Ayush Goyal

190905522 CSE D 62

DAA Lab-6 (Week 6) – Interfacing LED to ARM Microcontroller

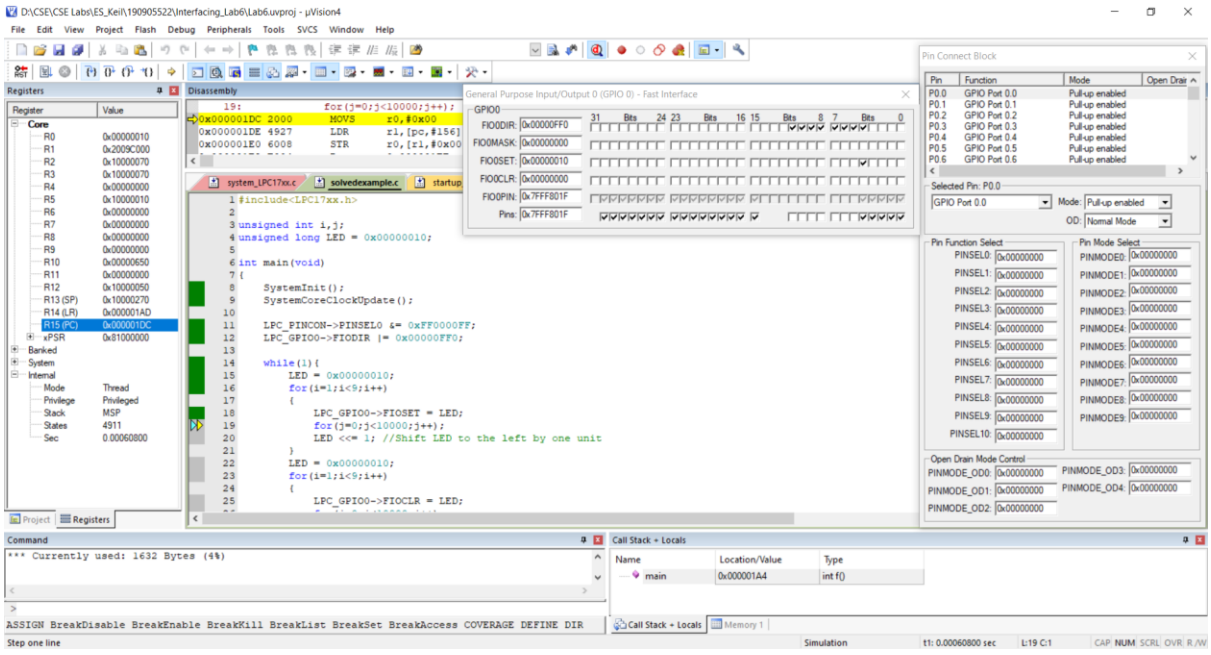
Solved Exercise: Write a program to turn on/off the LEDs serially.

CODE:

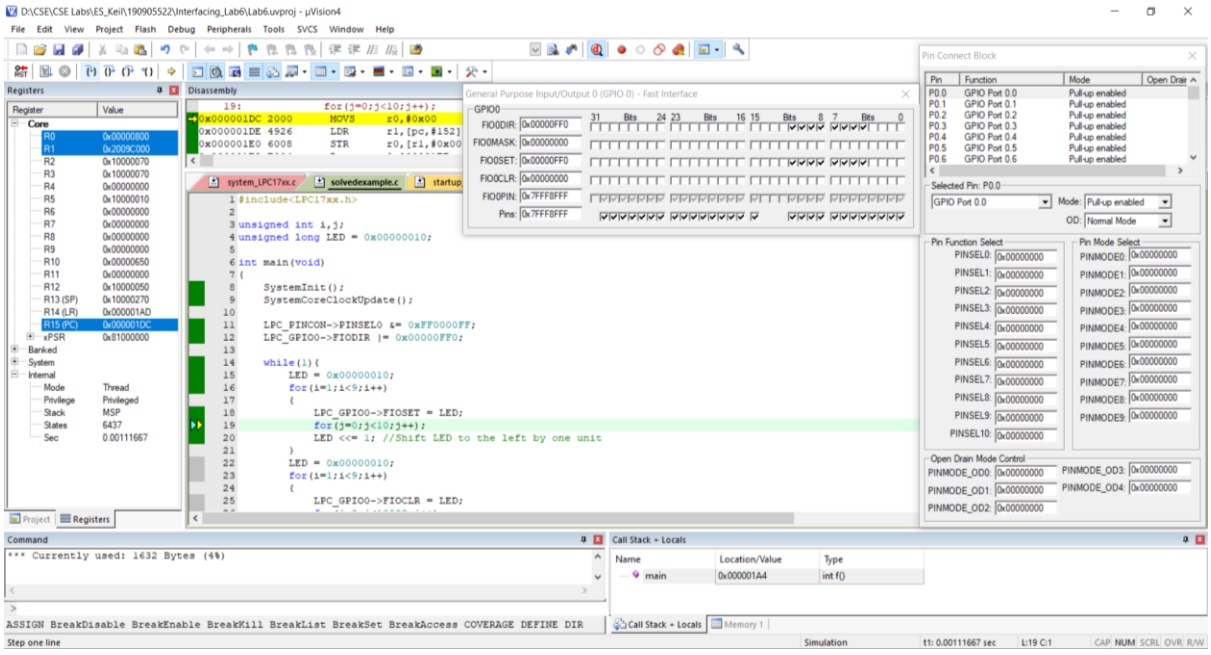
```
#include<LPC17xx.h>

unsigned int i,j;
unsigned long LED = 0x00000010;

int main(void)
{
    SystemInit();
    SystemCoreClockUpdate();

    LPC_PINCON->PINSEL0 &= 0xFF0000FF;
    LPC_GPIO0->FIODIR |= 0x00000FF0;

    while(1){
        LED = 0x00000010;
        for(i=1;i<9;i++)
        {
            LPC_GPIO0->FIOSET = LED;
            for(j=0;j<10000;j++);
            LED <<= 1; //Shift LED to the left by one unit
        }
        LED = 0x00000010;
        for(i=1;i<9;i++)
        {
            LPC_GPIO0->FIOCLR = LED;
            for(j=0;j<10000;j++);
            LED<<=1;
        }
    }
}
```

OUTPUT:

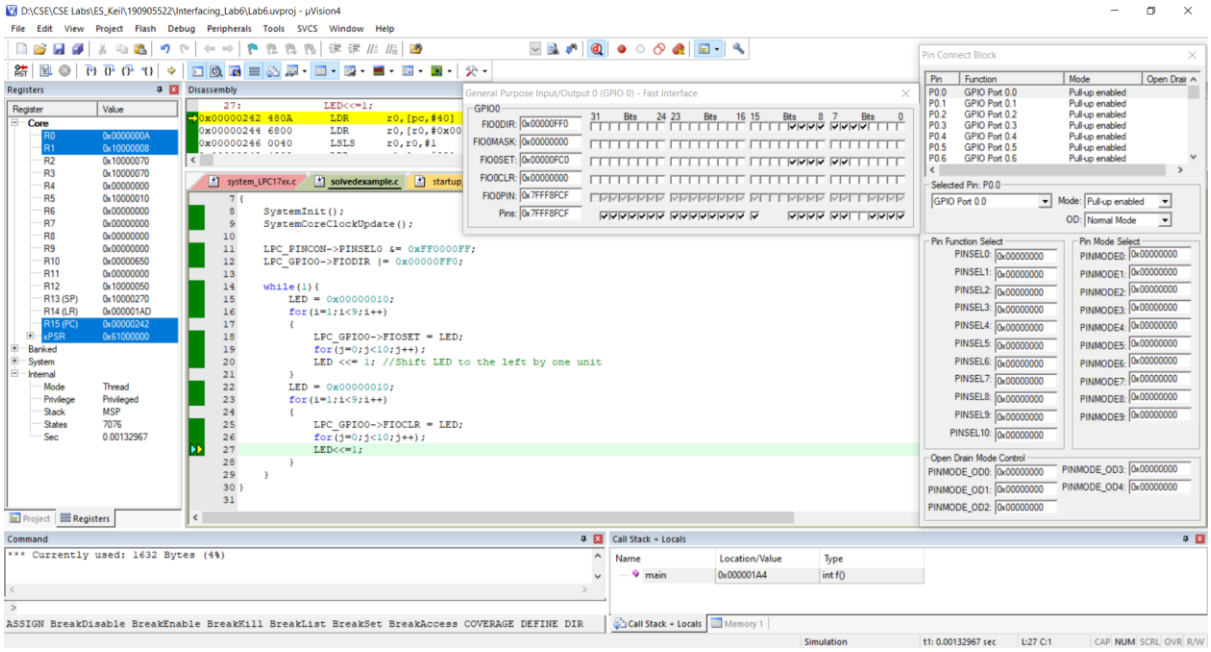
After the first LED is turned on:



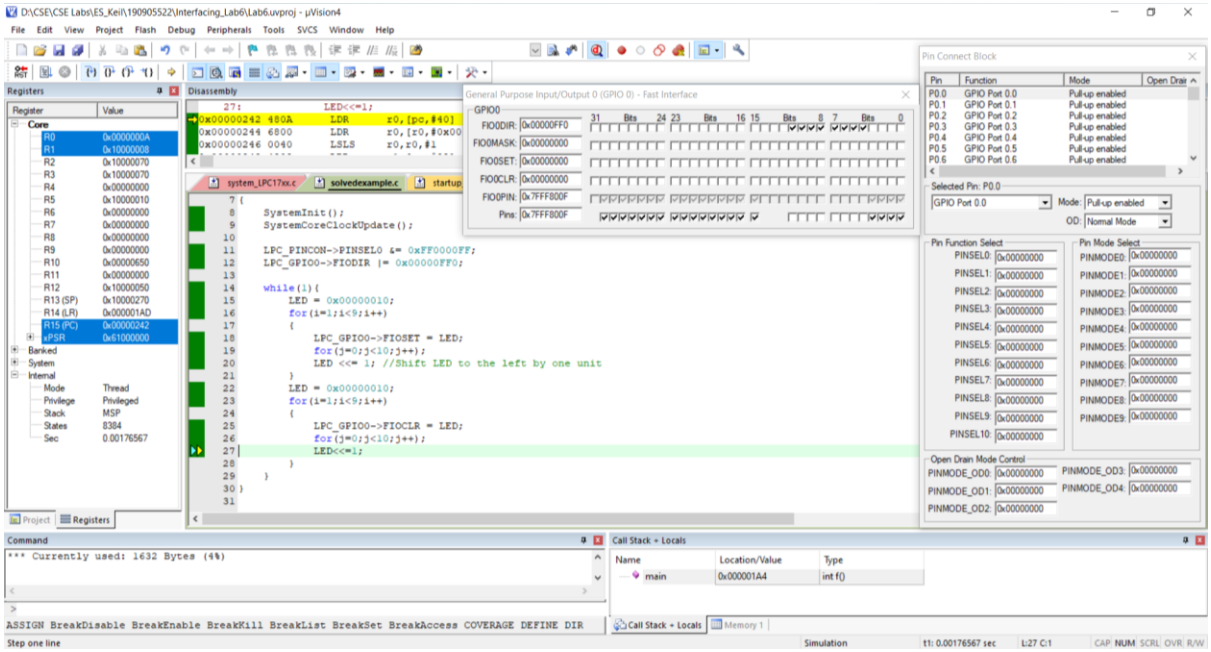
After the 8th LED is turned on:



After 1st and 2nd LED is turned off:



After 8 LEDs are turned off:



Thus, we can see that this while loop iterates continuously serially turning on and off the LEDs.

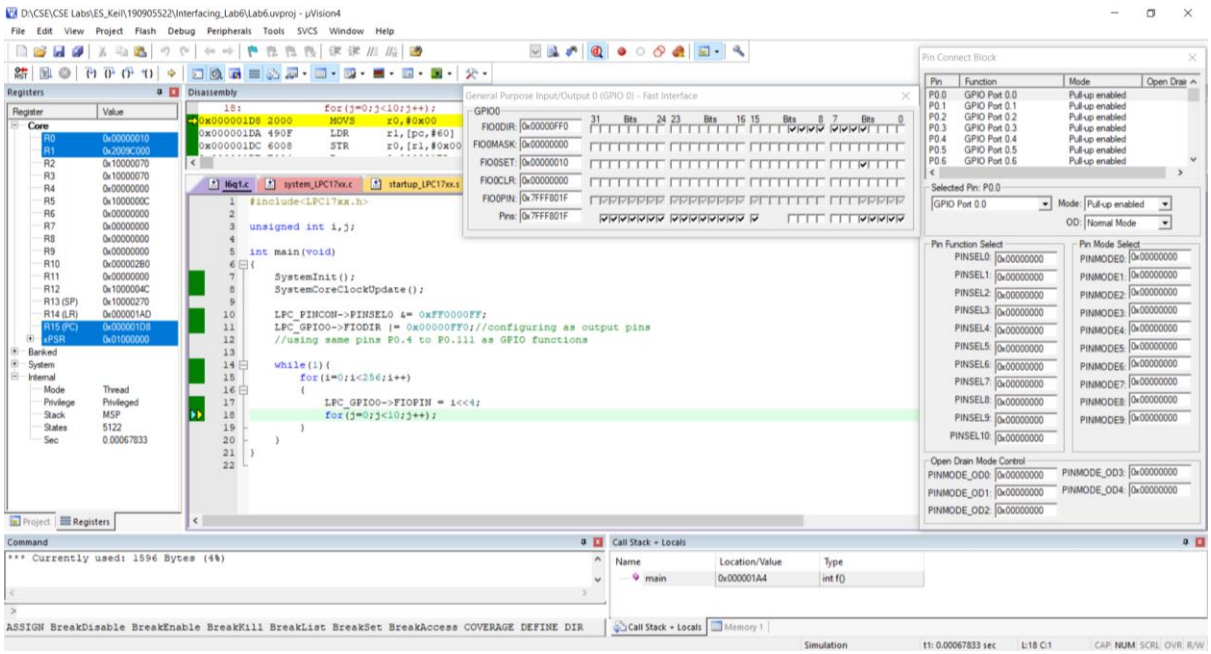
1) **Write a C program to display an 8-bit binary up counter on the LEDs.**

CODE:

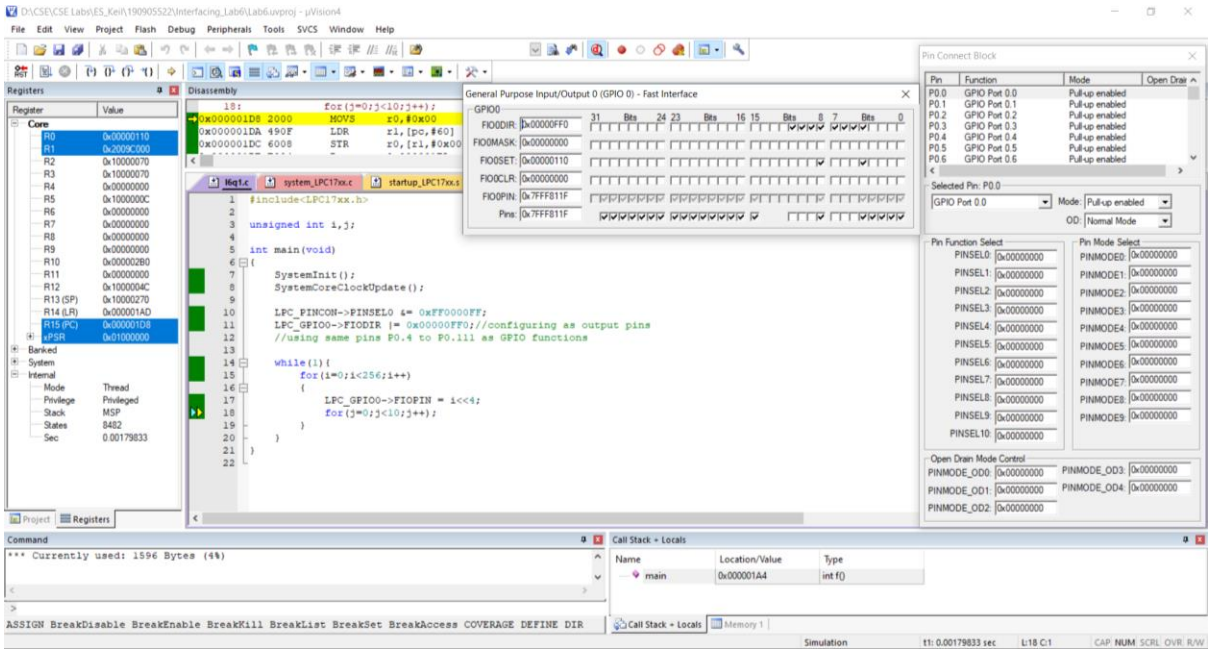
```
#include<LPC17xx.h>

unsigned int i,j;

int main(void)
{
    SystemInit();
    SystemCoreClockUpdate();

    LPC_PINCON->PINSEL0 &= 0xFF0000FF;
    LPC_GPIO0->FIODIR |= 0x00000FF0;//configuring as output pins
    //using same pins P0.4 to P0.11 as GPIO functions

    while(1){
        for(i=0;i<256;i++)
        {
            LPC_GPIO0->FIOPIN = i<<4;
            for(j=0;j<10;j++);
        }
    }
}
```

OUTPUT:

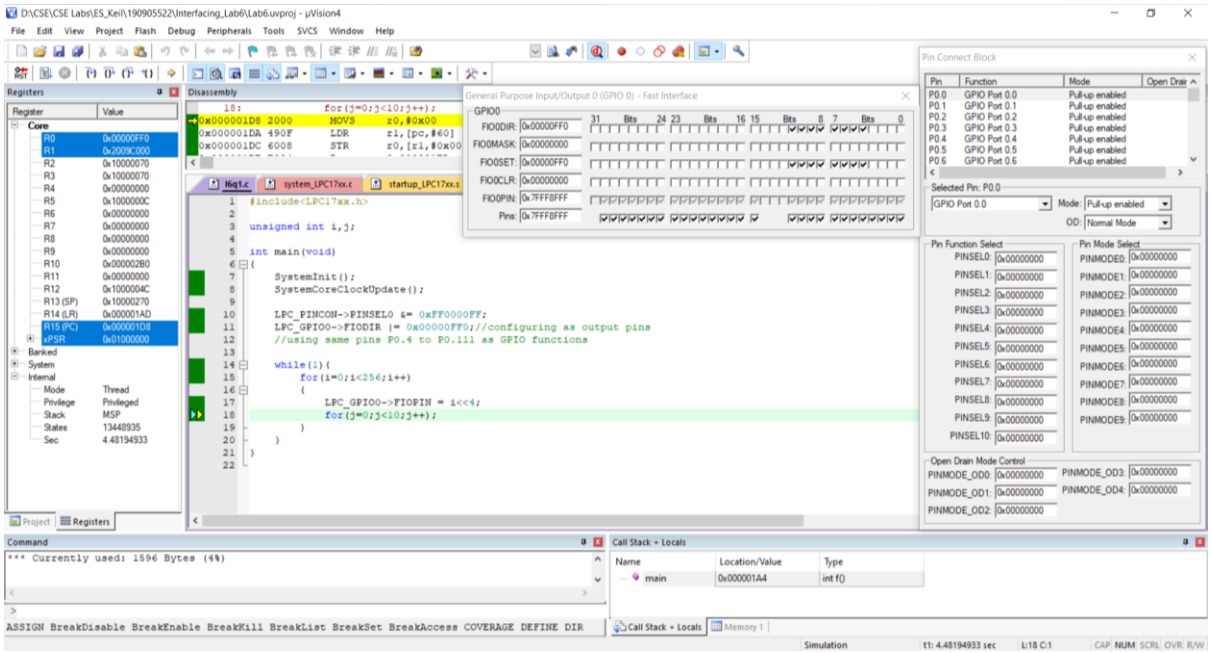
After 1st count it should be just 4th bit on:



After 18 counts or iterations, the binary counter should be 10001:



After 255 counts, the binary up counter should be at value 11111111:



Thus, we can see the functioning of an 8-bit binary up counter on the LEDs.

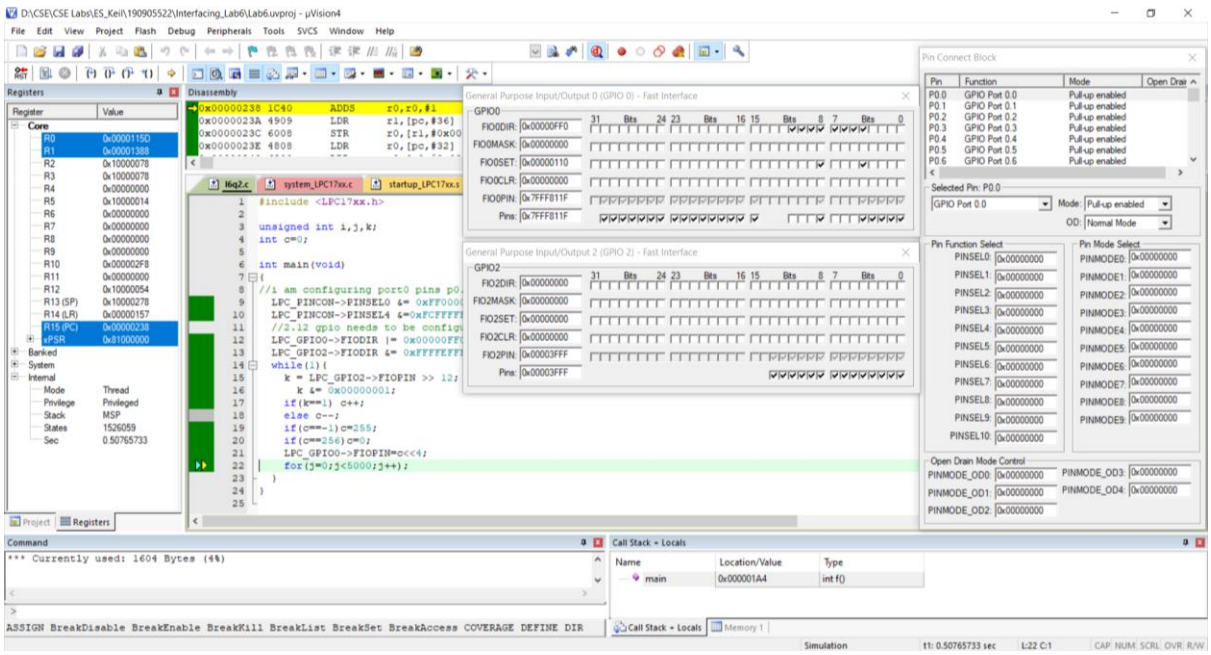
2) **Write a C program to read a key and display an 8-bit up/down counter on the LEDs. Hint: Use key SW2(if SW2=1, up counter else down counter), which is available at CNB1 pin 7. Connect CNB1 to any controller connector like CNB, CNC, etc. Configure the corresponding port pin as GPIO using corresponding PINSEL register and as input pin using corresponding FIODIR register.**

CODE:

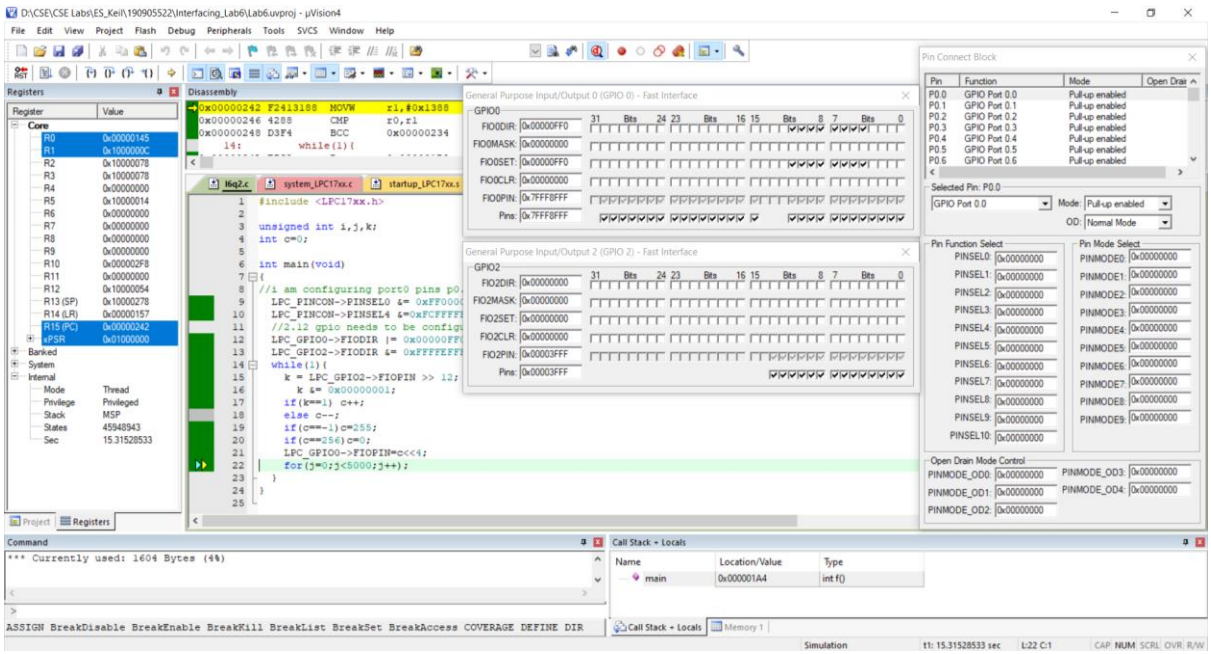
```
#include <LPC17xx.h>

unsigned int i,j,k;
int c=0;

int main(void)
{
//i am configuring port0 pins p0.4-p0.11 as gpio function
  LPC_PINCON->PINSEL0 &= 0xFF0000FF;
  LPC_PINCON->PINSEL4 &=0xFCFFFFFF;
  //2.12 gpio needs to be configured as input pin
  LPC_GPIO0->FIODIR |= 0x00000FF0;//configuring as output pins
  LPC_GPIO2->FIODIR &= 0xFFFFEFFF;//pin 12 is made to be 0 since i/p

    while(1){
        k = LPC_GPIO2->FIOPIN >> 12; //We read input from 2.12
        k &= 0x00000001;
        if(k==1)
            c++;
        else
            c--;
        if(c==-1)
            c=255;
        if(c==256)
            c=0;
        //left shifting by four because fourth bit is the starting bit
        LPC_GPIO0->FIOPIN=c<<4;
        //Putting a random delay here
        for(j=0;j<5000;j++);
    }
}
```

OUTPUT:

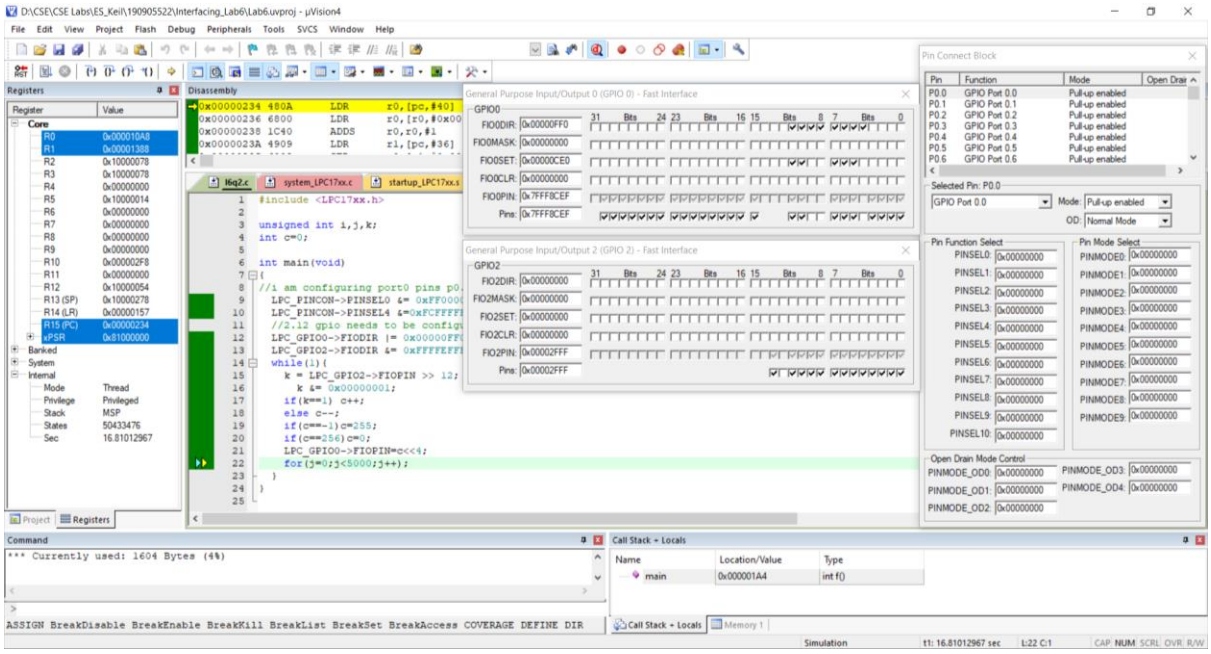
12th bit of the GPIO2 pin is turned on at this moment therefore we can see in GPIO0 that it acts as an up counter and the output is shown after 18 iterations therefore 10001.



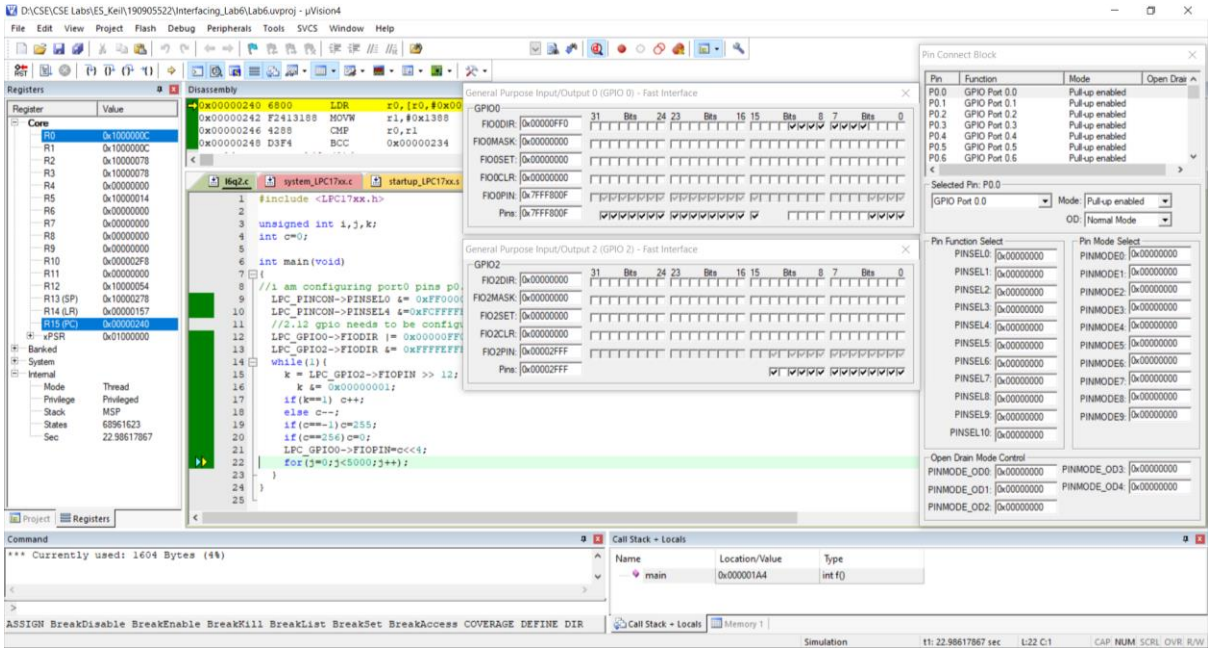
The switch remaining on, output is shown for after 256 iterations of the up counter therefore showing 1111 1111:



After the 12th bit of the GPIO2 register is unselected, the switch is turned off and the pins from 0.4 to 0.11 in GPIO0 act as a down counter and it shows the value after a few iterations of counting down from 1111 1111:



The switch bit remaining off we can see the down counter goes all the way to 0000 0000 after 256 iterations starting from 1111 1111. It will loop back as well.



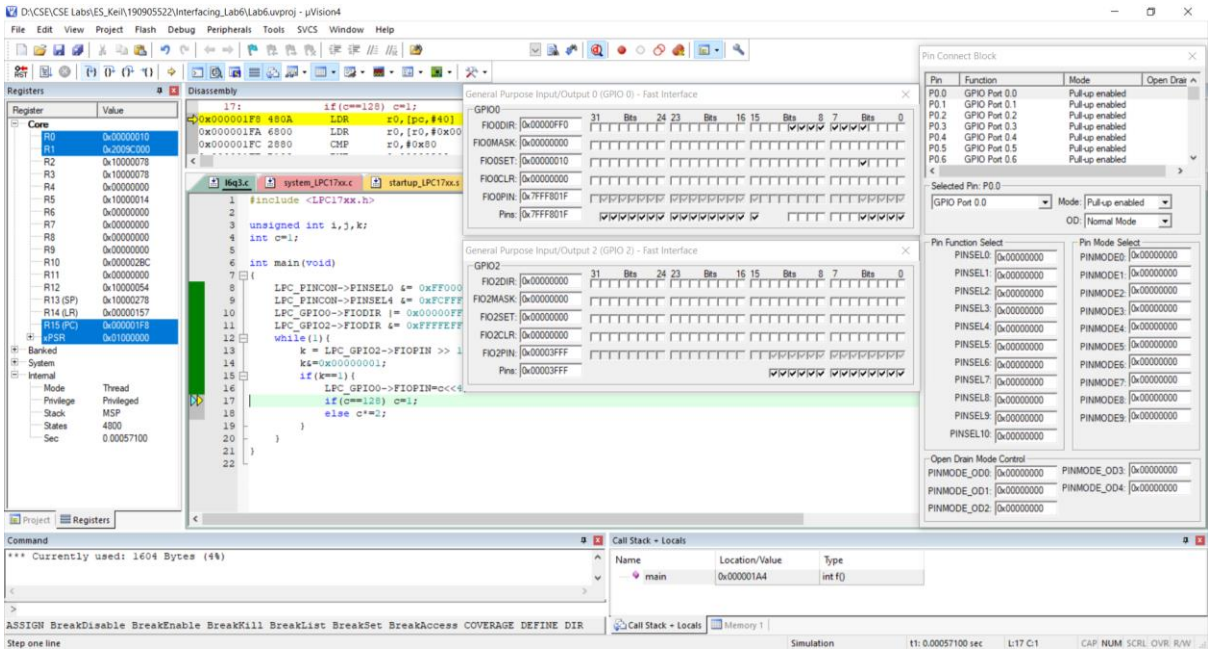
3) Write a program to simulate an 8-bit ring counter with key press (SW2).

CODE:

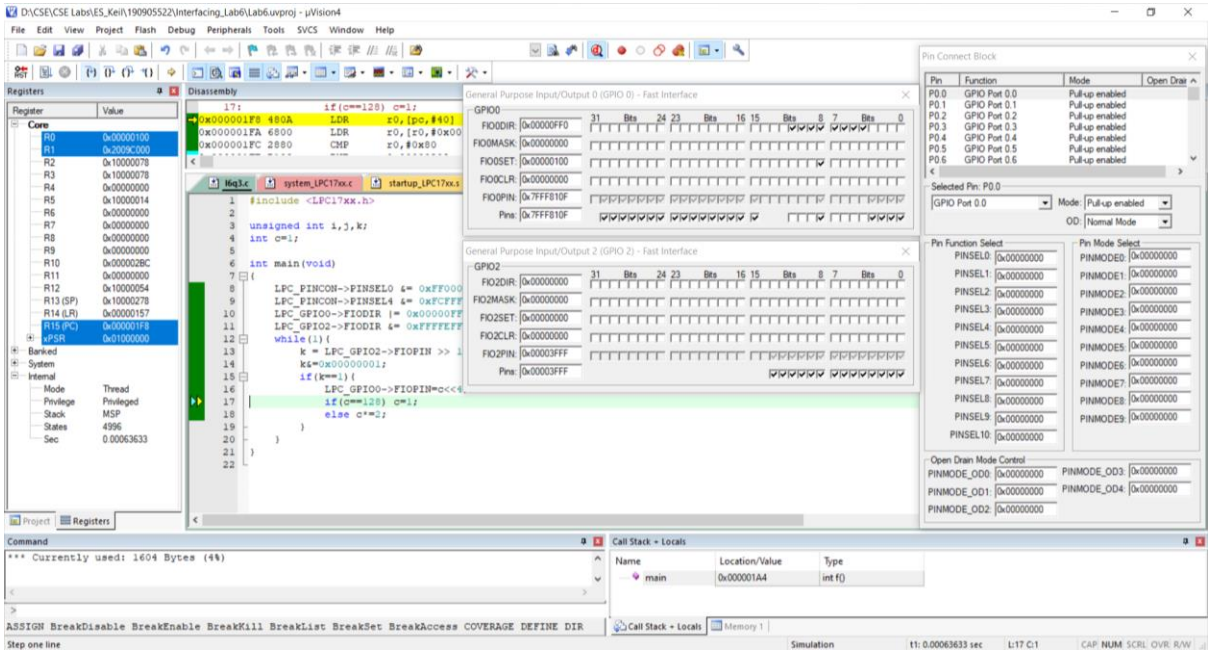
```
#include <LPC17xx.h>

unsigned int i,k;
int c=1;

int main(void)
{
    LPC_PINCON->PINSEL0 &= 0xFF0000FF; //pin 0.4-0.11 as gpio function
    LPC_PINCON->PINSEL4 &= 0xFCFFFFFF; //for pin 2.12 gpio2 as switch
    LPC_GPIO0->FIODIR |= 0x00000FF0; //config as output pin
    LPC_GPIO2->FIODIR &= 0xFFFFEFFF; //pin 12 is made to be 0 since i/p
    while(1){
        k = LPC_GPIO2->FIOPIN >> 12;
        k&=0x00000001; //reads switch press
        if(k==1){
            LPC_GPIO0->FIOPIN=c<<4; //sincce starts from bit 4 to bit 11
            if(c==128) c=1; //to reset the ring counter
            else c*=2;   //shifting by 1 bit
        }
        for(i=0;i<5000;i++);//random delay
    }
}
```

OUTPUT:

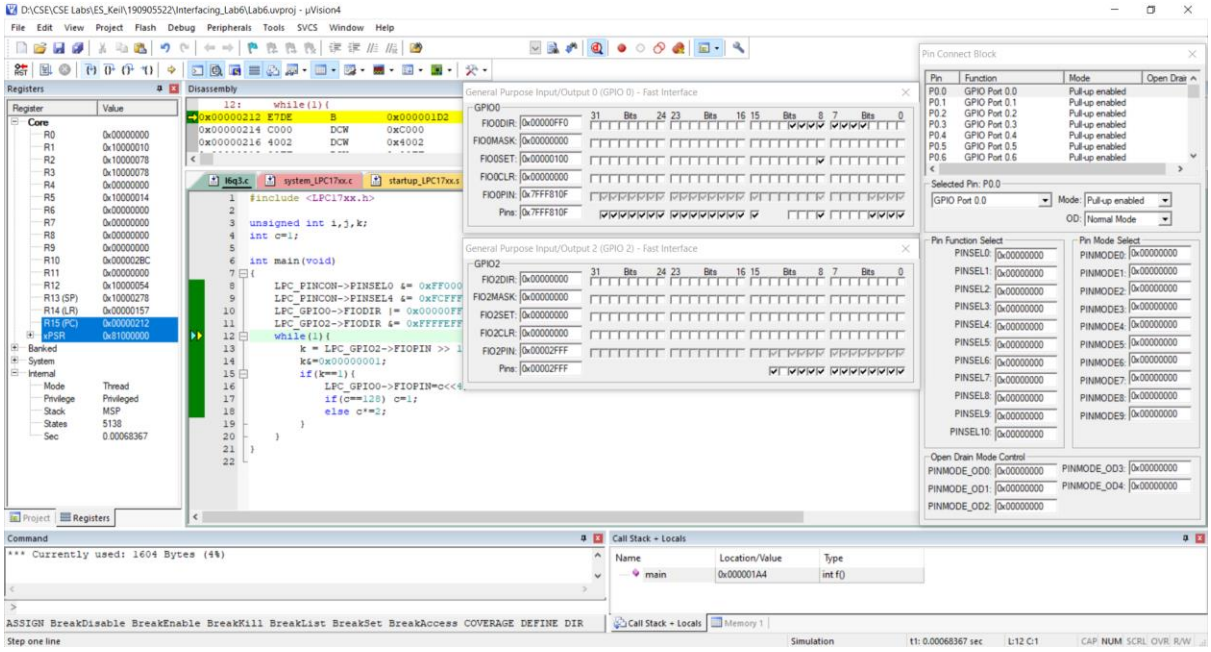
When the switch is pressed and the loop executes one time the output should be 0000 0001:



When the switch is pressed, and the loop is executed 5 times the output should be 0001 0000:



When the switch is not pressed, the 12th bit of the GPIO2 is unselected then the counter is stuck at the previous state 0001 0000 and will not start until the switch is pressed again:



The 8-bit ring counter goes till 1000 0000 and then back to 0000 0001 when the switch is pressed.

THE END