

Lab Exercises:

3. Write a C program to simulate 4-digit BCD up/down counter on the multiplexed seven-segment display.

Program:

```
#include<LPC17XX.h>
#define FIRSTSEG 0<<23;
#define SECONDSEG 1<<23;
#define THIRDSEG 2<<23;
#define FOURTHSEG 3<<23;
unsigned int dig_1 = 0x00, dig_2 = 0x00, dig_3 = 0x00, dig_4 = 0x00;
unsigned long i;
unsigned int dig_count = 0, temp1 = 0x00, one_sec_flag = 0x00;
unsigned int array_dec[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};
unsigned long int temp2 = 0x0;
void display(void)
{
    if(dig_count == 0x01)
    {
        temp1 = dig_1;
        LPC_GPIO1->FIOPIN = FIRSTSEG;
    }
    if(dig_count == 0x02)
    {
        temp1 = dig_2;
        LPC_GPIO1->FIOPIN = SECONDSEG;
    }
    if(dig_count == 0x03)
    {
        temp1 = dig_3;
        LPC_GPIO1->FIOPIN = THIRDSEG;
```

```

}
if(dig_count == 0x04)
{
temp1 = dig_4;
LPC_GPIO1->FIOPIN = FOURTHSEG;
}
temp1 &= 0x0F;
temp2 = array_dec[temp1];
temp2<<=4;
LPC_GPIO0->FIOPIN = temp2;
for(i=0; i<50000; i++);
LPC_GPIO0->FIOCLR = 0x00000FF0;
}
void delay()
{
for(i=0; i<10000; i++);
}
int main()
{
LPC_GPIO0->FIODIR |=0xFF<<4;
LPC_GPIO1->FIODIR |=15<<23;
while(1)
{
delay();
dig_count +=1;
if(dig_count == 0x05)
{
delay();
dig_count = 0x01;
one_sec_flag = 0xFF;
}
if((LPC_GPIO2->FIOPIN & 1))
{

```

```

if(one_sec_flag == 0xFF)
{
one_sec_flag = 0x00;
dig_1+=1;
if(dig_1 == 0xA)
{
dig_1 = 0;
dig_2+=1;
if(dig_2 == 0xA)
{
dig_2=0;
dig_3+=1;
if(dig_3 == 0xA)
{
dig_3 = 0;
dig_4+=1;
if(dig_4==0xA)
{
dig_4 = 0;
}}}}}}
else if(!(LPC_GPIO2->FIOPIN & 1))
{
if(one_sec_flag==0xFF)
{
one_sec_flag = 0x00;
dig_1-=1;
if(dig_1==0xFFFFFFFF)
{
dig_1=9;
dig_2-=1;
if(dig_2==0xFFFFFFFF)
{
dig_2=9;

```

```

dig_3--;
if(dig_3==0xFFFFFFFF)
{
dig_3 = 9;
dig_4 -= 1;
if(dig_4==0xFFFFFFFF)
{
dig_4 = 9;
}}}}}}
display();}}

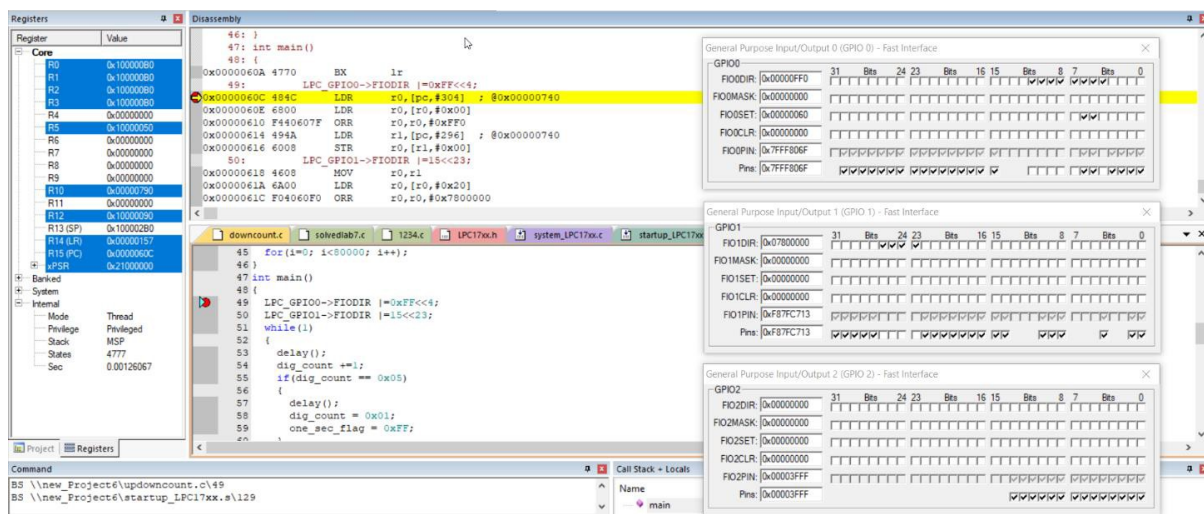
```

Output:

Counting up from 1000 to 2000

GPIO Port 0 : (00000110)2, 0x06 = Displaying 1, GPIO Port 1: (00)2,

GPIO Port 2 : P2.0 SET



GPIO Port 0 : (01011011)2, 0x5B = Displaying 2, GPIO Port 1: (00)2 GPIO Port 2 : P2.0 SET

Counting down from 9998 to 9997

GPIO Port 0 : (01111111)2, 0x7F = Displaying 8, GPIO Port 1:
(00)2 GPIO Port 2 : P2.0 CLEAR

GPIO Port 0: (00000111)2, 0x7D = Displaying 7, GPIO Port 1:
(00)2 GPIO Port 2: P2.0 CLEAR

4. Write a C program for 4-digit hexadecimal up/down counter on seven segment using a switch and timer with a delay of 1 second between each count.

Program:

```
#include<LPC17XX.h>
#define FIRSTSEG 0<<23;
#define SECONDSEG 1<<23;
#define THIRDSEG 2<<23;
#define FOURTHSEG 3<<23;
unsigned int dig_1 = 0x00, dig_2 = 0x00, dig_3 = 0x00, dig_4 = 0x00;
unsigned int i, dig_count = 0, temp1 = 0x00, one_sec_flag = 0x00;
unsigned int array_dec[16] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F, 0x77, 0x7C, 0x39, 0x5E, 0x79, 0x71};
unsigned long int temp2 = 0x0;
void display(void){
    if(dig_count == 0x01){
        temp1 = dig_1;
        LPC_GPIO1->FIOPIN = FIRSTSEG;
    }
    if(dig_count == 0x02){
        temp1 = dig_2;
        LPC_GPIO1->FIOPIN = SECONDSEG;
    }
    if(dig_count == 0x03){ temp1 = dig_3;
        LPC_GPIO1->FIOPIN = THIRDSEG;
    }
    if(dig_count == 0x04){
        temp1 = dig_4;
        LPC_GPIO1->FIOPIN = FOURTHSEG;
    }
    temp1 &= 0x0F;
    temp2 = array_dec[temp1];
    temp2<<=4;
    LPC_GPIO0->FIOPIN = temp2;
    for(i=0; i<50000; i++);
    LPC_GPIO0->FIOCLR = 0x00000FF0;
}
void delay(){
    for(i=0; i<50000; i++);
}
int main(){
    LPC_GPIO0->FIODIR |=0xFF<<4;
    LPC_GPIO1->FIODIR |=15<<23;
    while(1)
    {
        delay();
        dig_count +=1;
        if(dig_count==0x05){
            delay();
            dig_count = 0x01;
            one_sec_flag = 0xFF;
        }
        if((LPC_GPIO2->FIOPIN & 1))
        {
```

```

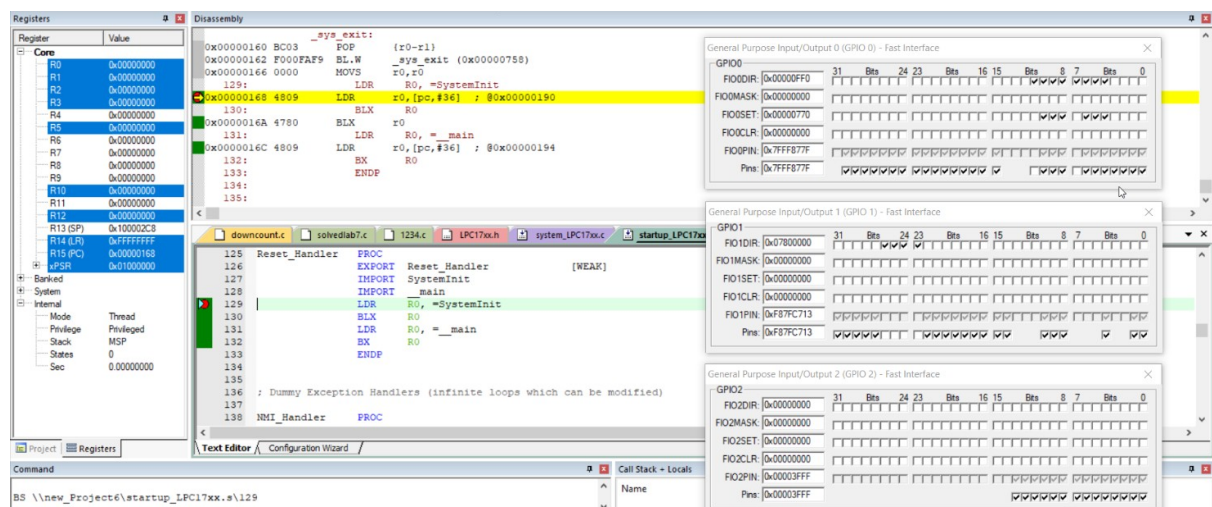
if(one_sec_flag==0xFF){
one_sec_flag = 0x00;
dig_1+=1;
if(dig_1>0xF){
dig_1 = 0;
dig_2+=1;
if(dig_2>0xF){
dig_2=0;
dig_3+=1;
if(dig_3>0xF){
dig_3 = 0;
dig_4+=1;
if(dig_4>0xF){
dig_4 = 0;}}}}}}
else if(!(LPC_GPIO2->FIOPIN & 1))
{
if(one_sec_flag==0xFF){
one_sec_flag = 0x00;
dig_1-=1;
if(dig_1==0xFFFFFFFF){
dig_1=0xF;
dig_2-=1;
if(dig_2==0xFFFFFFFF){
dig_2=0xF;
dig_3-=1;
if(dig_3==0xFFFFFFFF){
dig_3 =0xF;
dig_4-=1;
if(dig_4==0xFFFFFFFF){
dig_4 = 0xF;}}}}}}
display();}}

```

Output:

Counting up from 10 to 11 (A to B)

GPIO Port 0: (01110111)2 = 0x77 = 10, GPIO Port 1: (00)2, GPIO Port 2: P2.0 SET 0x77 to 0x7C



GPIO Port 0 : (01111100)2 = 0x7C = Displaying 11, GPIO Port 1: (00)2, GPIO Port 2 : P2.0 SET

The screenshot shows the Keil uVision IDE with the assembly view of the startup code. The registers window on the left shows the core registers. The disassembly window in the center shows the following code:

```

129: 0x00000160 BC03 POP (r0-r1)
130: 0x00000162 F000FADF BL.W _sys_exit (0x00000758)
131: 0x00000166 0000 MOV.S r0,r0
132: 0x00000168 4809 LDR r0,[pc,#36] ; @0x00000190
133: 0x0000016A 4780 BLX r0
134: 0x0000016C 4809 LDR r0,[pc,#36] ; @0x00000194
135: 0x0000016E 0000 BX r0
136: 0x00000170 ENDP

```

On the right, three GPIO configuration windows are open:

- GPIO0 - Fast Interface:** FIODIR=0x0000FFD0, FIO1MASK=0x00000000, FIO1SET=0x000007C0, FIO1CLR=0x00000000, FIO1PIN=0x7FFF87C0. Pins: 0-7 are checked.
- GPIO1 - Fast Interface:** FIODIR=0x78000000, FIO1MASK=0x00000000, FIO1SET=0x00000000, FIO1CLR=0x00000000, FIO1PIN=0xF87FC713. Pins: 0-7 are checked.
- GPIO2 - Fast Interface:** FIODIR=0x00000000, FIO2MASK=0x00000000, FIO2SET=0x00000000, FIO2CLR=0x00000000, FIO2PIN=0x00003FFF. Pins: 0-7 are checked.

Counting down from E to D (14 to 13)

GPIO Port 0: (01111001)₂ = 0x79 = Displaying 14, GPIO Port 1: (00)₂ GPIO Port 2: P2.0 CLEAR

The screenshot shows the Keil uVision IDE with the assembly view of the startup code. The registers window on the left shows the core registers. The disassembly window in the center shows the following code:

```

129: 0x00000160 BC03 POP (r0-r1)
130: 0x00000162 F000FADF BL.W _sys_exit (0x00000760)
131: 0x00000166 0000 MOV.S r0,r0
132: 0x00000168 4809 LDR r0,[pc,#36] ; @0x00000190
133: 0x0000016A 4780 BLX r0
134: 0x0000016C 4809 LDR r0,[pc,#36] ; @0x00000194
135: 0x0000016E 0000 BX r0
136: 0x00000170 ENDP

```

On the right, three GPIO configuration windows are open:

- GPIO0 - Fast Interface:** FIODIR=0x0000FFD0, FIO1MASK=0x00000000, FIO1SET=0x00000790, FIO1CLR=0x00000000, FIO1PIN=0x7FFF879F. Pins: 0-7 are checked.
- GPIO1 - Fast Interface:** FIODIR=0x78000000, FIO1MASK=0x00000000, FIO1SET=0x00000000, FIO1CLR=0x00000000, FIO1PIN=0xF87FC713. Pins: 0-7 are checked.
- GPIO2 - Fast Interface:** FIODIR=0x00000000, FIO2MASK=0x00000000, FIO2SET=0x00000000, FIO2CLR=0x00000000, FIO2PIN=0x00003FFE. Pins: 0-7 are checked.

GPIO Port 0: (01011110)₂ = 0x5E = Displaying 13, GPIO Port 1: (00)₂, GPIO Port 2: P2.0 CLEAR

The screenshot shows the Keil uVision IDE with the assembly view of the startup code. The registers window on the left shows the core registers. The disassembly window in the center shows the following code:

```

129: 0x00000160 BC03 POP (r0-r1)
130: 0x00000162 F000FADF BL.W _sys_exit (0x00000760)
131: 0x00000166 0000 MOV.S r0,r0
132: 0x00000168 4809 LDR r0,[pc,#36] ; @0x00000190
133: 0x0000016A 4780 BLX r0
134: 0x0000016C 4809 LDR r0,[pc,#36] ; @0x00000194
135: 0x0000016E 0000 BX r0
136: 0x00000170 ENDP

```

On the right, three GPIO configuration windows are open:

- GPIO0 - Fast Interface:** FIODIR=0x0000FFD0, FIO1MASK=0x00000000, FIO1SET=0x000005E0, FIO1CLR=0x00000000, FIO1PIN=0x7FFF85EF. Pins: 0-7 are checked.
- GPIO1 - Fast Interface:** FIODIR=0x78000000, FIO1MASK=0x00000000, FIO1SET=0x00000000, FIO1CLR=0x00000000, FIO1PIN=0xF87FC713. Pins: 0-7 are checked.
- GPIO2 - Fast Interface:** FIODIR=0x00000000, FIO2MASK=0x00000000, FIO2SET=0x00000000, FIO2CLR=0x00000000, FIO2PIN=0x00003FFE. Pins: 0-7 are checked.