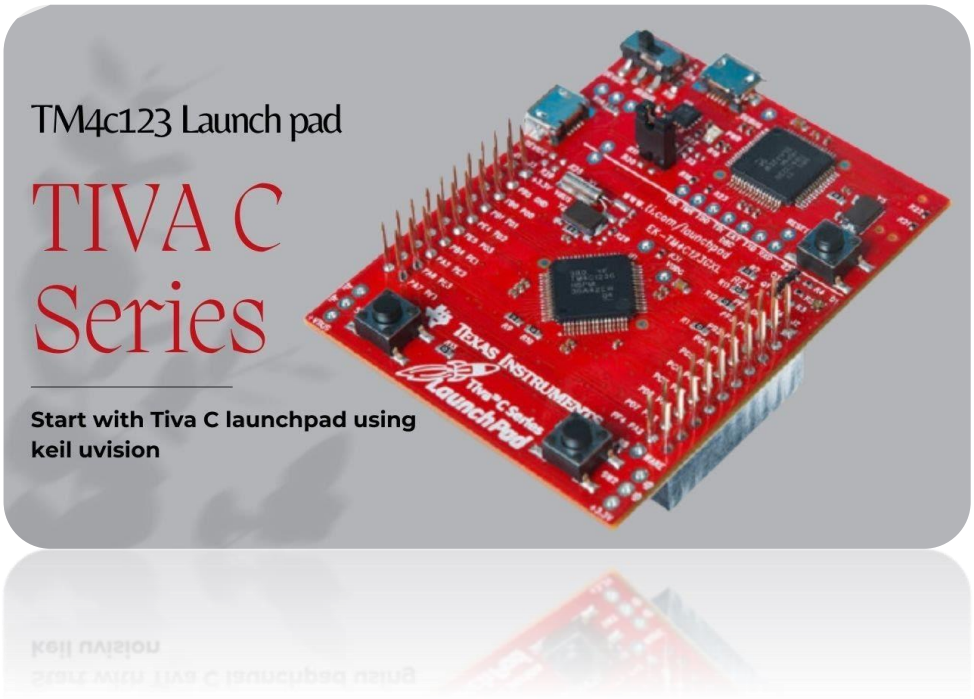
Tiva C Introduction and GPIO

Practical Lab 5



Objective:

In this Lab we are going to learn how to use ARM controllers and Tiva C launchpad

And interface with GPIO Peripheral

Requirements:

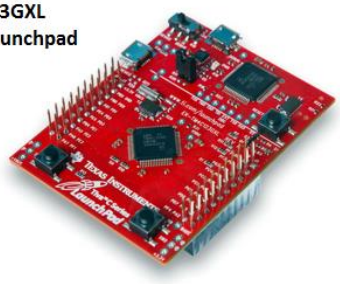
- **Software**:

1. Keil MDK

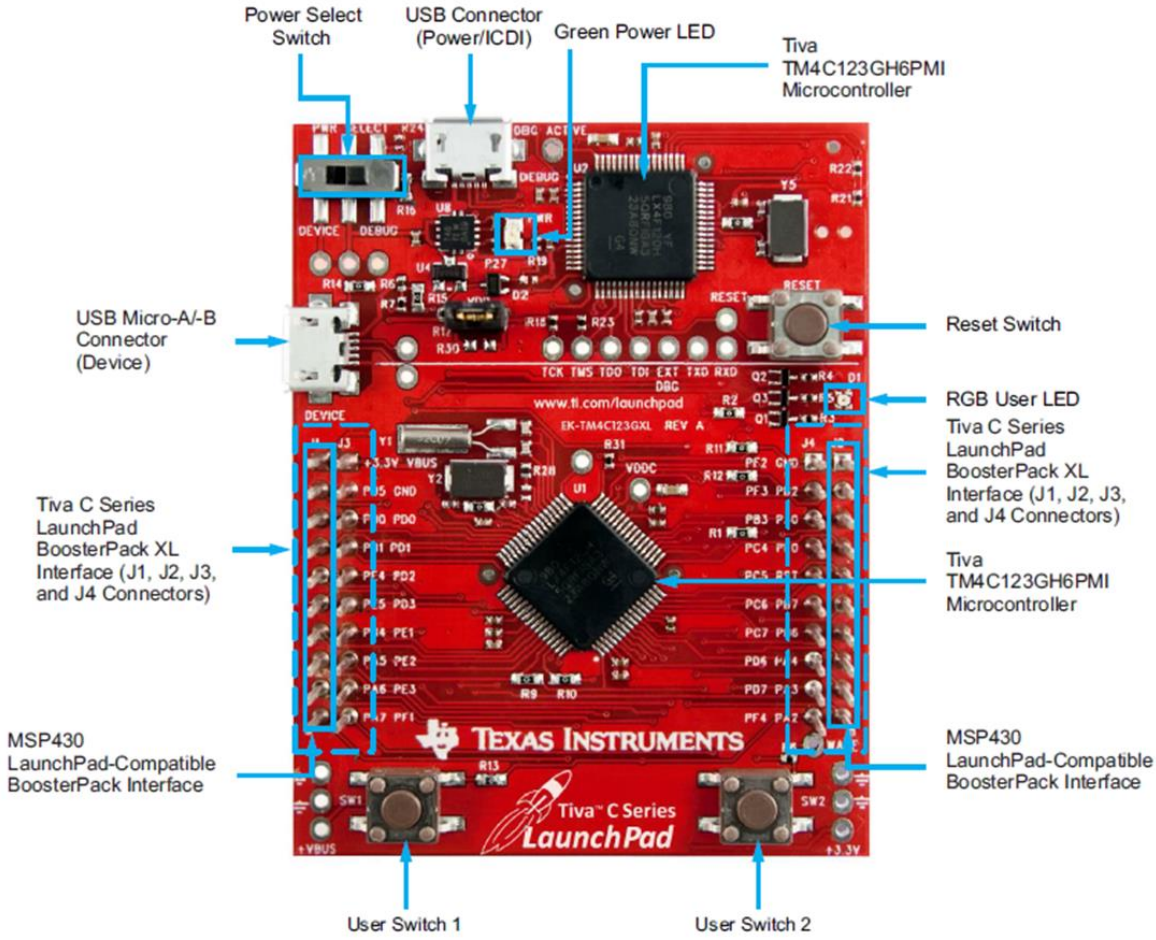
Let's Know the Tiva C Kit

There are two versions of tiva c

TM4C123GXL
Tiva C Launchpad



TM4C129xXL
Connected Launchpad



Tiva C Series TM4C123G LaunchPad Evaluation Board

Some of the features of this micro are as follows:

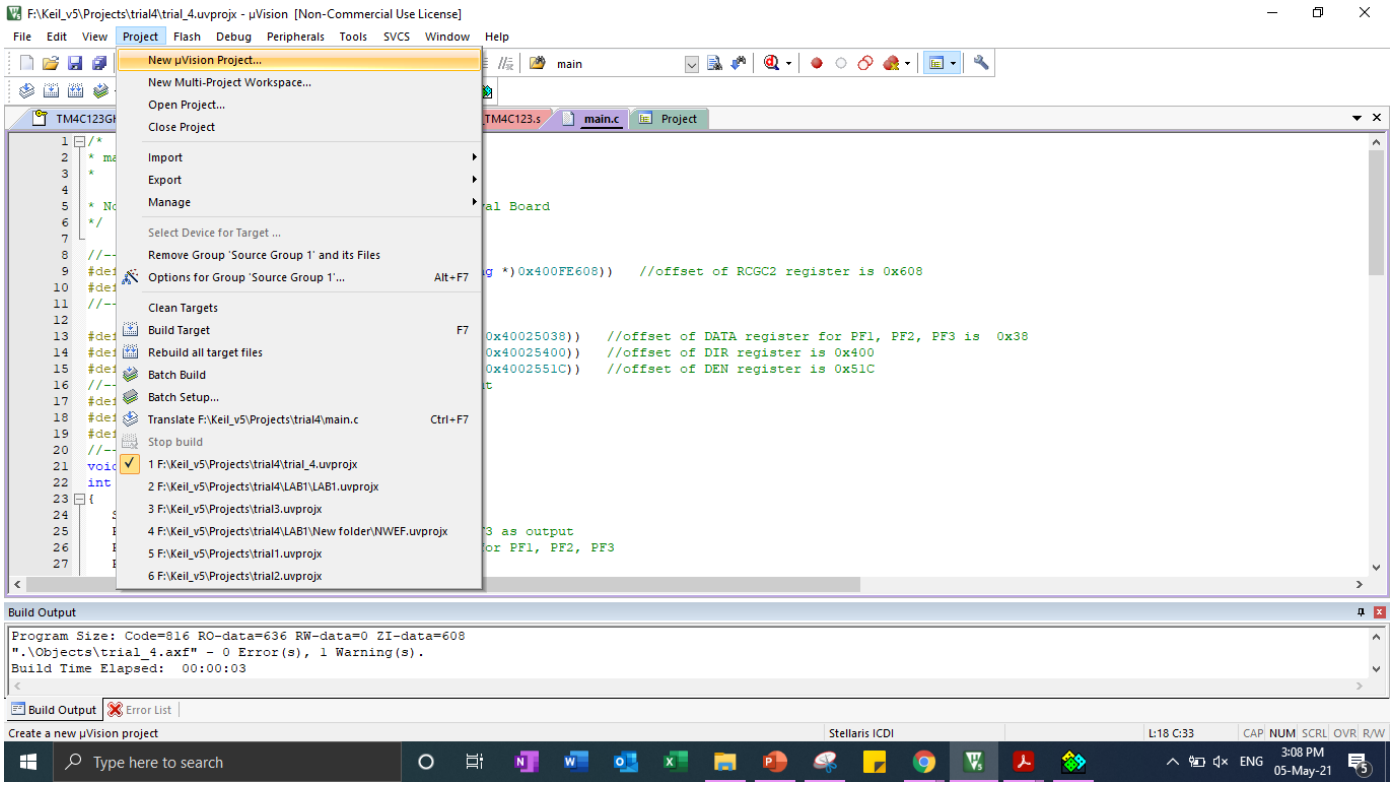
- Low Power 80 MHz (100 DMIPS) ARM Cortex-M4 CPU with Floating Point Unit
- 256KB Flash Memory, 32KB RAM, 2KB EEPROM
- 2 x 12-bit 1MSPS ADCs
- 2 x Analog Comparators
- Internal Temperature Sensor
- 6 x 64-bit and 6 x 32-bit Timers
- 16 x Motion PWM Channels
- 2 x Quadrature Encoders
- 8 x UART
- 4 x I2C
- 4 x SSI
- 2 x CAN

- 1 x USB 2.0 OTG/Host/Device
- 32 Channel DMA

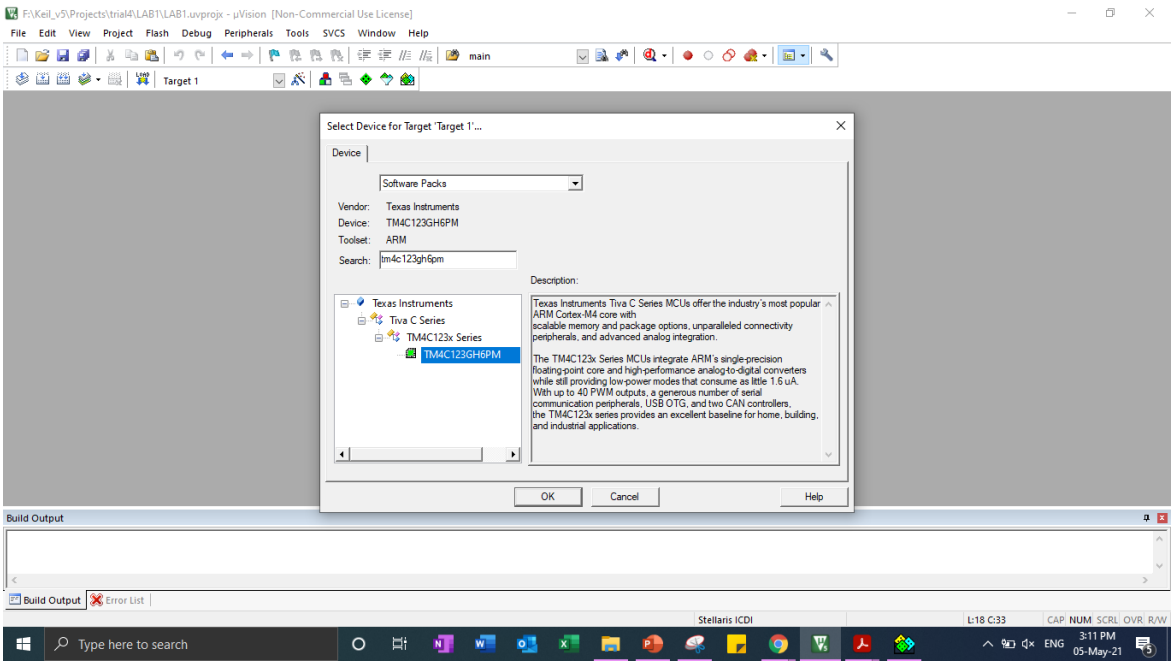
Experiment Steps:

1. **Open KEIL IDE:**

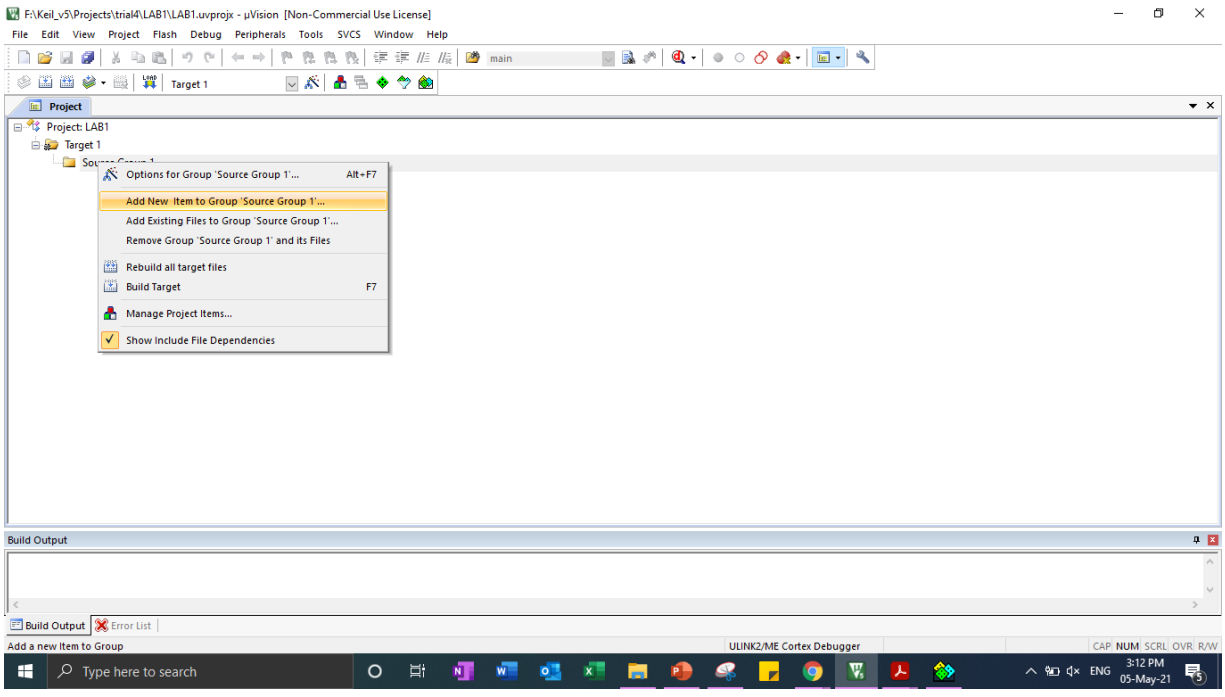
   a. Open Keil, create a new project.

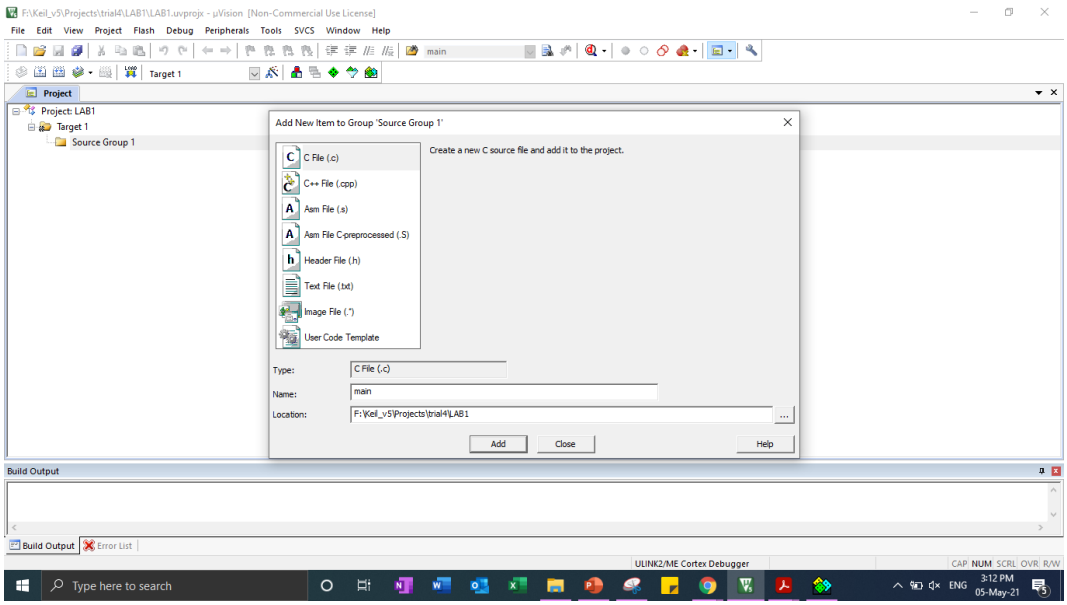


   b. Choose TM4c123gh6pm

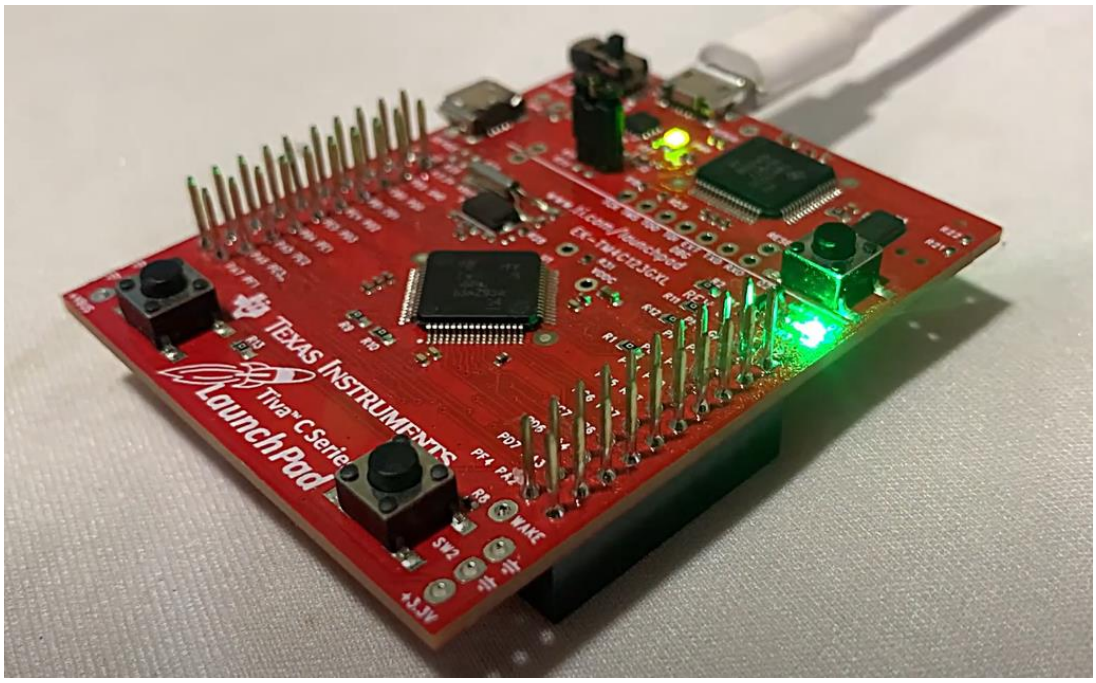


c. Create main.c .





Now WE Need to Blink the Green LED.



Code:

```
#define SYSCTL_RCGCGPIO_R (*(( volatile unsigned long *)0x400FE608 ) )
#define GPIO_PORTF_DATA_R (*(( volatile unsigned long *)0x40025038 ) )
#define GPIO_PORTF_DIR_R  (*(( volatile unsigned long *)0x40025400 ) )
#define GPIO_PORTF_DEN_R  (*(( volatile unsigned long *)0x4002551C ) )
void Delay(unsigned long i );

int main ( void )
{
SYSCTL_RCGCGPIO_R |= 0x20; // Enable clock for PORTF
GPIO_PORTF_DEN_R  = 0x0E;  // Enable PORTF Pin1, 2 and 3 as a digital pins
GPIO_PORTF_DIR_R  = 0x0E;  // Configure ORTF Pin1, 2 and 3 digital output pins

  while (1)
    {
    GPIO_PORTF_DATA_R |= 0x08; // turn on red LED
    Delay(2000000);
    GPIO_PORTF_DATA_R &= ~(0x08); // turn on red LED
    Delay(2000000);

    }
}

void Delay(volatile unsigned long i ){
  while(i--);
}
```

Introduction To Debugging

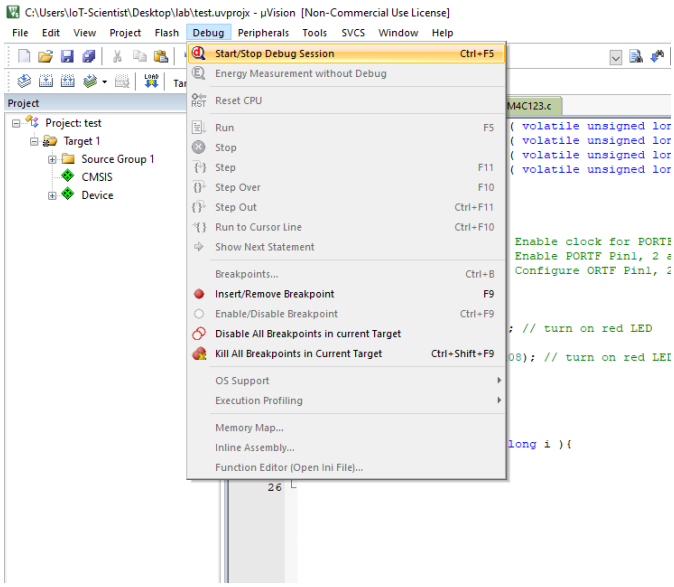
How to Debug :

There is A many methods to debug

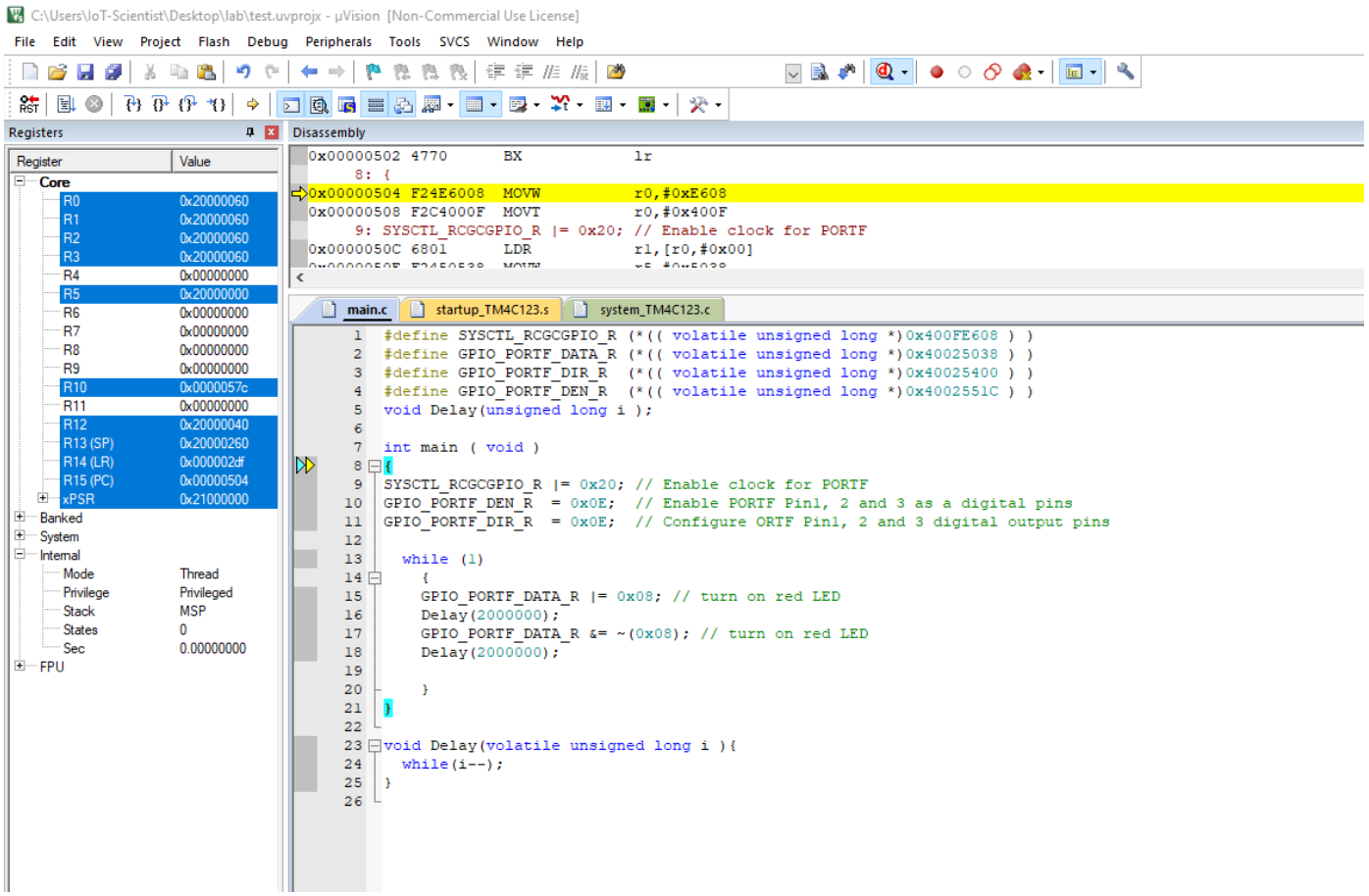
1- Step in
2- Step over
3- Step out
4- Break point

Steps

From **debug** menu choose start debugging or (F5)



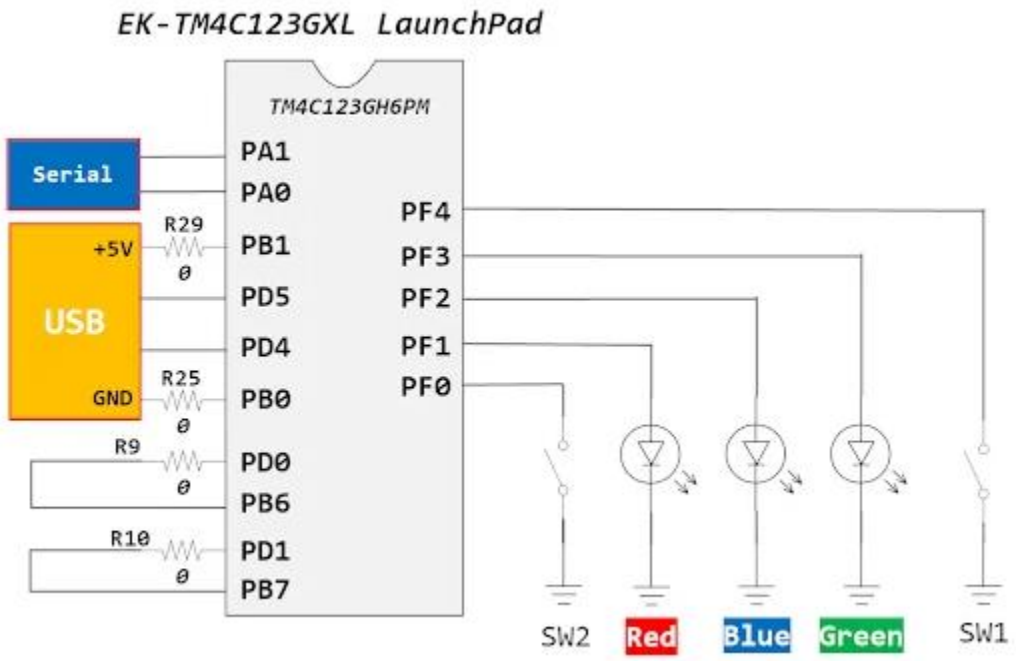
The view will change to debug view like this



Here you can run , stop , add break point , and run step by step

Toggle LED with Push Button

Note that Tiva c is connected with buttons with portf pint 0 & 4



Now we need to interface with button as You know From PIC lab :

We need to make lab to toggle green led when you press sw1

```c
# define SYSCTL_RCGCGPIO_R (*(( volatile unsigned long *)0x400FE608))
# define GPIO_PORTF_DATA_RD (*(( volatile unsigned long *)0x40025040))
# define GPIO_PORTF_DATA_WR (*(( volatile unsigned long *)0x40025020))
# define GPIO_PORTF_DIR_R (*(( volatile unsigned long *)0x40025400))
# define GPIO_PORTF_DEN_R (*(( volatile unsigned long *)0x4002551C))
# define GPIO_PORTF_PUR_R (*(( volatile unsigned long *)0x40025510 ))

# define SYSCTL_RCGC2_GPIOF 0x0020
# define GPIO_PORTF_PIN3_EN 0x08
# define GPIO_PORTF_PIN4_EN 0x10
# define SYSTEM_CLOC_FREQUENCY  16000000

#define DELAY_DEBOUNCE          SYSTEM_CLOC_FREQUENCY/1000
void Delay(volatile unsigned long counter);

int main ()
{
  static char flag = 0;
  SYSCTL_RCGCGPIO_R |= SYSCTL_RCGC2_GPIOF;
  GPIO_PORTF_DEN_R |= GPIO_PORTF_PIN3_EN +GPIO_PORTF_PIN4_EN;
  GPIO_PORTF_DIR_R |= GPIO_PORTF_PIN3_EN;
  GPIO_PORTF_DIR_R &= (~GPIO_PORTF_PIN4_EN);
  GPIO_PORTF_PUR_R |= GPIO_PORTF_PIN4_EN;

while(1)
{
  if(GPIO_PORTF_DATA_RD == 0)
  {
    Delay(DELAY_DEBOUNCE);
    if(( flag == 0) && (GPIO_PORTF_DATA_RD == 0))
    {
      GPIO_PORTF_DATA_WR ^= GPIO_PORTF_PIN3_EN;
      flag = 1;
    }
  }
  else
  {
    flag = 0;
  }
}
}
void Delay(volatile unsigned long counter)
{
  unsigned long i = 0;
  for(i = 0; i<counter; i++);
}
```

Lab report:

Submit a PDF file with Code, snapshots and “Small Video for the practical work” of the work you did and upload the project file.