# LAB-01

# ESD-I

# Please visit lab website-

# 

# <https://labs.dese.iisc.ac.in/embeddedlab/how-to-do/>

# (or)

1. Download Code Compose Studio windows-[download](https://dr-download.ti.com/software-development/ide-configuration-compiler-or-debugger/MD-J1VdearkvK/12.2.0/CCS12.2.0.00009_win64.zip)

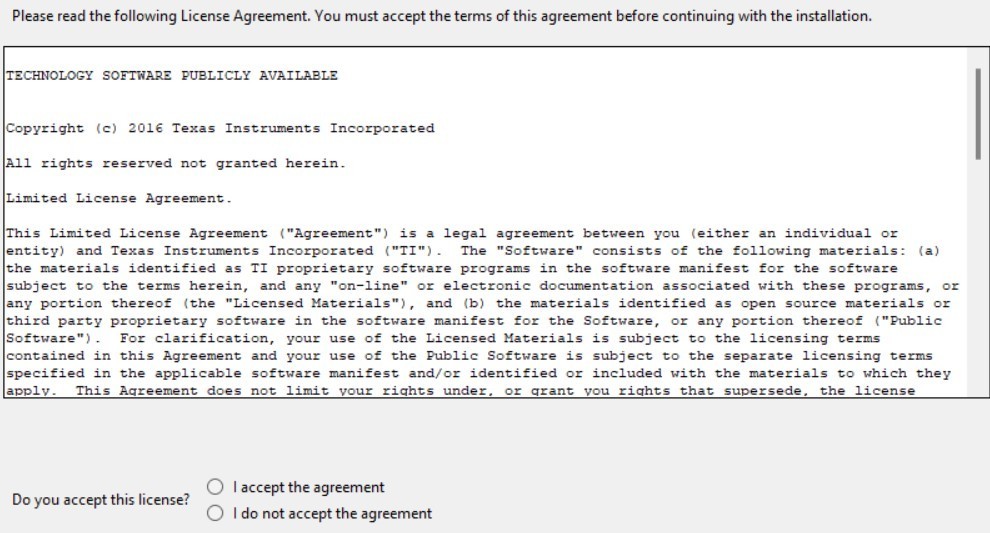
Linux- [download](https://dr-download.ti.com/software-development/ide-configuration-compiler-or-debugger/MD-J1VdearkvK/12.2.0/CCS12.2.0.00009_linux-x64.tar.gz)

update system and Install Dependent libraries.

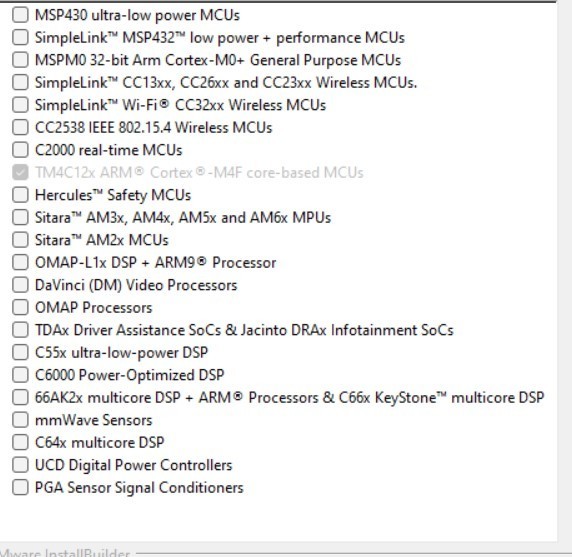
* + update system: sudo apt update
  + install dependent libraries:

sudo apt install libc6-i386 libusb-0.1-4 libgconf-2-4 libncurses5 libpython2.7 libtinfo5

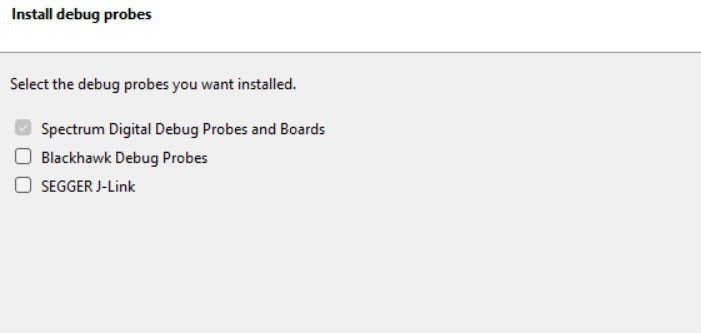
* + install CCS
  + if CCS was installed as user then run the driver install script
    - go to <CCS\_INSTALL\_DIR>/ccs/install\_scripts
    - sudo ./install\_drivers.sh
* Once download finished unzip the CCS\_Setup and Run.
* This will start CCS Install Wizard.
* Accept the software License Agreement and click Next.



* choose installation typeselect the components TM4C123X....



* select debug probes.. and finish installation.



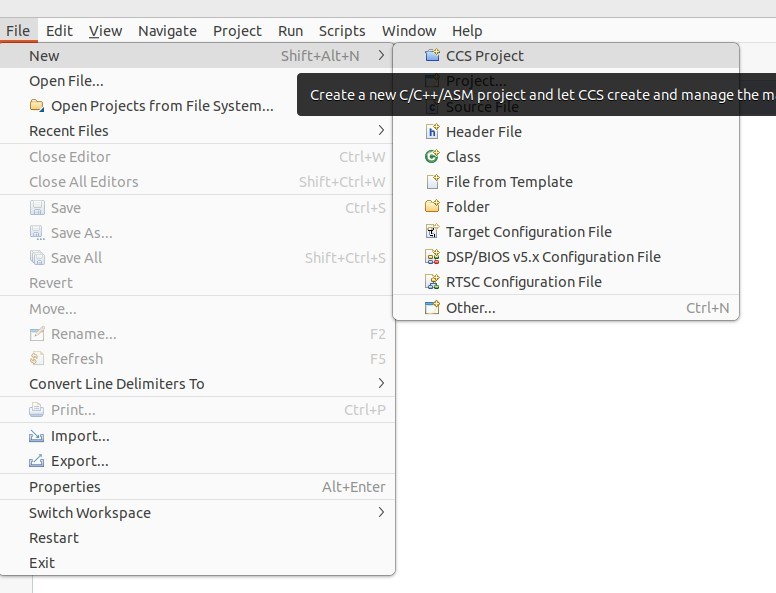
1. Download TIVAWARE C Series

(or)

[Download](https://indianinstituteofscience.sharepoint.com/:u:/s/E3-2572024/ETtPeumcobpJo5qbkQIWd3ABwvNpwhT38MTYEnIRob64fA?e=atZq5d)

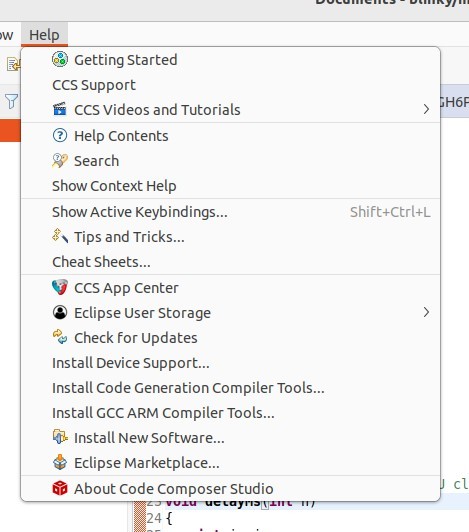
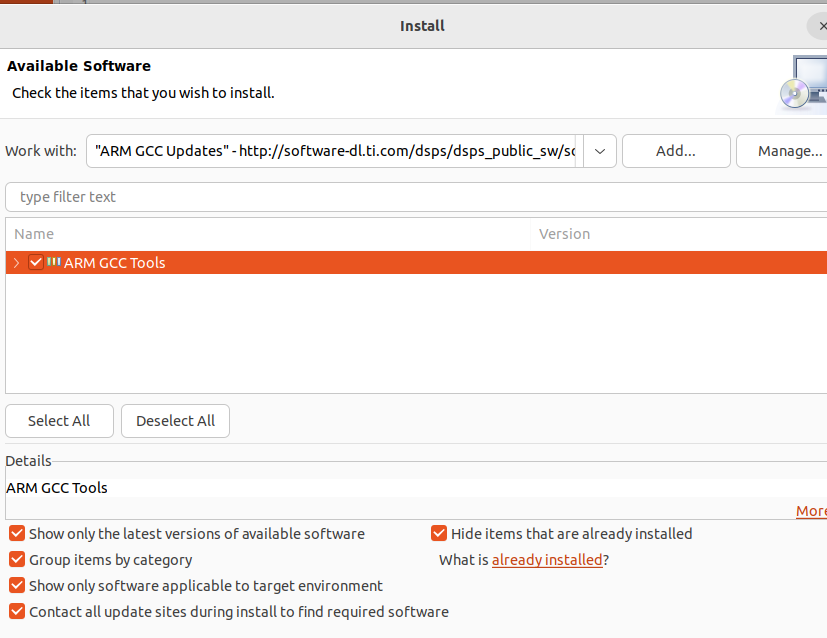
1. Create a New Project

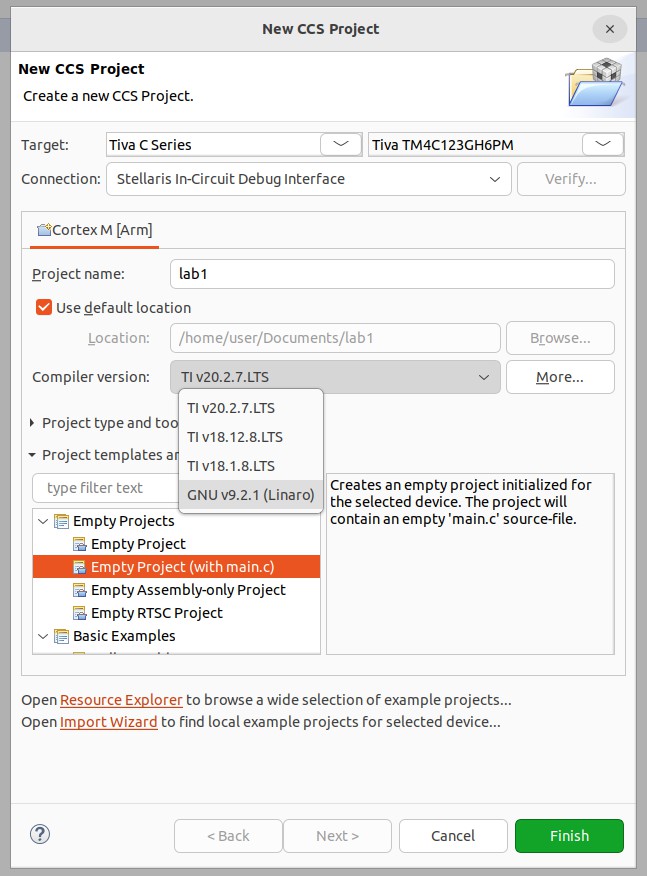
To Create new project Select -> Select New CCS project



->Select Compiler version, Target, connectioon as follows:

> in Help install GCC Arm Compiler tools.code composer studio includes a Linaro distribution of the ARM GNU compiler, so Programmers have the option of using GCC Compiler insted of the TI compiler to build their projects.



## Add path Build Variables:

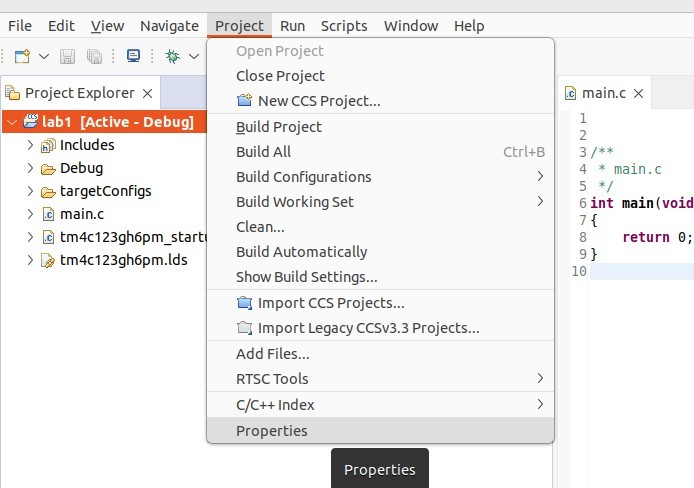
**Path variable** –

when you add (link) a file to your project, you can specify a “relative to” path. The default is PROJECT\_LOC which means that your linked resource (like a .lib file) will be linked relative to your project directory.

## Build variable –

used for items such as the search path for include files associated with a library – i.e. it is used when you build your project.

Variables can either have a project scope (that they only work for this project) or a WORKSPACE scope (that they work across all projects in the workspace).

In the next step, we need to add (link) a library file and then add a search path for include files. First, we’ll add these variables MANUALLY as PROJECT variables. Later, we will show you a quick and easy way to add these variables into your WORKSPACE so that any project in your workspace can use the variables.

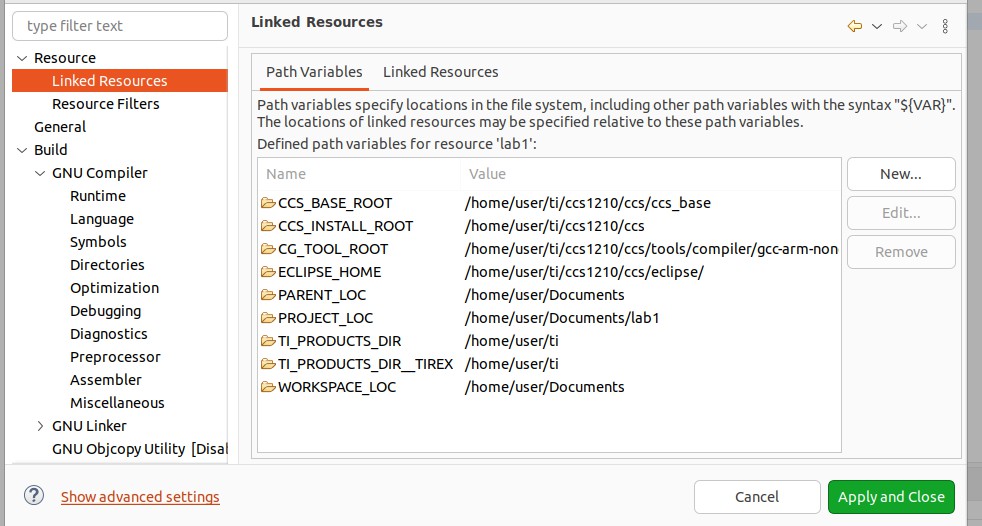
# adding path variable:

->click New

->Type Tivaware\_Install for the Name

->For the location ,Click the Folder and navigate to Tivaware Installation.

->click apply and close.



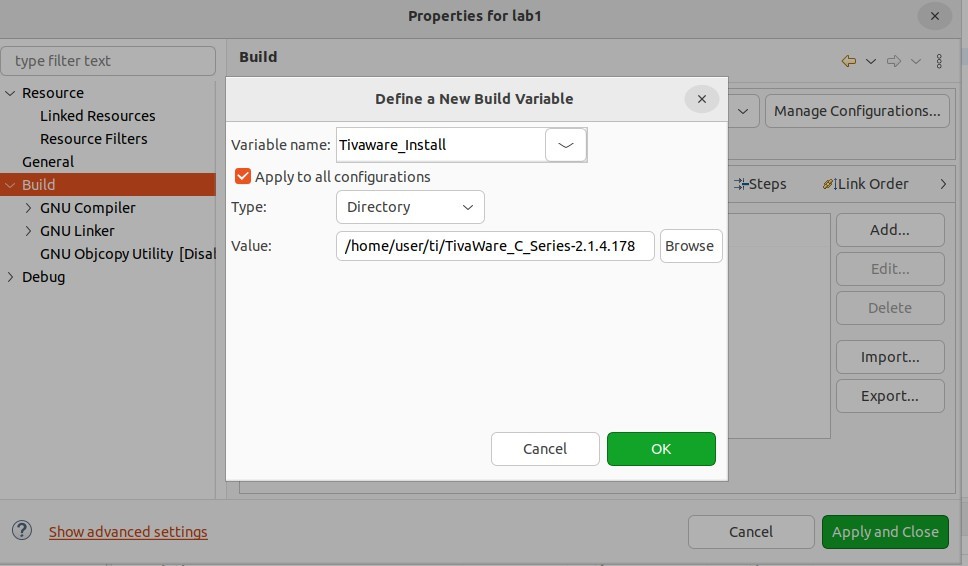
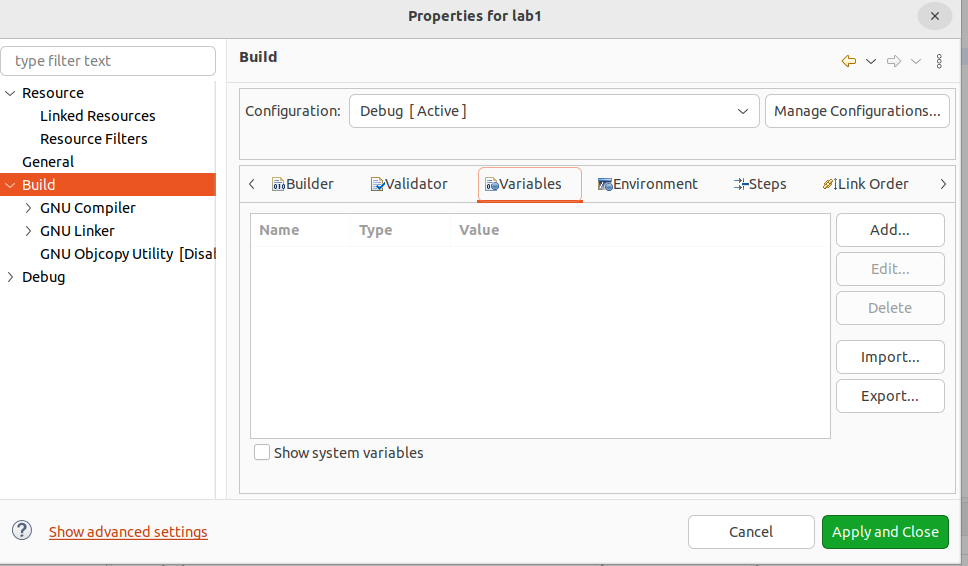
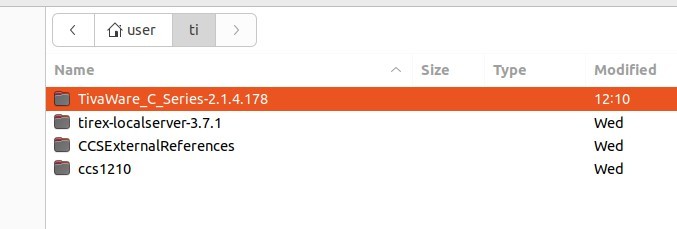
# Adding a Build Variable:

->click build Variable tab

-> select Tivaware\_install

-> apply to all configurations.

->change Type to “Directory” and click “Ok”

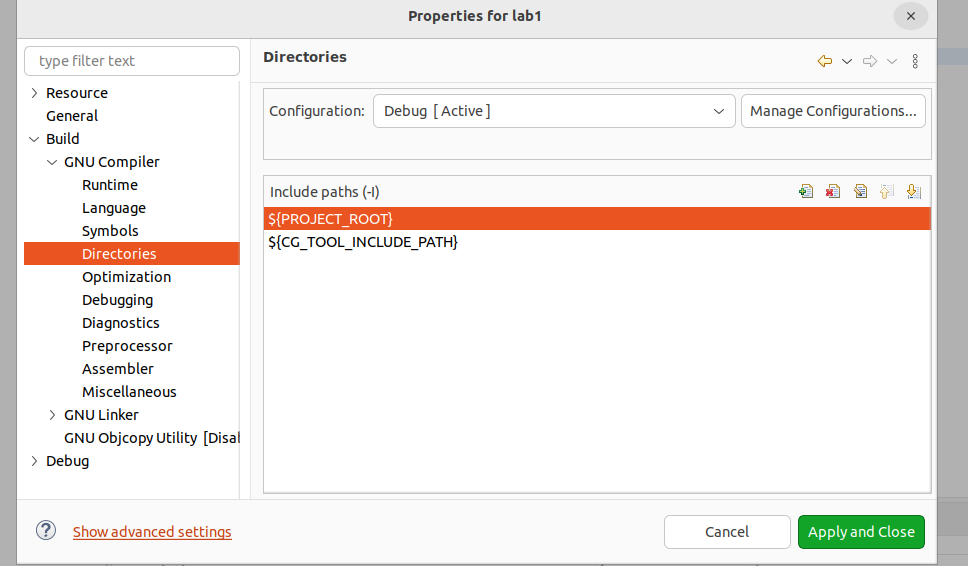


Add the Include Search Paths:

->In Properties click Build->GNU Compiler->Directories:

->click “+” sign add the variable(Tivaware\_Install).

->click ok.

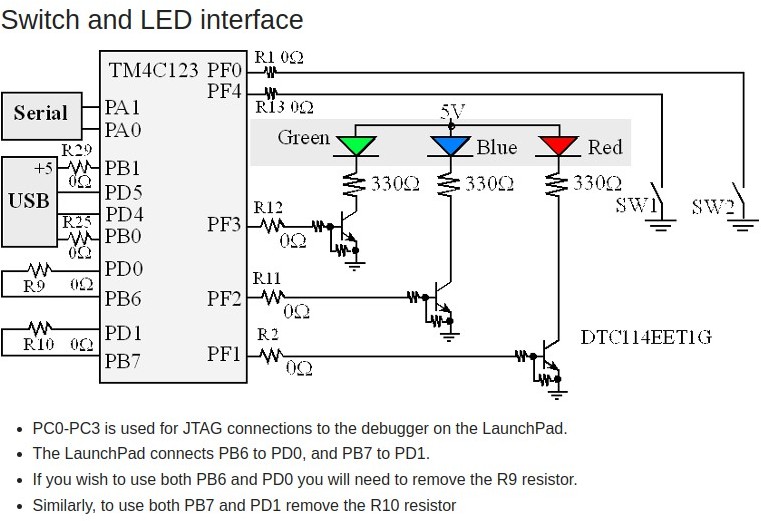


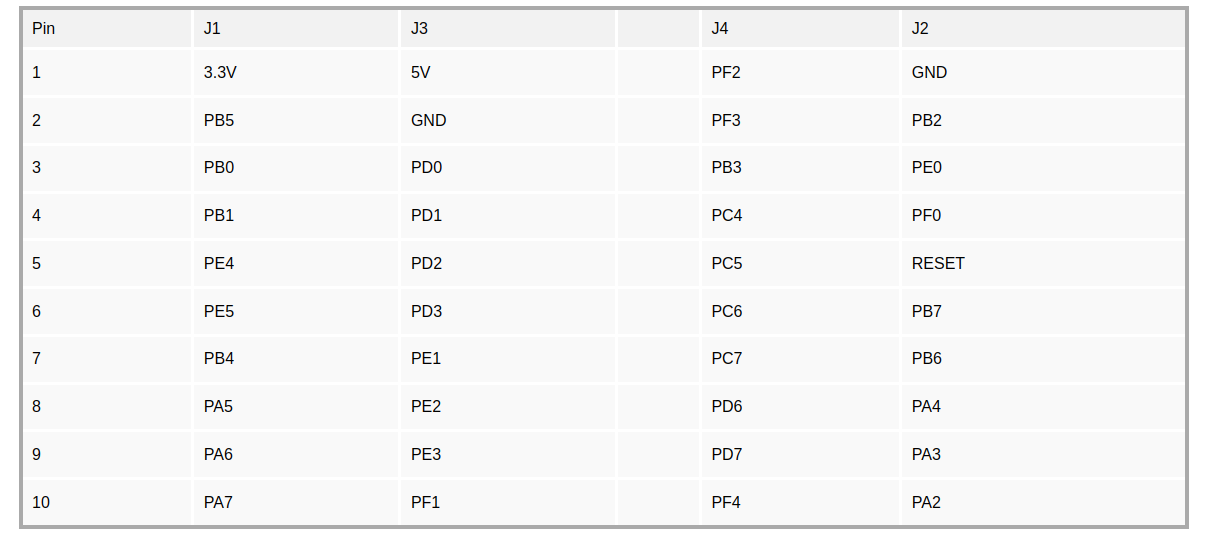
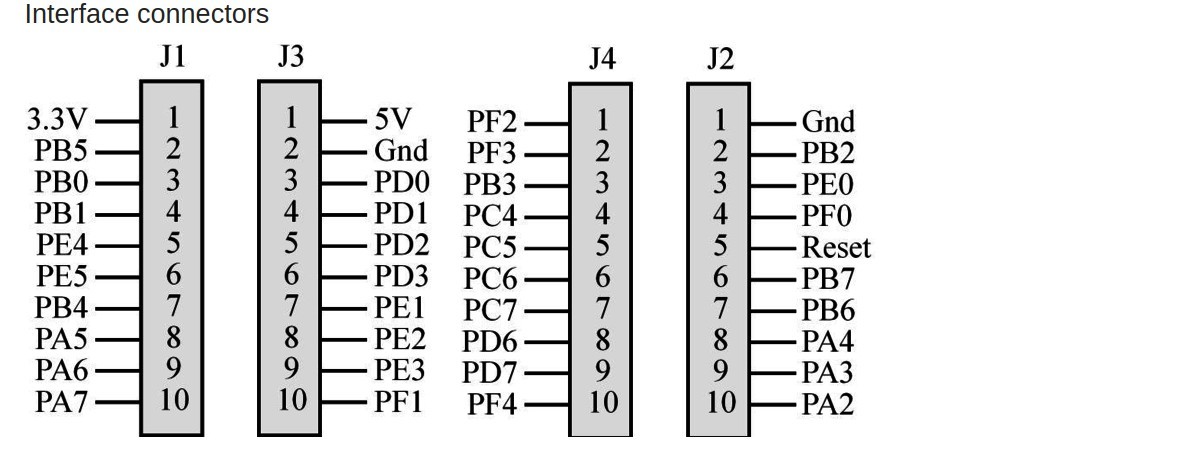
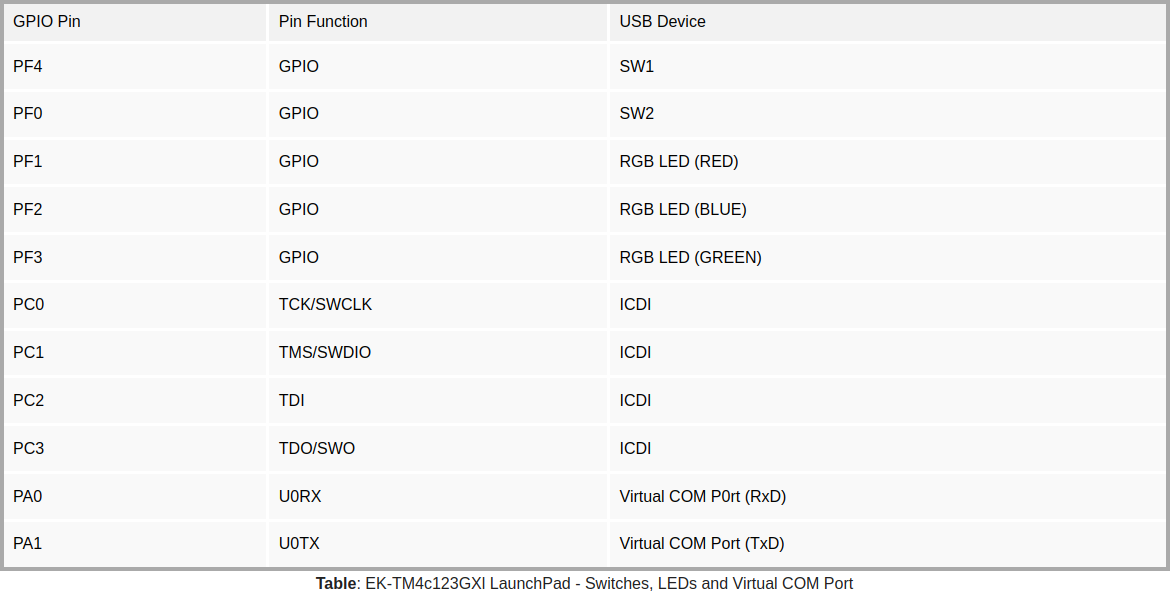
Readme first-

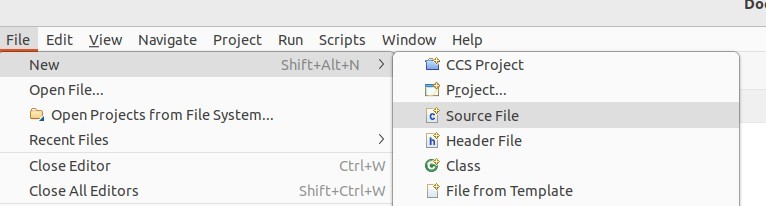
[https://www.ti.com/lit/ug/spmu296/spmu296.pdf?ts=1673930434318&ref\_url=https%253A%252F](https://www.ti.com/lit/ug/spmu296/spmu296.pdf?ts=1673930434318&ref_url=https%253A%252F%252Fwww.google.com%252F)

[%252Fwww.google.com%252F](https://www.ti.com/lit/ug/spmu296/spmu296.pdf?ts=1673930434318&ref_url=https%253A%252F%252Fwww.google.com%252F)

[Data sheet download for Tiva c series](https://www.ti.com/lit/ds/spms376e/spms376e.pdf?ts=1673868314076&ref_url=https%253A%252F%252Fwww.google.com%252F) [User Manual -Tiva c series](https://www.ti.com/lit/ug/spmu296/spmu296.pdf?ts=1673930434318&ref_url=https%253A%252F%252Fwww.google.com%252F)







Add files to your Project:

select File->New->Source File

copy the following code and paste it into the newly created main.c file: Blinky program:

#include <stdint.h>

#include "inc/tm4c123gh6pm.h" void delayMs(int n);

int main(void)

{ SYSCTL\_RCGC2\_R |= 0x00000020; */\* enable clock to GPIOF at clock gating control register \*/*

GPIO\_PORTF\_DIR\_R = 0x0E; */\* enable the GPIO pins for the LED (PF3, 2 1) as output \*/*

GPIO\_PORTF\_DEN\_R = 0x0E; */\* enable the GPIO pins for digital function*

*\*/*

while(1) {

GPIO\_PORTF\_DATA\_R = 0x0E; */\* turn on all LEDs \*/*

delayMs(500);

GPIO\_PORTF\_DATA\_R = 0; */\* turn off all LEDs \*/*

delayMs(500);

}

}

*/\* delay n milliseconds (16 MHz CPU clock) \*/*

void delayMs(int n)

{

int i, j;

for(i = 0 ; i < n; i++)

for(j = 0; j < 3180; j++) {} */\* do nothing for 1 ms \*/*

}

->Build your project &fix any errors

->Load Debug and Run