

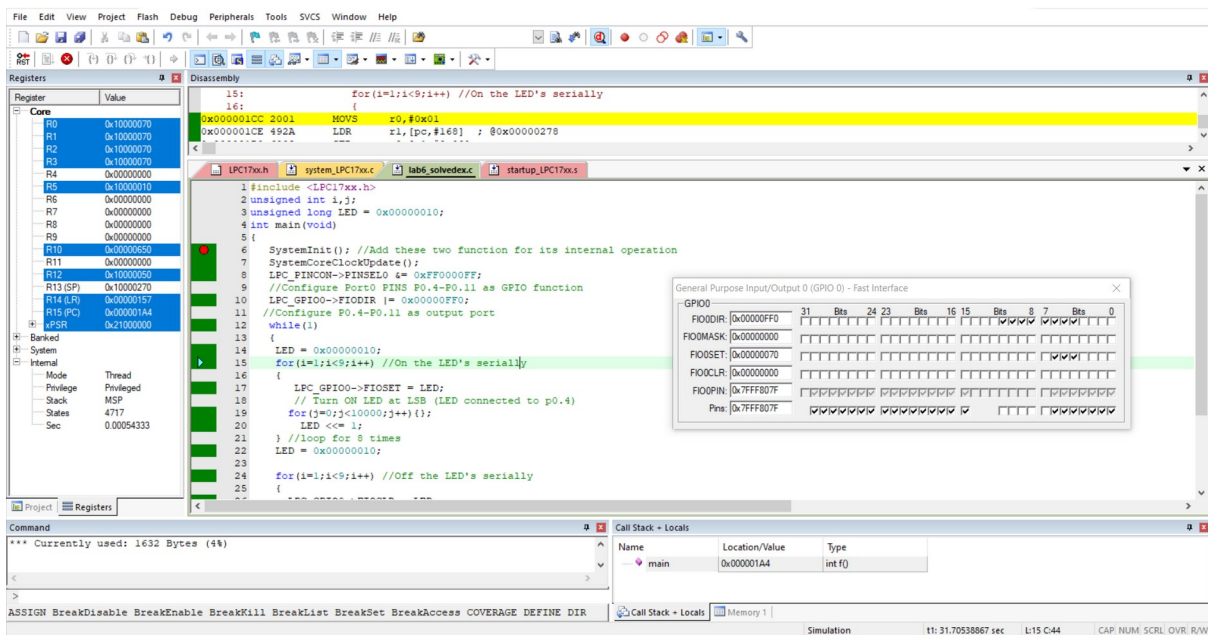
SOLVED EXCERSIZE :

```
/* Write a program to turn on/off the LEDs serially. */

#include <LPC17xx.h>
unsigned int i,j;
unsigned long LED = 0x00000010;
int main(void)
{
    SystemInit(); //Add these two function for its internal operation
    SystemCoreClockUpdate();
    LPC_PINCON->PINSEL0 &= 0xFF0000FF;
    //Configure Port0 PINS P0.4-P0.11 as GPIO function
    LPC_GPIO0->FIODIR |= 0x00000FF0;
    //Configure P0.4-P0.11 as output port
    while(1)
    {
        LED = 0x00000010;
        for(i=1;i<9;i++) //On the LED's serially
        {
            LPC_GPIO0->FIOSET = LED;
            // Turn ON LED at LSB (LED connected to p0.4)
            for(j=0;j<10000;j++){};
            LED <<= 1;
        } //loop for 8 times
        LED = 0x00000010;

        for(i=1;i<9;i++) //Off the LED's serially
        {
            LPC_GPIO0->FIOCLR = LED;
            //Turn OFF LED at LSB (LED connected to p0.4)
            for(j=0;j<10000;j++){};
            LED <<= 1;
        }
    }
}
```

OUTPUT :

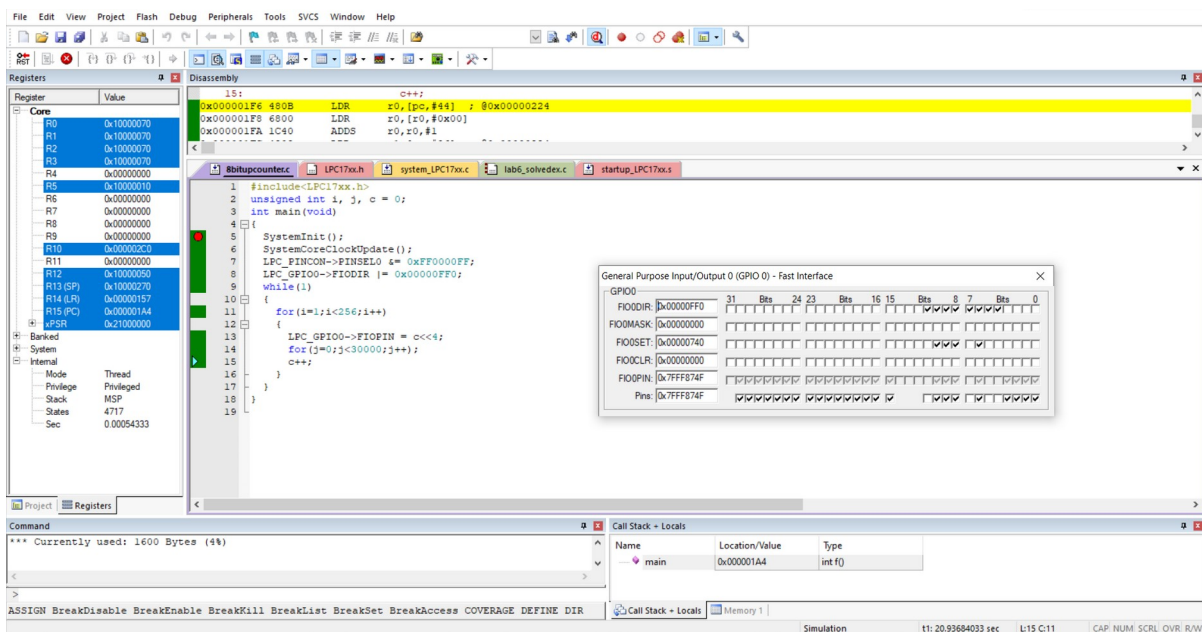


LAB EXCERSIZE :

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

```
#include<LPC17xx.h>
unsigned int i, j, c = 0;
int main(void)
{
    SystemInit();
    SystemCoreClockUpdate();
    LPC_PINCON->PINSEL0 &= 0xFF0000FF;
    LPC_GPIO0->FIODIR |= 0x000000FF0;
    while(1)
    {
        for(i=1;i<256;i++)
        {
            LPC_GPIO0->FIOPIN = c<<4;
            for(j=0;j<30000;j++);
            c++;
        }
    }
}
```

OUTPUT :

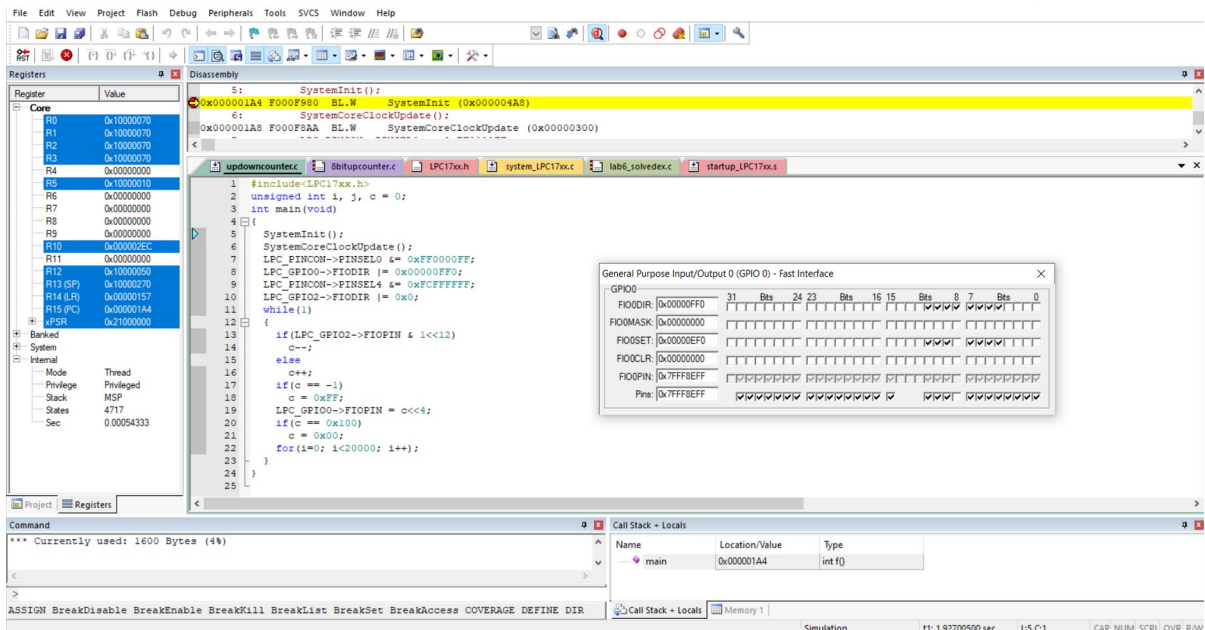


2. Write a C program to read a key and display an 8-bit up/down counter on the LEDs

```
#include<LPC17xx.h>
unsigned int i, j, c = 0;
int main(void)
{
    SystemInit();
    SystemCoreClockUpdate();
    LPC_PINCON->PINSEL0 &= 0xFF0000FF;
    LPC_GPIO0->FIODIR |= 0x00000FF0;
    LPC_PINCON->PINSEL4 &= 0xFCFFFFFF;
    LPC_GPIO2->FIODIR |= 0x0;
    while(1)
    {
        if(LPC_GPIO2->FIOPIN & 1<<12)
        c--;
        else
        c++;
        if(c == -1)
        c = 0xFF;
        LPC_GPIO0->FIOPIN = c<<4;
        if(c == 0x100)
        c = 0x00;
        for(i=0; i<20000; i++);
    }
}
```

```
}
}
```

OUTPUT :

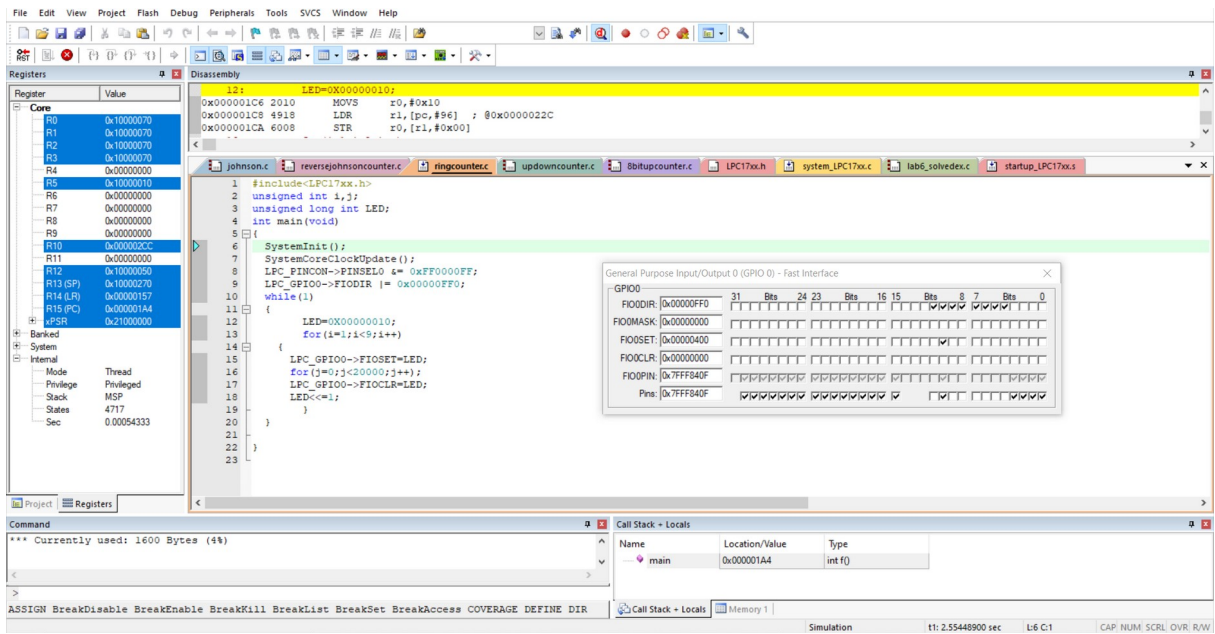


3. Write a program to simulate an 8- bit ring counter with key press (SW2).

```
#include<LPC17xx.h>
unsigned int i,j;
unsigned long int LED;
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<20000;j++);
            LPC_GPIO0->FIOCLR=LED;
            LED<=<=1;
        }
    }
}
```

}

OUTPUT :



The screenshot displays the Keil uVision IDE interface during a simulation. The main window shows the assembly view of a program, with the following instructions visible:

```
12: LED=0x00000010;
0x000001C6 2010 MOVs r0,#0x10
0x000001C8 4918 LDR r1,[pc,#96] ; @0x0000022C
0x000001CA 6008 STR r0,[r1,#0x00]
```

The registers window on the left shows the current state of the processor registers, with the following values:

Register	Value
R0	0x10000070
R1	0x10000070
R2	0x10000070
R3	0x10000070
R4	0x20000000
R5	0x10000010
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x000002C0
R11	0x00000000
R12	0x10000050
R13 (SP)	0x10000070
R14 (LR)	0x00000157
R15 (PC)	0x000001A4
MPSR	0x21000000

The command window at the bottom shows the current memory usage: "*** Currently used: 1600 Bytes (4%)". The call stack window shows the current function call stack, with the main function at the top.

The GPIO pin configuration window is also visible, showing the pin configuration for the LED. The pin is configured as an output pin, and the pin number is 7.