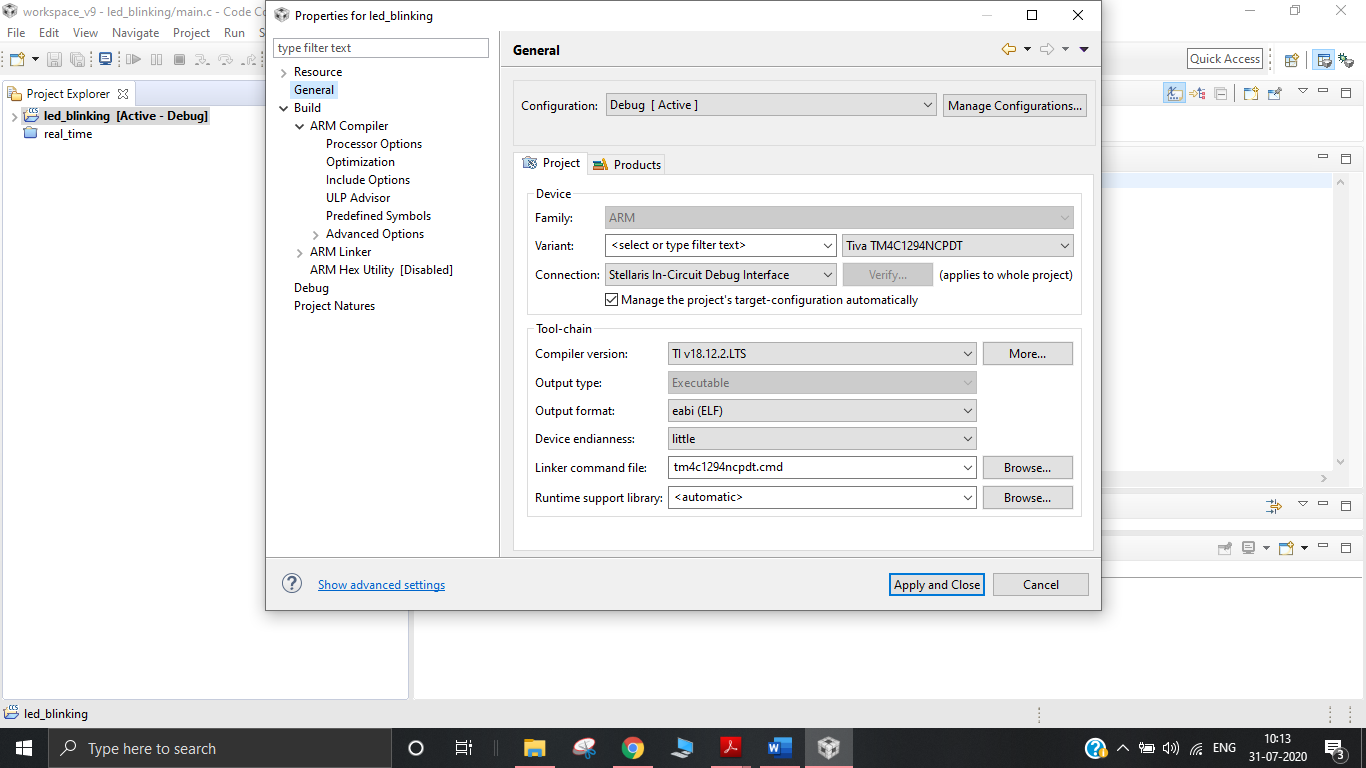


Step 4: Linking with TIVA libraries

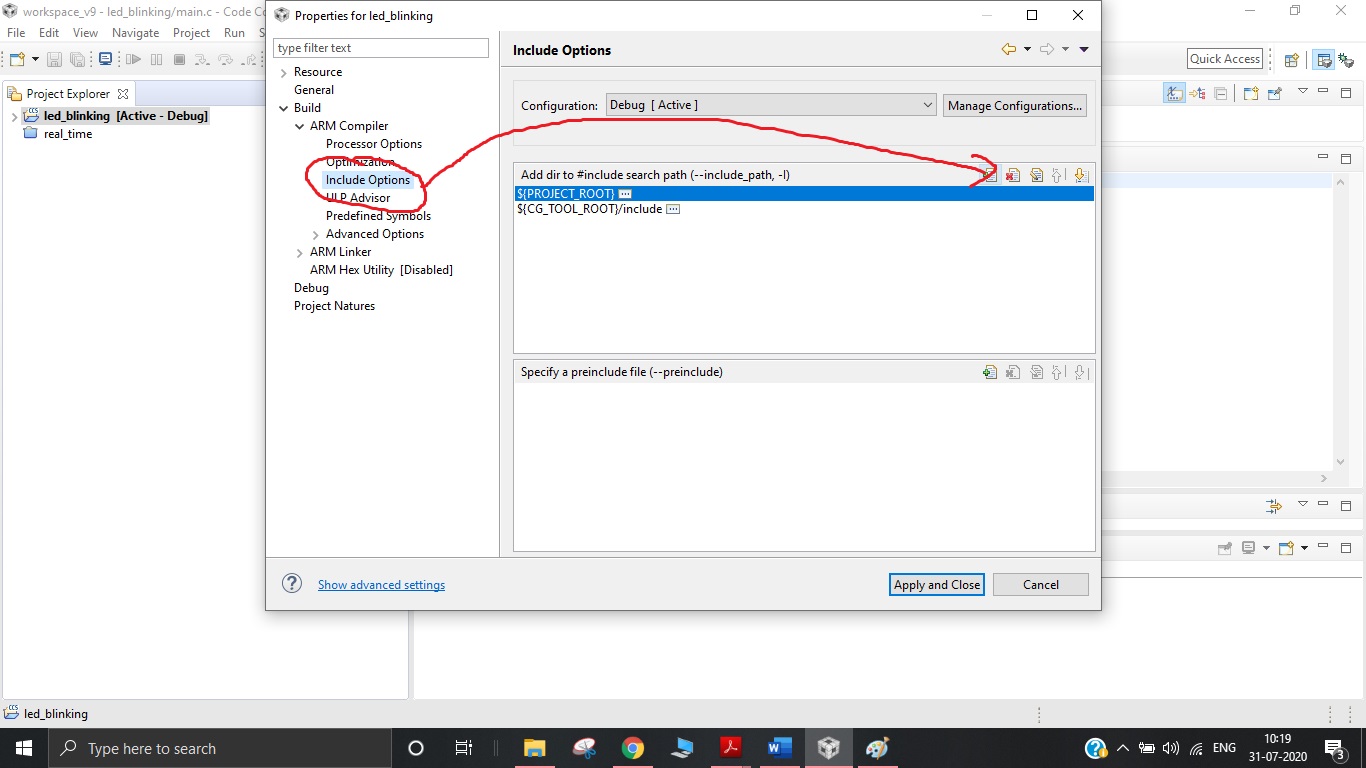


Now that you've created the project you should see it appear in the left-hand column named **led\_blinking**. We are going to start linking in everything needed for the TivaWare libraries.

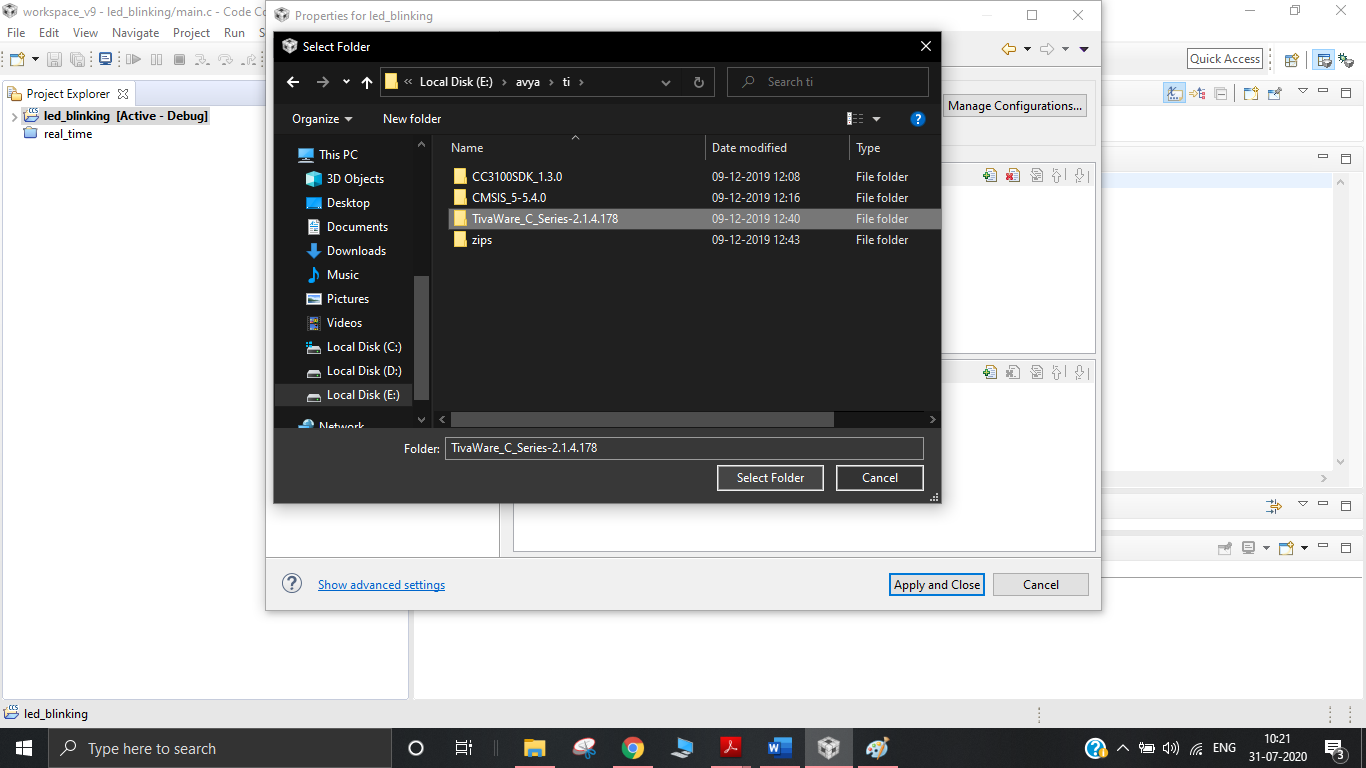
* Right click on **project name** and **select properties**.

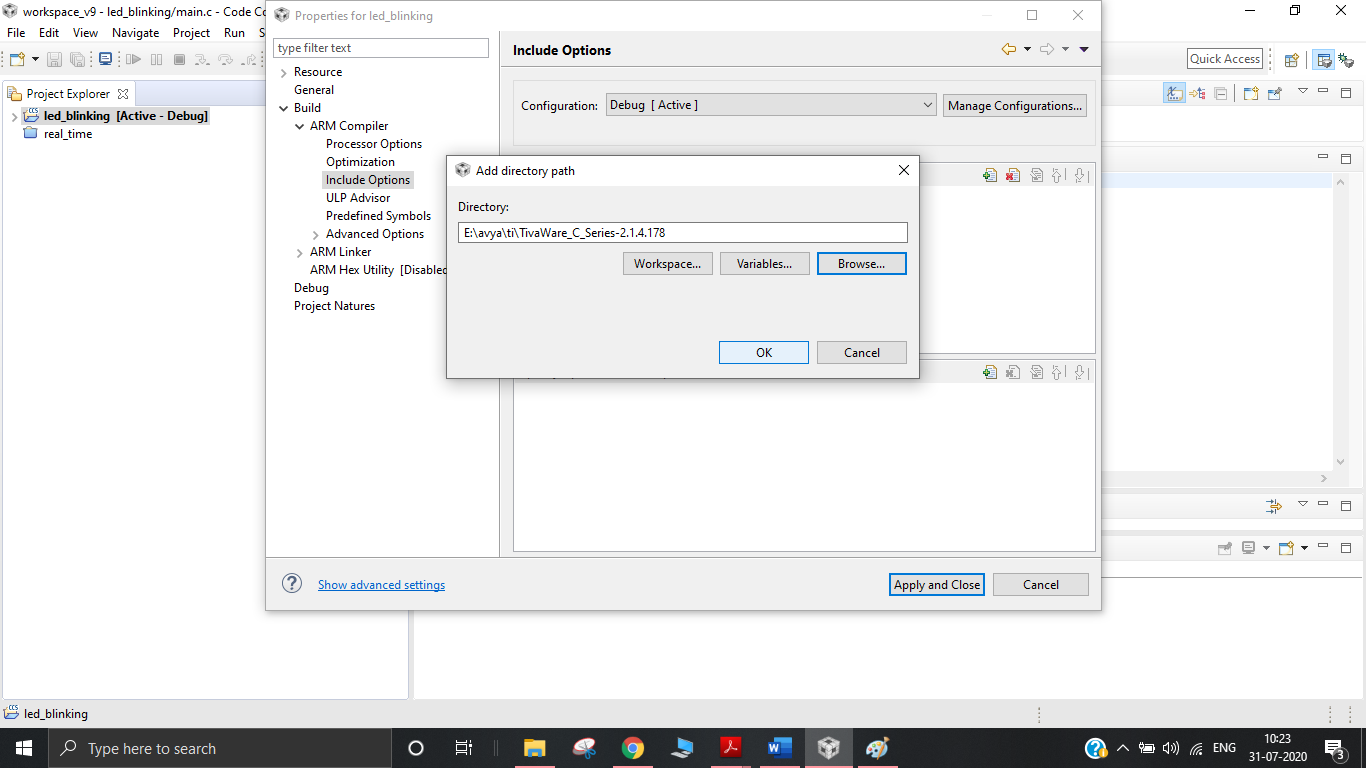


* From here go to the **ARM Compiler** section and click on the **Include Options.**



In here we are going to click the paper with a **plus sign icon** to **add a new directory** to our project. In this case we want the TivaWare Library which you have downloaded on your computer.

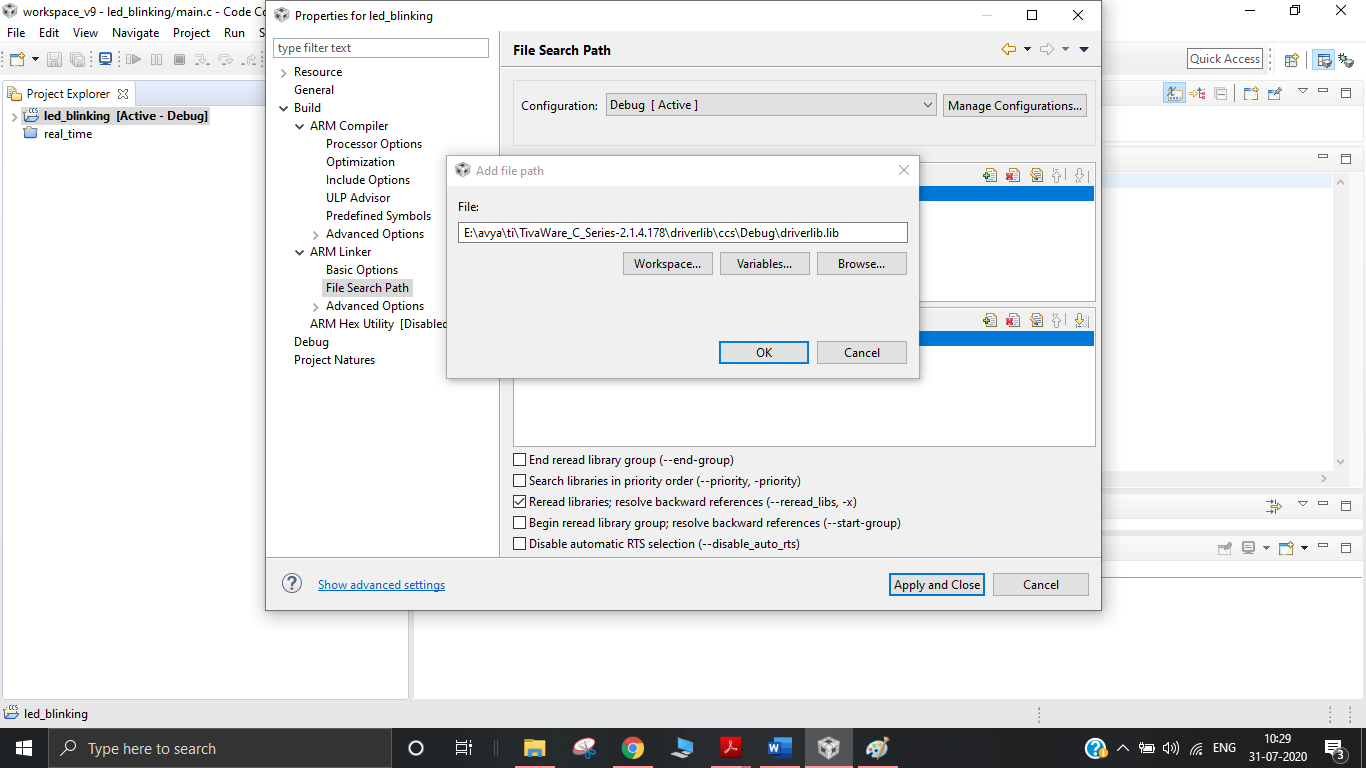
Choose **whole folder** and then click ok as follow:



* Now move **to ARM linker section** on same page and go to **file search path.**

Similar to last step we will hit the paper with **a plus sign icon** to add a new **library file**

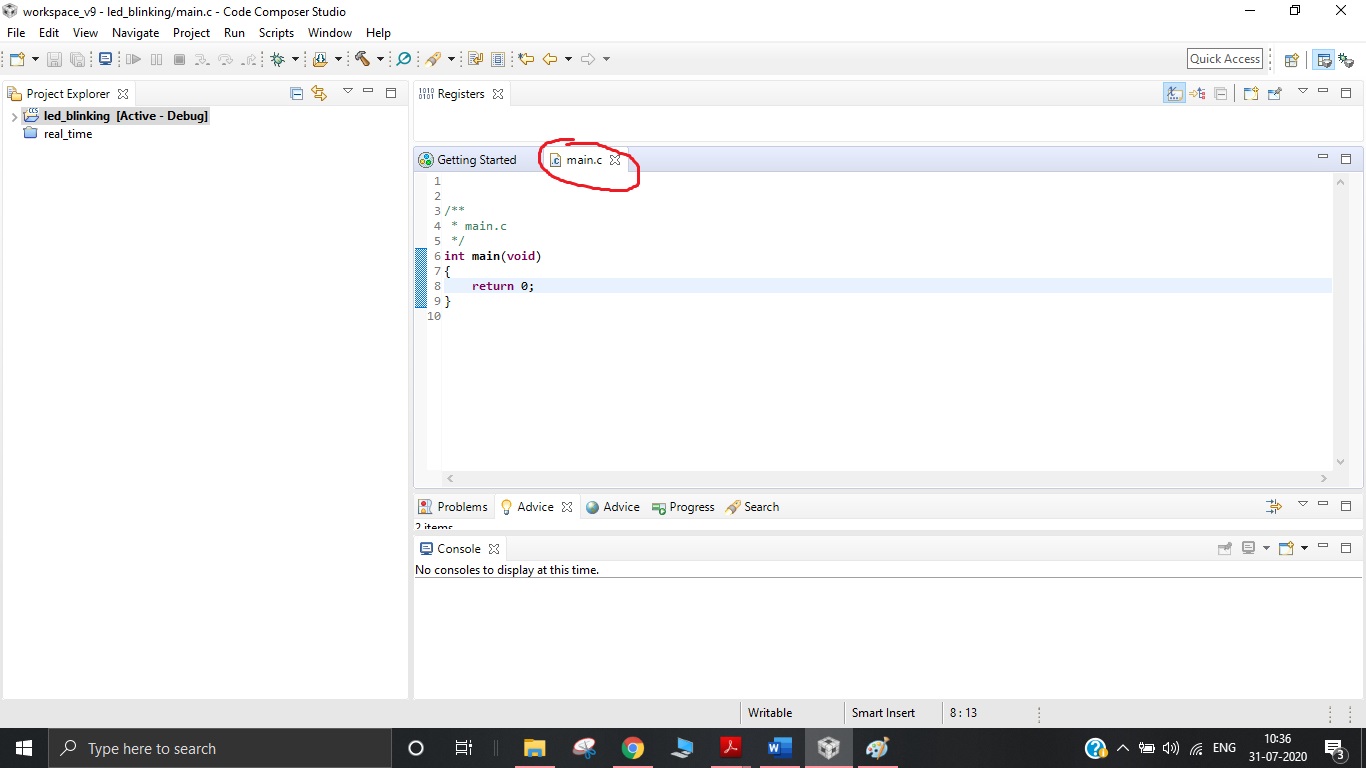
Remember: You have to add library file so **file search path.**add it on library file section It is in same tivaware folder which you add earlier path is : TivaWare\_C\_Series-2.1.4.178\driverlib\ccs\Debug\driverlib.lib



* Click on Apply and close

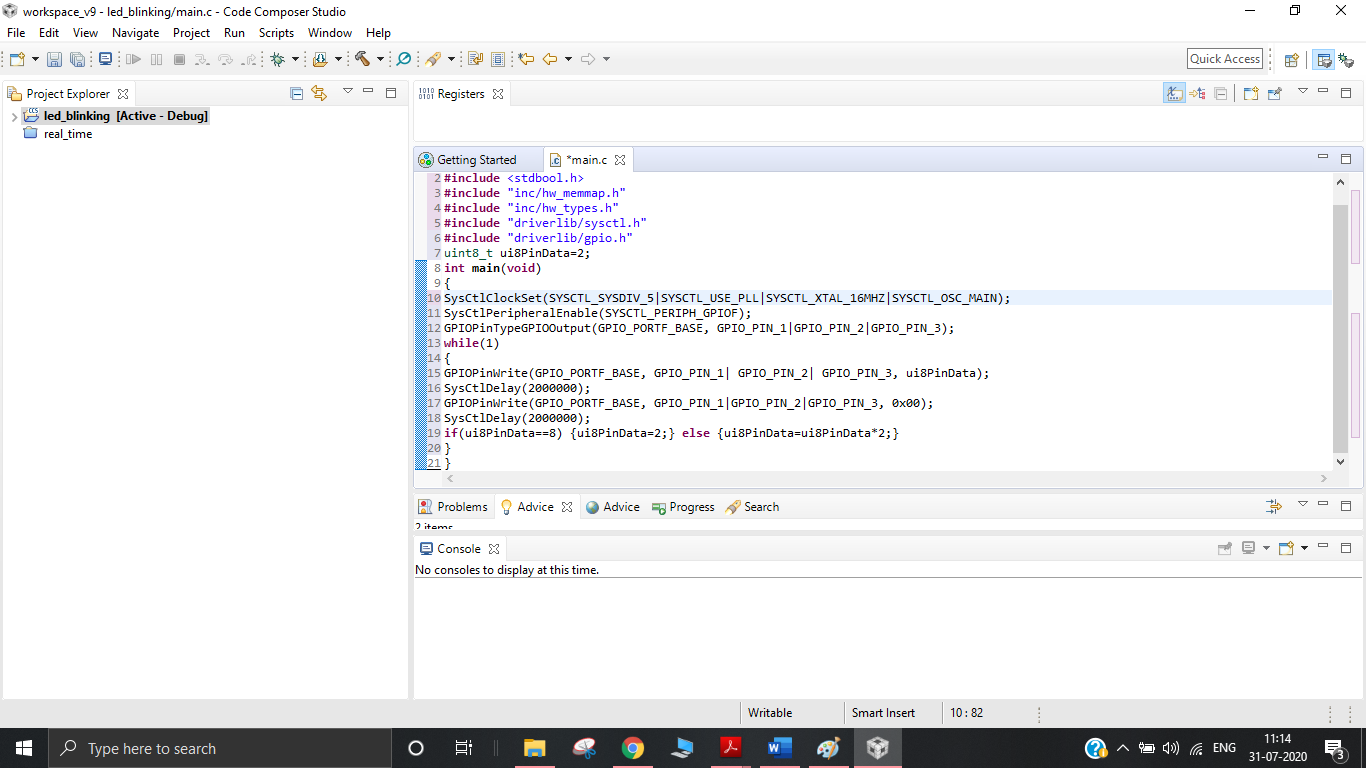
**Step 5**: **Writing and Running project**

Now we are almost done and we can write programme in main.c.

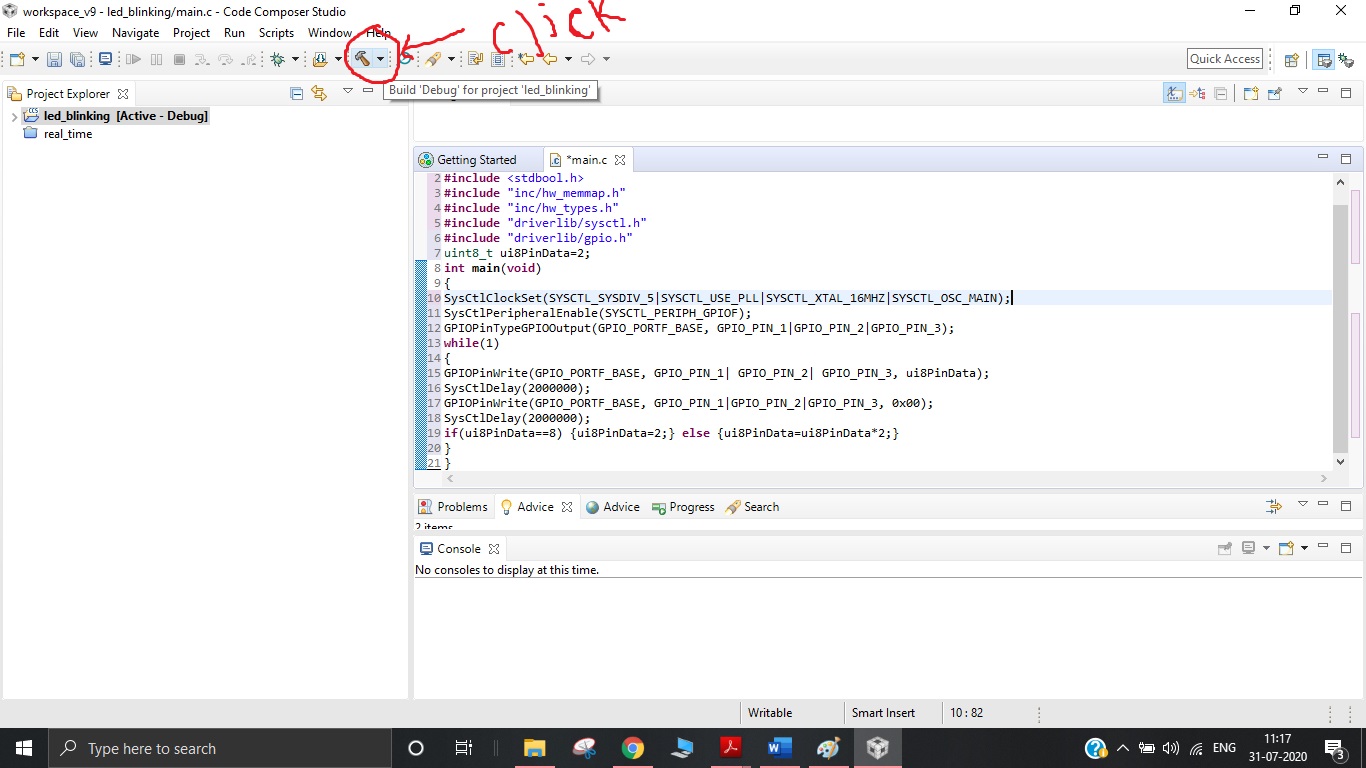


**Example of blinking LED:**

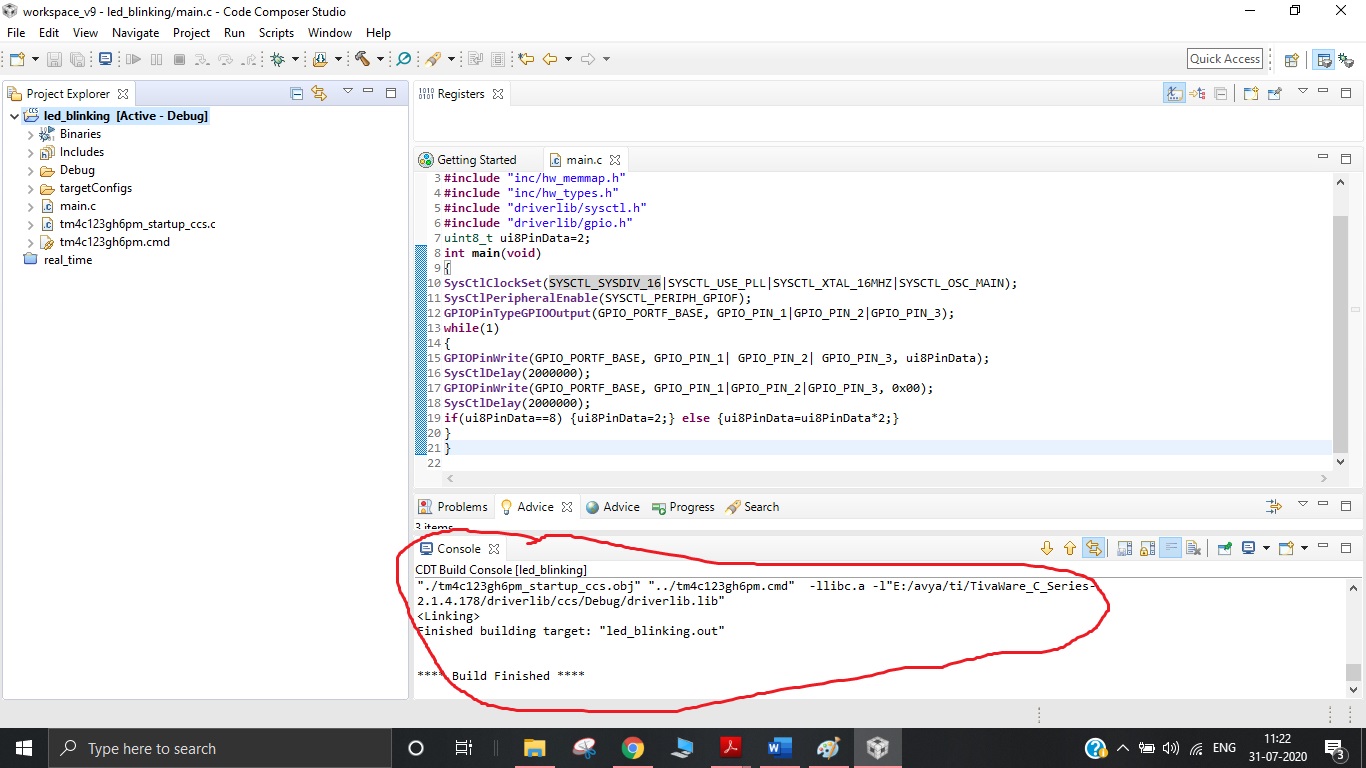
* I write Led\_blinking Code in main.c



* After writing whole code we need to compile it **click on build** button as shown and code will start to compile.

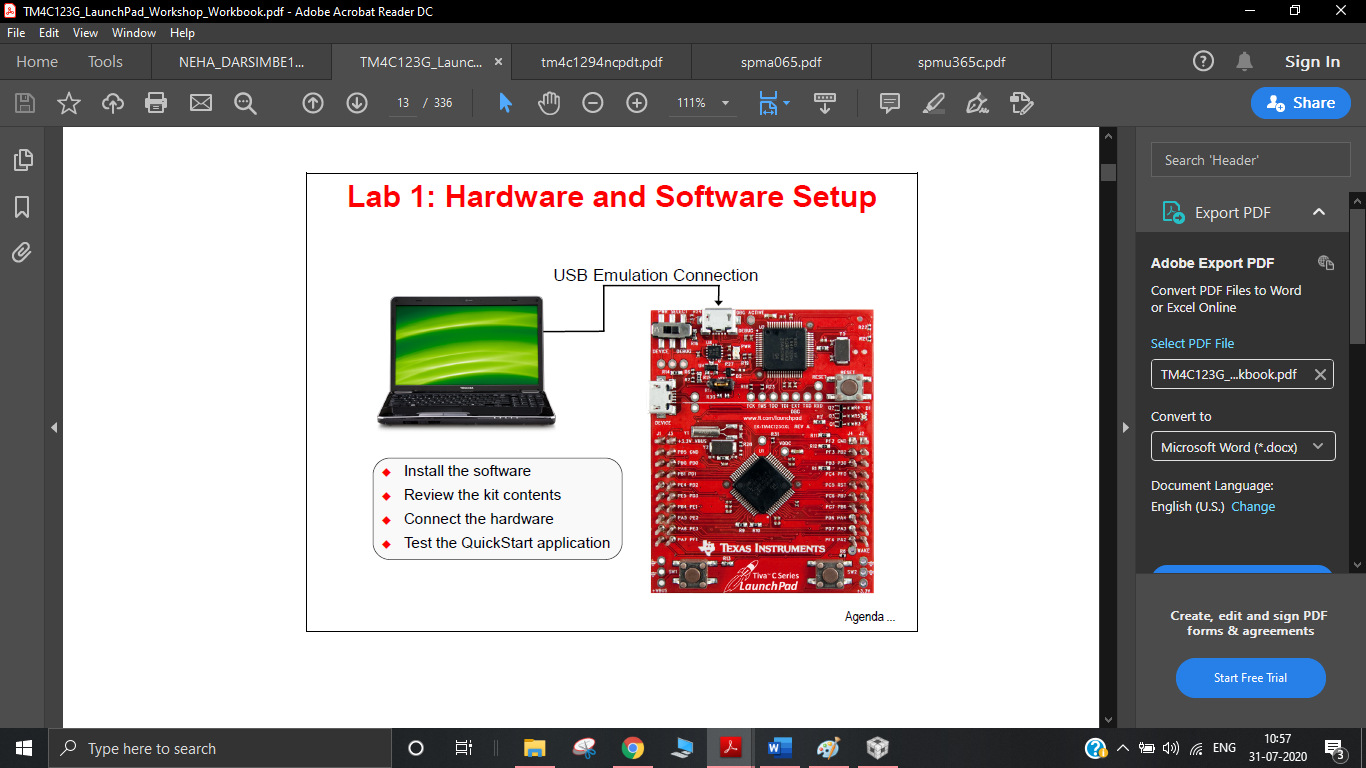


* If the Error exit it will give error and build will not get finished .You can check the errors if the errors are not in the code check for step4 again if files are there or not and the compile .An error free code will get build and you can see it as below

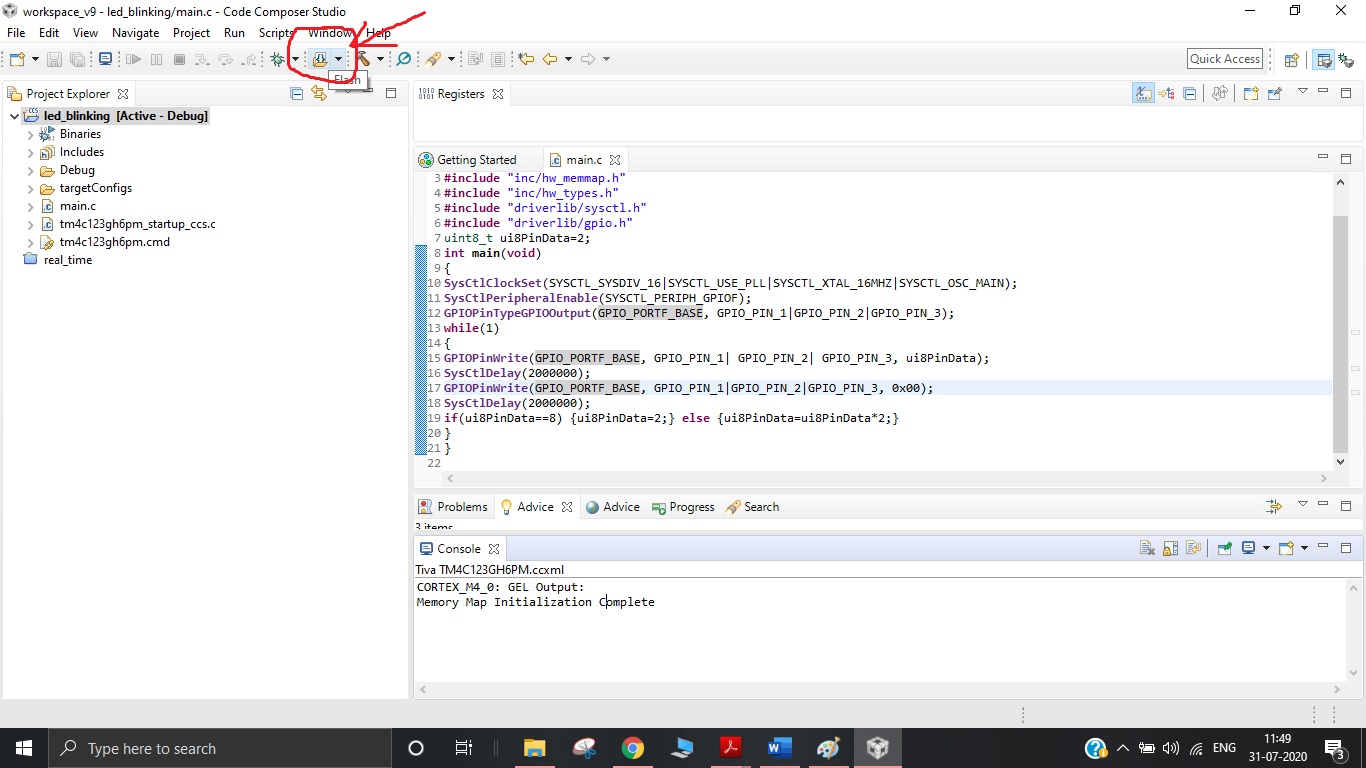


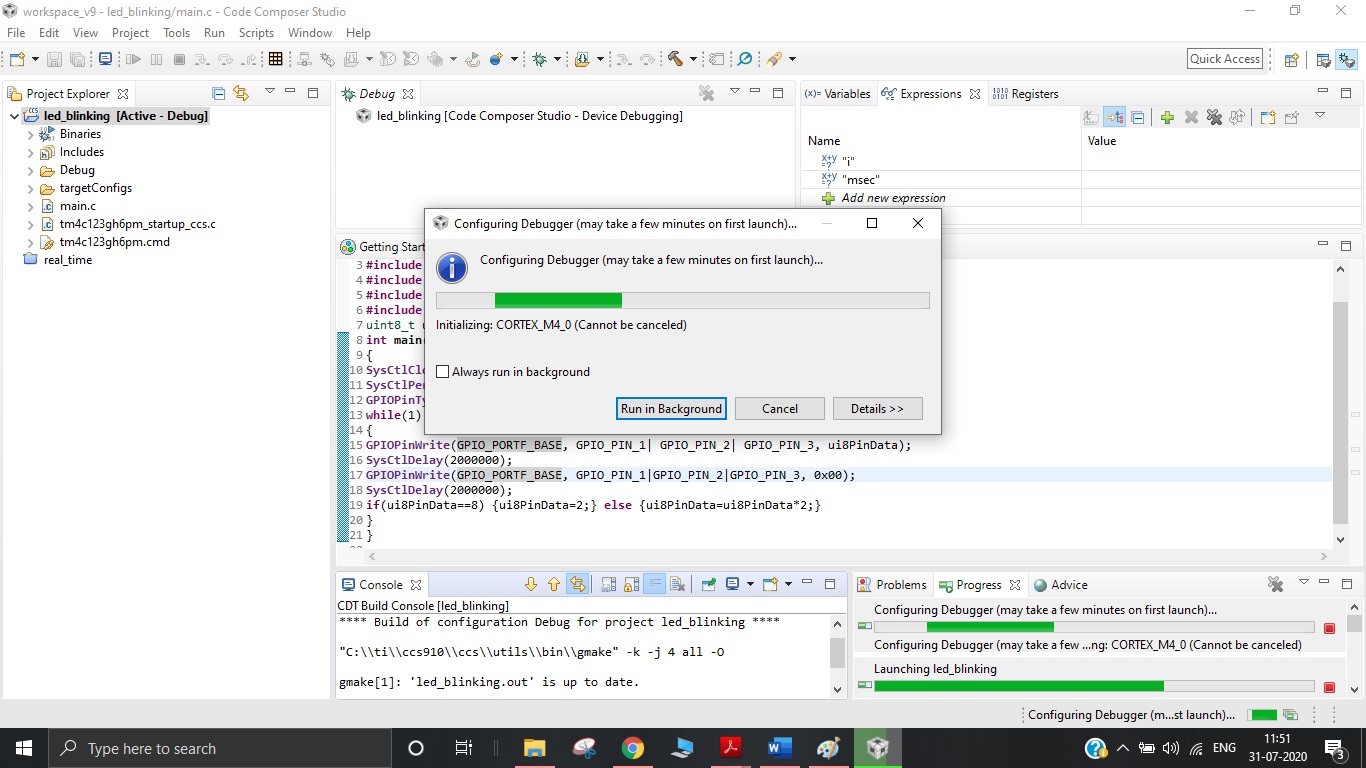
**Step 6:** Debug and Run program on hardware

* Now Connect your Board to computer
* I have written led\_blinking code which will blink leds present on board.So no extra hardware or module connection needed



* After Connecting the board to PC click on the **flash icon.**





After launching you will see on console that

CORTEX\_M4\_0: GEL Output:

Memory Map Initialization Complete

And you will go into debugger section where you can debug your code.

**Step 7**: **Done ..!!** the code will get dump into hardware and will start to work:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Sample Code of LED\_BLINKING for testing on TM4C1294XL board.**

//Copy paste in main.c file

#include <stdint.h>

#include <stdbool.h>

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/sysctl.h"

#include "driverlib/gpio.h"

uint8\_t ui8PinData=1;

int main(void)

{

SysCtlClockFreqSet((SYSCTL\_XTAL\_25MHZ | SYSCTL\_OSC\_MAIN | SYSCTL\_USE\_PLL |

SYSCTL\_CFG\_VCO\_480), 120000000);

SysCtlPeripheralEnable(SYSCTL\_PERIPH\_GPION);

GPIOPinTypeGPIOOutput(GPIO\_PORTN\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1);

GPIOPinWrite(GPIO\_PORTN\_BASE, GPIO\_PIN\_0|GPIO\_PIN\_1, 0x00);

while(1)

{

GPIOPinWrite(GPIO\_PORTN\_BASE, GPIO\_PIN\_0 | GPIO\_PIN\_1, ui8PinData);

SysCtlDelay(2000000);

if(ui8PinData==4) {ui8PinData=1;} else {ui8PinData=ui8PinData\*2;}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*