Home Work 3

ENGR 844

SFSU

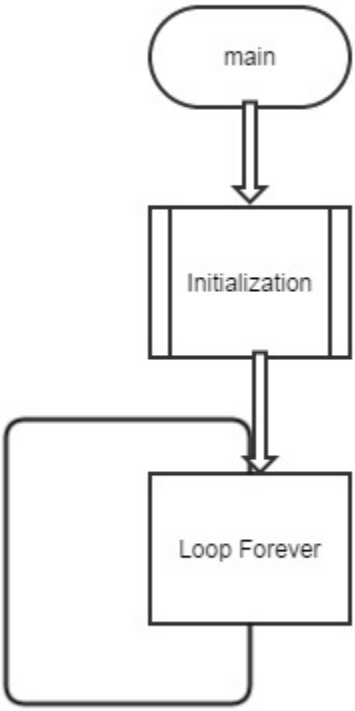
2014

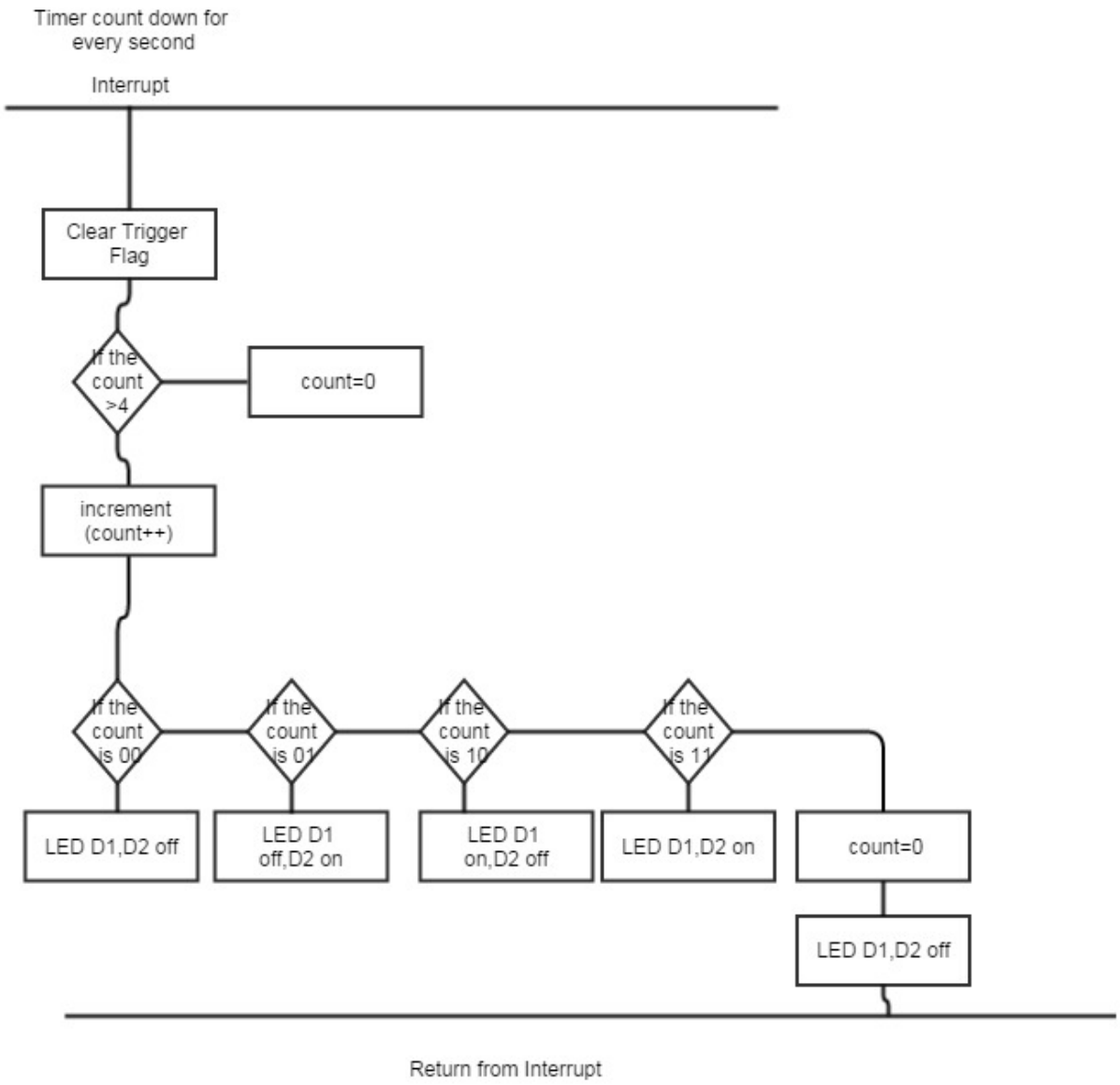
Authored by: Ravi Teja Mamidipaka

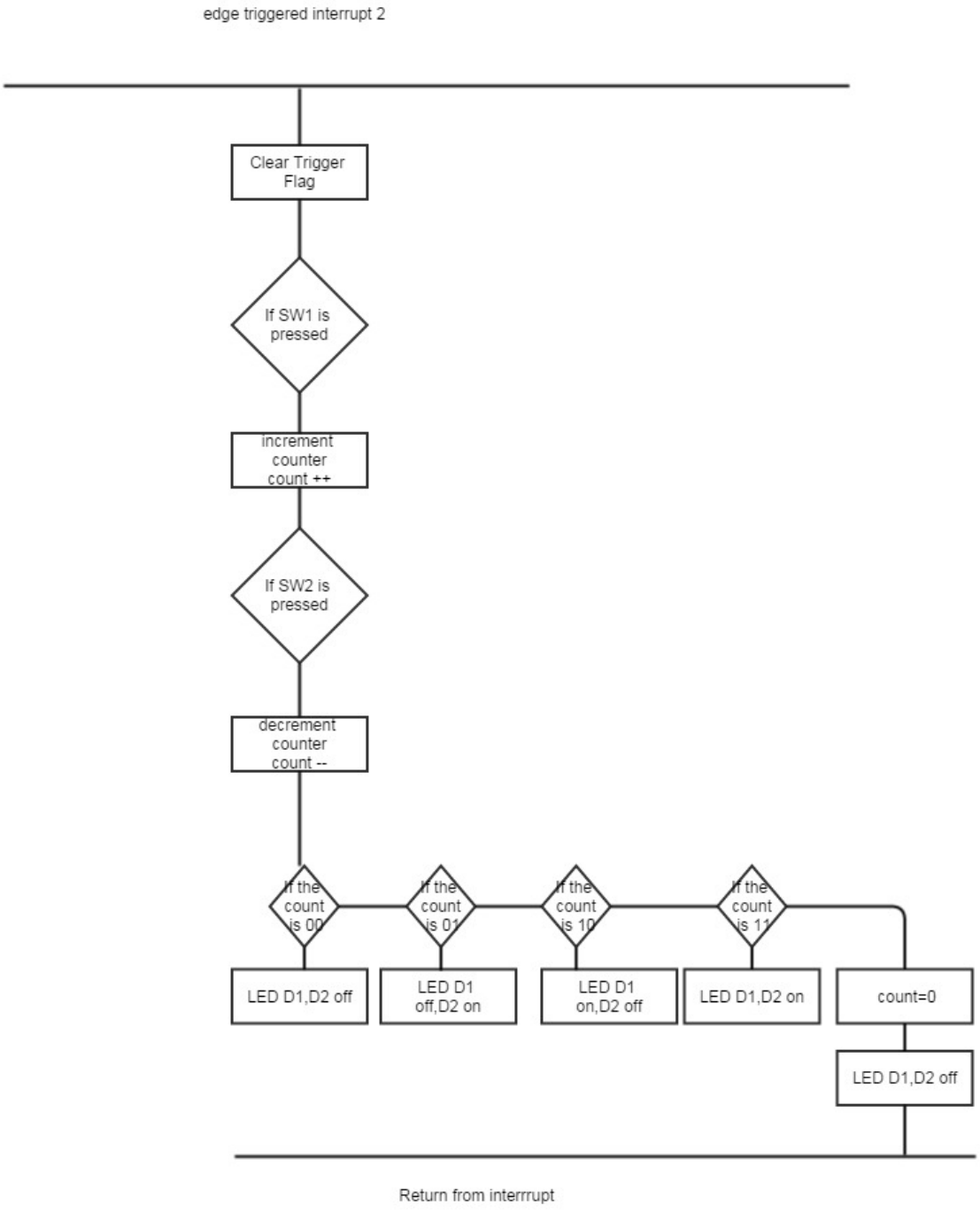
Home Work 3

ENGR 844

FLOW CHART




Timer count down for every second
Interrupt
Clear Trigger Flag
If the count >4
count=0
increment (count++)
If the count is 00
If the count is 01
If the count is 10
If the count is 11
LED D1,D2 off
LED D1 off,D2 on
LED D1 on,D2 off
LED D1,D2 on
count=0
LED D1,D2 off
Return from Interrupt


edge triggered interrupt 2
Clear Trigger Flag
If SW1 is pressed
increment counter count ++
If SW2 is pressed
decrement counter count --
If the count is 00
If the count is 01
If the count is 10
If the count is 11
LED D1,D2 off
LED D1 off,D2 on
LED D1 on,D2 off
LED D1,D2 on
count=0
LED D1,D2 off
Return from interrrupt

Procedure of Development

- Determine the specifications of system find the appropriate pins for the system.
- Use the pin mux utility to initialize the pins as output or input as per requirement.
- Configure the interrupts as per requirements.
- This system has two interrupts so the priority of timer0A is 0 and the edge triggered input is 2.
- The main objective of this problem is to increment counter for only 4 values 0,1,2,3.
- Each number operates as LED as its bits. LED D1 represent 0 bit and D2 represents 1 bit.
- We configure LED's respectively as per the number in counter.
- The edge triggered interrupt is initiated by the two switches on board SW1, SW2
- SW1 increases the count of counter, SW2 decreases the count in counter.
- The counter only counts 0-1-2-3 and it returns to 0 after it completes.
- With all these requirements draw the flow chart required for the problem.
- With drawing the flow chart we are done with problem transform it into code.

Source Code:

```
#include <stdint.h>
#include <stdbool.h>
#include "homework_3.h"
#include "inc/hw_types.h"
#include "inc/hw_memmap.h"
#include "inc/hw_gpio.h"
#include "driverlib/sysctl.h"
#include "driverlib/pin_map.h"
#include "driverlib/rom_map.h"
#include "driverlib/gpio.h"
#include "driverlib/interrupt.h"
#include "driverlib/timer.h"
#include "inc/tm4c1294ncpdt.h"

//*****************************************************************************
volatile uint8_t count;
void
PortFunctionInit(void)
{
    //
    // Enable Peripheral Clocks
    //
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOJ);
    MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPION);

    //
    // Enable pin PJ0 for GPIOInput
```

```
    //
    MAP_GPIOPinTypeGPIOInput(GPIO_PORTJ_AHB_BASE, GPIO_PIN_0);

    //
    // Enable pin PJ1 for GPIOInput
    //
    MAP_GPIOPinTypeGPIOInput(GPIO_PORTJ_AHB_BASE, GPIO_PIN_1);

    //
    // Enable pin PN1 for GPIOOutput
    //
    MAP_GPIOPinTypeGPIOOutput(GPIO_PORTN_BASE, GPIO_PIN_1);

    //
    // Enable pin PN0 for GPIOOutput
    //
    MAP_GPIOPinTypeGPIOOutput(GPIO_PORTN_BASE, GPIO_PIN_0);

                //
                //Enable Pullup for PJ0 PJ1
                GPIO_PORTJ_AHB_PUR_R |= 0x03;
}
void
Interrupt_Init(void)
{
  IntEnable(INT_GPIOJ);                                         // enable interrupt 51
in NVIC (GPIOJ)
      IntPrioritySet(INT_GPIOJ, 0x02);              // configure GPIOJ interrupt priority as 2
      GPIO_PORTJ_AHB_IM_R |= 0x03;                  // arm interrupt on PJ0,PJ1
      GPIO_PORTJ_AHB_IS_R &= ~0x03;     // PJ0,PJ1 is edge-sensitive
  GPIO_PORTJ_AHB_IBE_R &= ~0x03;    // PJ0,PJ1 are not both edge-triggered.
  GPIO_PORTJ_AHB_IEV_R &= ~0x03;  // PJ0,PJ1 falling edge event
      IntMasterEnable();                    // globally enable interrupt
}
void Timer0A_Init(unsigned long period)
{
      //
  // Enable Peripheral Clocks
  //
  SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);
  TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC);                  // configure for 32-bit
timer mode
  TimerLoadSet(TIMER0_BASE, TIMER_A, period -1);      //reload value
      IntPrioritySet(INT_TIMER0A, 0x00);
      // configure Timer0A interrupt priority as 0
  IntEnable(INT_TIMER0A);
                                  // enable interrupt 19 in NVIC (Timer0A)
```

```
    TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);    // arm timeout interrupt
  TimerEnable(TIMER0_BASE, TIMER_A);                                                    //
enable timer0A
}

//interrupt handler for Timer0A

void Timer0A_Handler(void)
{
        // acknowledge flag for Timer0A timeout
        TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
    if(count>4)
    {
        count=0;
    }
    else{
                    count++;
    }

    if(count==0)
                    {
                        GPIO_PORTN_DATA_R =0x00;
                        //LED D1,D2 off
                    }
                    else if(count==1)
                    {
                        GPIO_PORTN_DATA_R =0x01;
                        //LED D1 on,D2 off
                    }
                    else if(count==2)
                    {
                        GPIO_PORTN_DATA_R =0x02;
                    }else if(count==3)
                    {
                        GPIO_PORTN_DATA_R =0x03;
                        //LED D1,D2 on
                    }else
                    {
                        count=0;
                        GPIO_PORTN_DATA_R= 0x00;
                        //LED D1,D2 off
                    }


}

//interrupt handler
```

```
void GPIOPortJ_Handler(void)
{

    //SW1 is pressed
    if(GPIO_PORTJ_AHB_RIS_R&0x01)
    {
            // acknowledge flag for PJ0
            GPIOIntClear(GPIO_PORTJ_AHB_BASE, GPIO_PIN_0);
            //counter imcremented by 1
            count++;
    }

    //SW2 is pressed
 if(GPIO_PORTJ_AHB_RIS_R&0x02)
    {
            // acknowledge flag for PJ1
            GPIOIntClear(GPIO_PORTJ_AHB_BASE, GPIO_PIN_1);
            //counter imcremented by 1
            count--;
    }
    if(count==0)
                        {
                                GPIO_PORTN_DATA_R =0x00;
                                //LED D1,D2 off
                        }
                        else if(count==1)
                        {
                                GPIO_PORTN_DATA_R =0x01;
                                //LED D1 off,D2 on
                        }
                        else if(count==2)
                        {
                                GPIO_PORTN_DATA_R =0x02;
                                //LED D1 on ,D2 off
                        }else if(count==3)
                        {
                                GPIO_PORTN_DATA_R =0x03;
                                //LED D1,D2 on
                        }else
                        {
                                count=0;
                                GPIO_PORTN_DATA_R= 0x00;
                                //LED D1,D2 off
                        }


}
```

```
int main(void)
    {
                unsigned long period = 16000000; //reload value to Timer0A to generate a
second delay
                PortFunctionInit();
                //intialize ports
                Timer0A_Init(period);
                // Load the timer with value
                Interrupt_Init();
                //
                //loop forever
                //
                while(1)
                {
                }
}
```

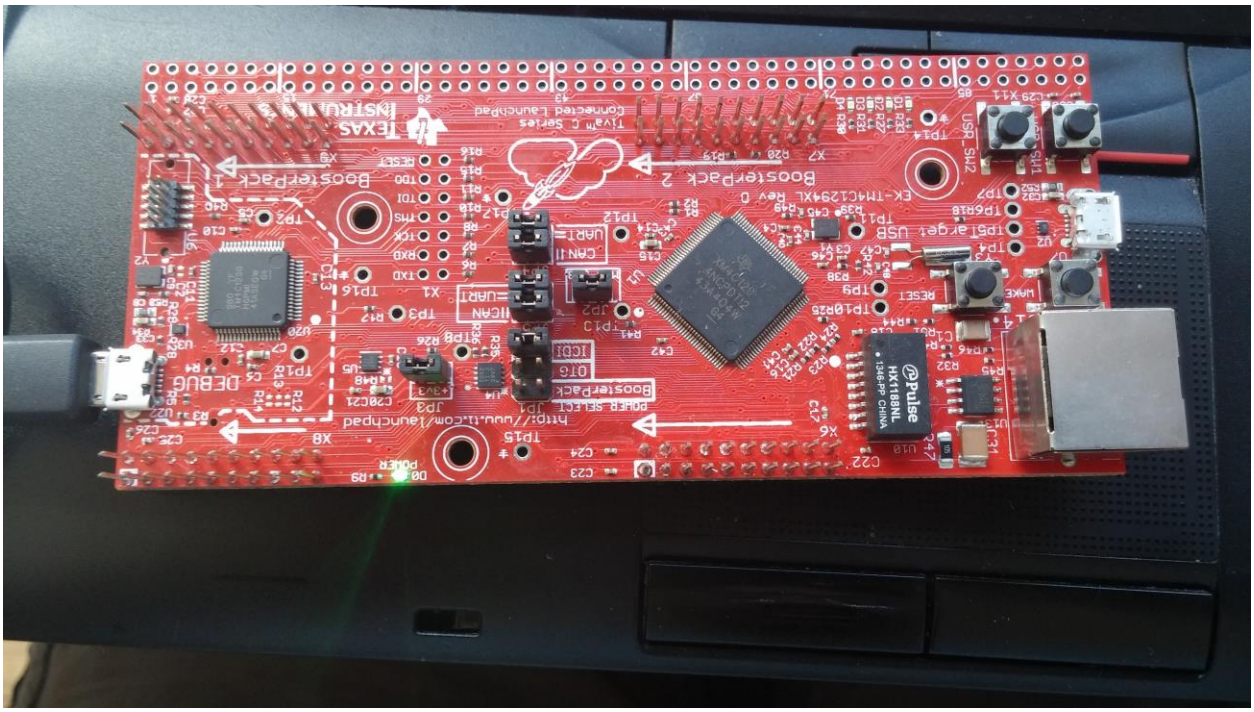
Launch PAD Video after dumping source code.

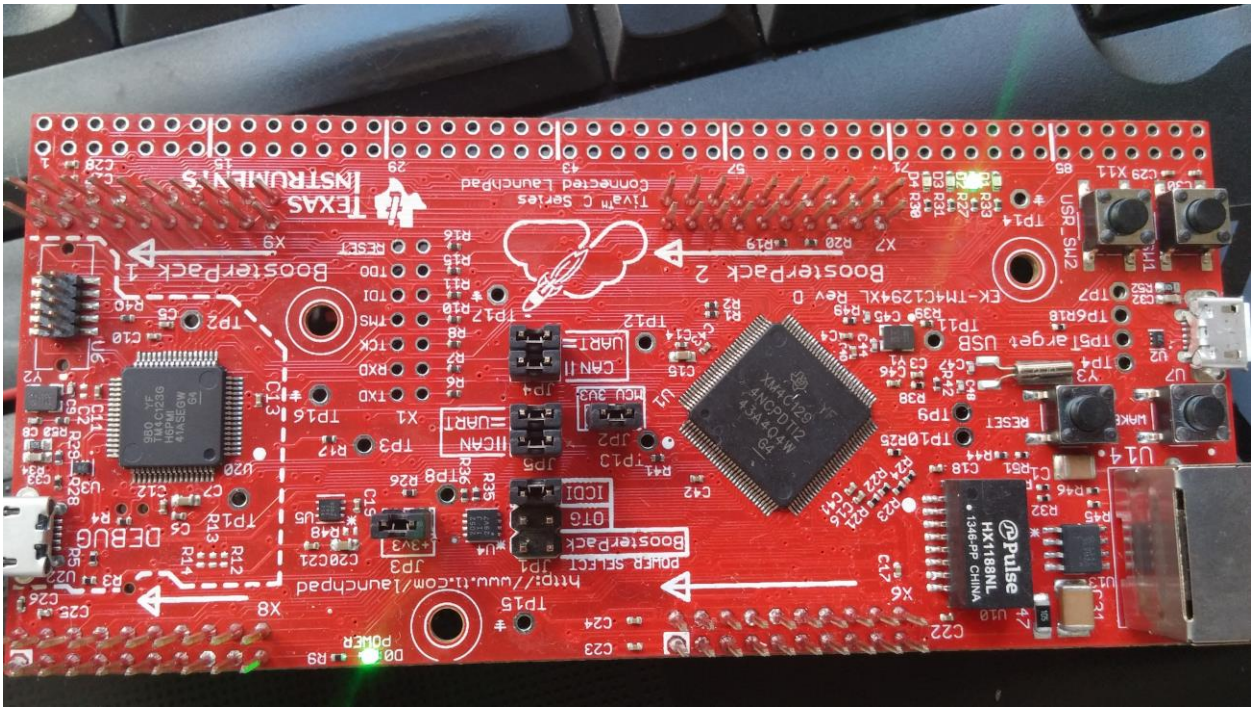
https://www.dropbox.com/s/embz9vt6bxmvpx4/20141018_133113.mp4?dl=0

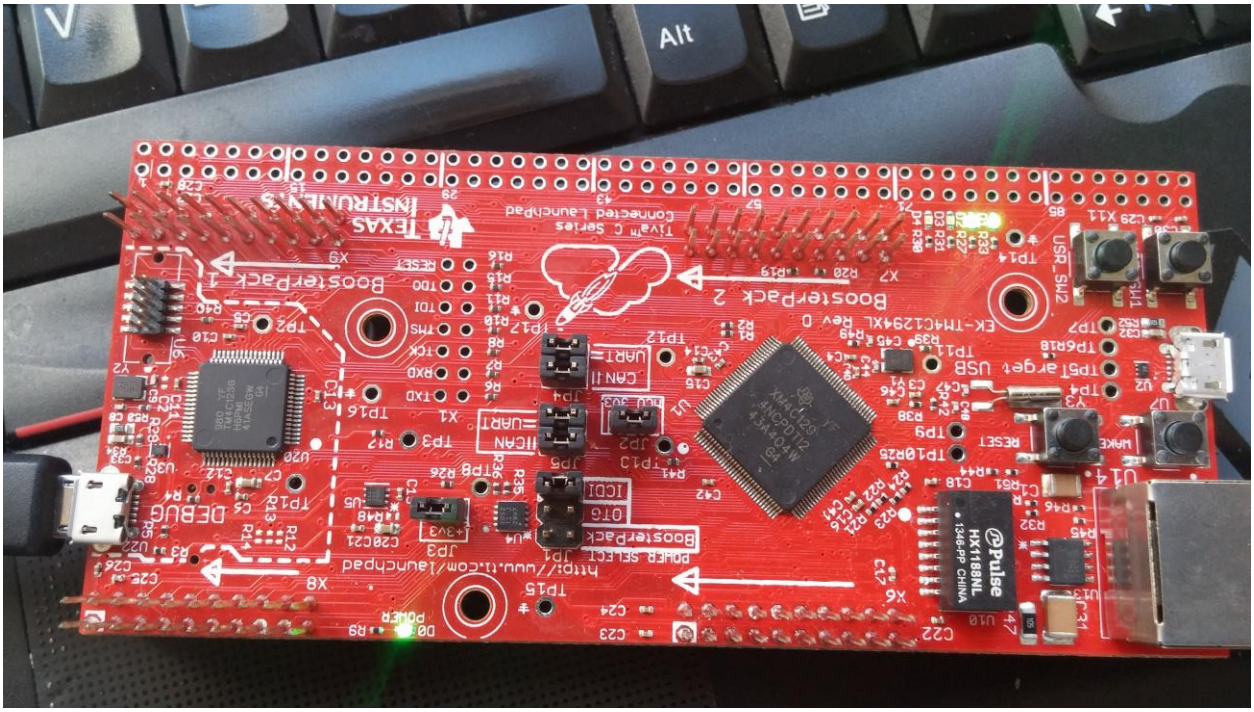
Video of the count function in watch window

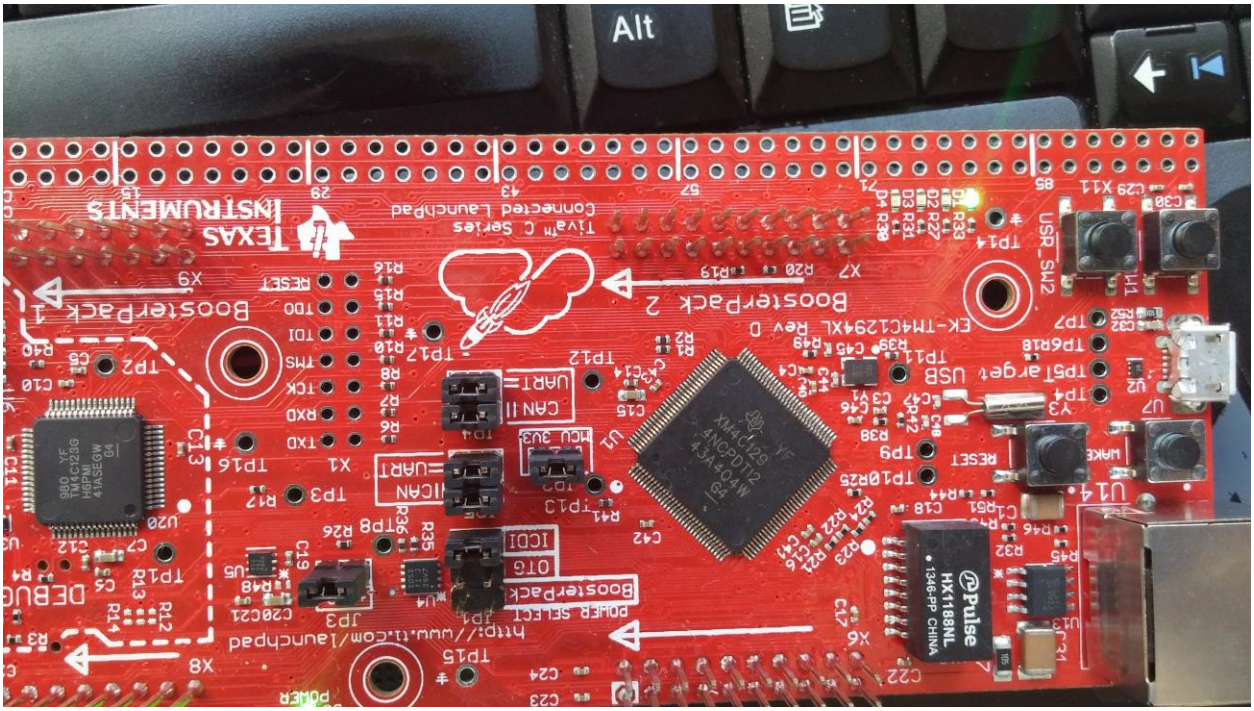
https://www.dropbox.com/s/xjxm3xl7tr5x04k/2014-10-18_13-48-39.MP4?dl=0

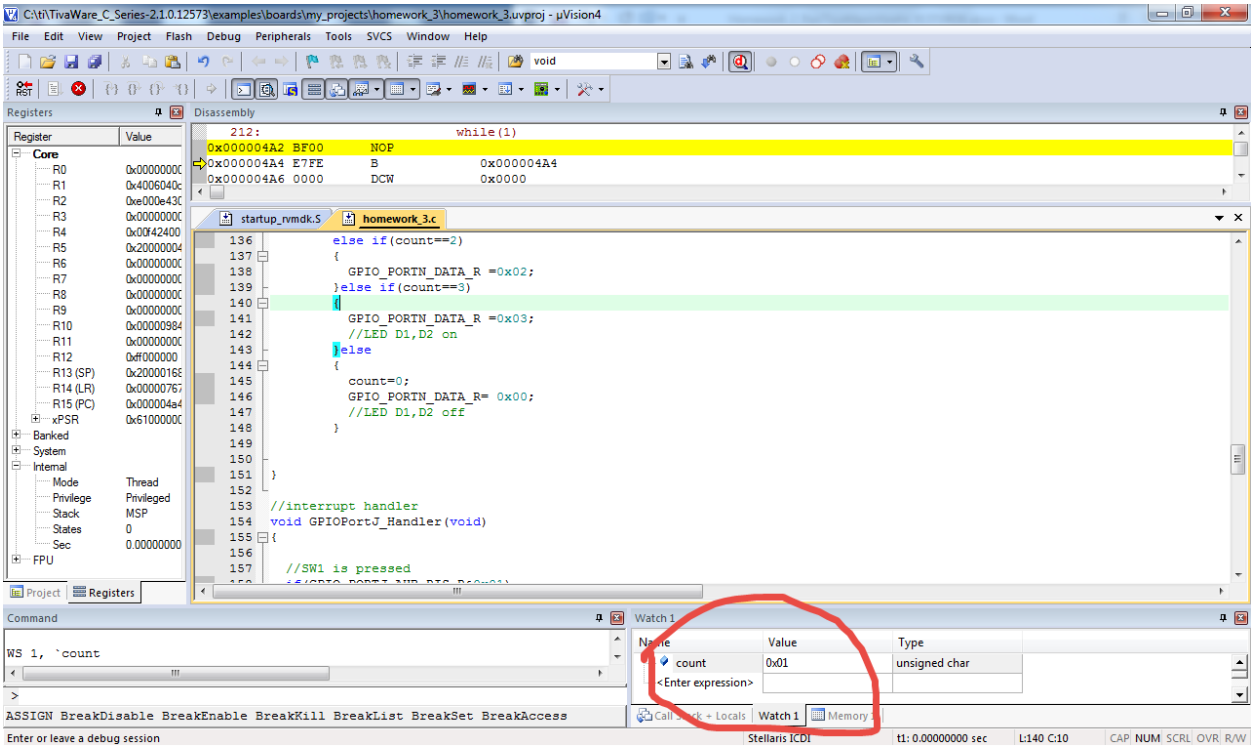
d1 d2 0 0 0 1 10 11

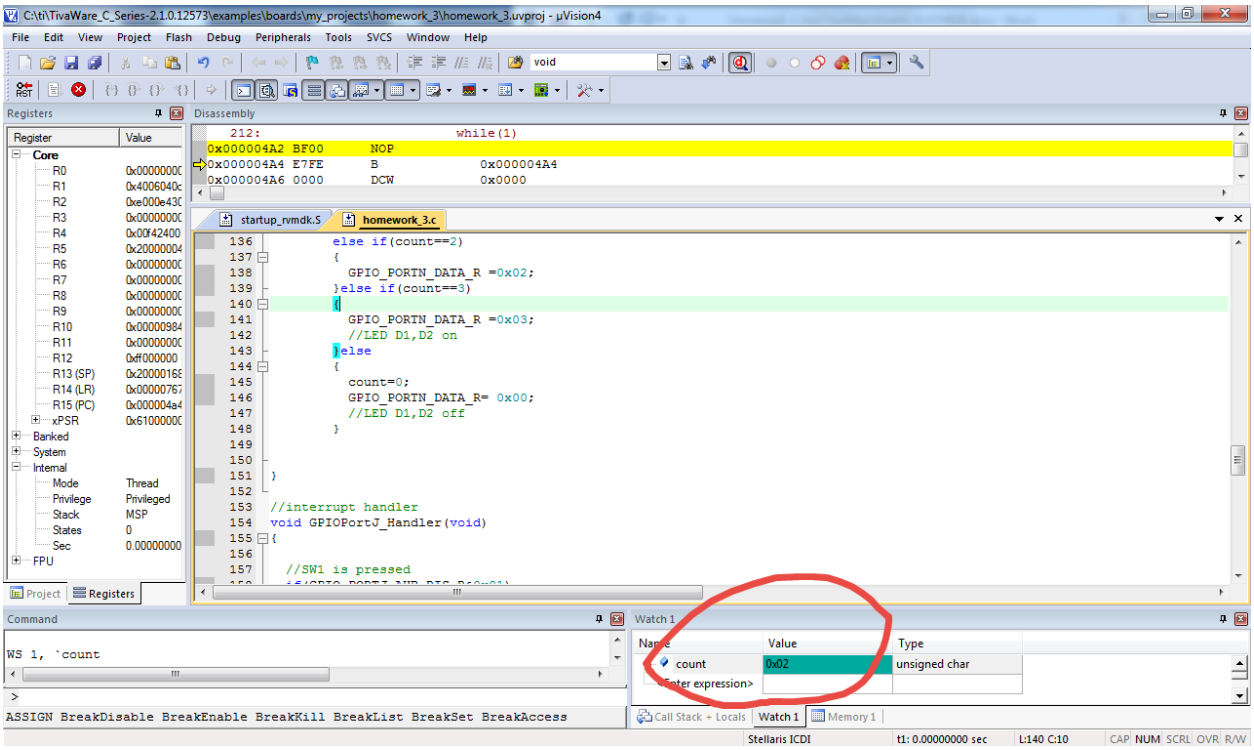


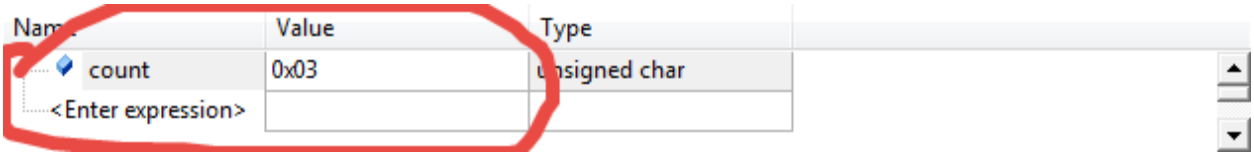



TEXAS INSTRUMENTS
Tiva™ C Series
Connected LaunchPad
BoosterPack 1
BoosterPack 2
EK-TM4C1294XL Rev D
USR_SW2
USB Target
DEBUG
UART
CAN
I2C
OTG
POWER SELECT
http://www.ti.com/launchpad
HX1188NL
1346-PP CHINA


TEXAS INSTRUMENTS
Tiva™ C Series
Connected LaunchPad
BoosterPack 1
BoosterPack 2
EK-TM4C1294XL Rev D
USR_SW2
USB Target
DEBUG
UART
CAN
I2C
OTG
POWER SELECT
http://www.ti.com/launchpad
HX1188NL
1346-PP CHINA


C:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\my_projects\homework_3\homework_3.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
startup_rvmdk.S
homework_3.c
Command
Watch 1
Name
Value
Type
count
0x01
unsigned char
<Enter expression>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Enter or leave a debug session
Stellaris ICDI


C:\ti\TivaWare_C_Series-2.1.0.12573\examples\boards\my_projects\homework_3\homework_3.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
startup_rvmdk.S
homework_3.c
Command
Watch 1
Name
Value
Type
count
0x02
unsigned char
<Enter expression>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
Stellaris ICDI


Value
Type
count
0x03
<Enter expression>

Discussion:

- After going through lecture 4, 5 we can do most of the problem with ease.
- The sample programs toggle and switch delay were the most helpful resources for the project.