EMBEDDED PROGRAMMING LAB

LAB-6   DATE:23-10-2024

PREETHISH K R

Steps followed for programming in kiel using C programming

- **Creating a project file**
  -Open project
  -new uvision project
  - project name then save
- **Startup for c**
  -Source group
  -add existing file to group 'source group 1'
  -Disk C
  - Keil
  -ARM
  -Startup
  - NXP
  -LPC17xx
  -systemLPC17xx
  - Add
- **Flashing the program to hardware**
- **Target 1**
  -option for target
- Target
  -use micro lib
  -IROM1-0X0…….0X80000
  -IRAM2-0X10000000………0X8000
- Output
  -create hex file
- Listing
  -C preprocessor listing
- Linker
  -use memory layout from target dialog

- **Flash magic**
  -Select devise LPC1768
  -Baud rate-115200
  -interface None(ISP)
  -oscillation 12
- Erase blocks used by hexfile

1.Write a code to toggle the LED with some delay
Program:

```
#include<lpc17xx.h>
int main()
{
unsigned int delay;
LPC_PINCON->PINSEL5=0X00000000;
LPC_GPIO2->FIODIR=0X00FF0000;
while(1)
 {
LPC_GPIO2->FIOSET=0X00FF0000;
for(delay=0;delay<50;delay++);
LPC_GPIO2->FIOCLR=0X00FF0000;
 for(delay=0;delay<50;delay++);
 }
}
```
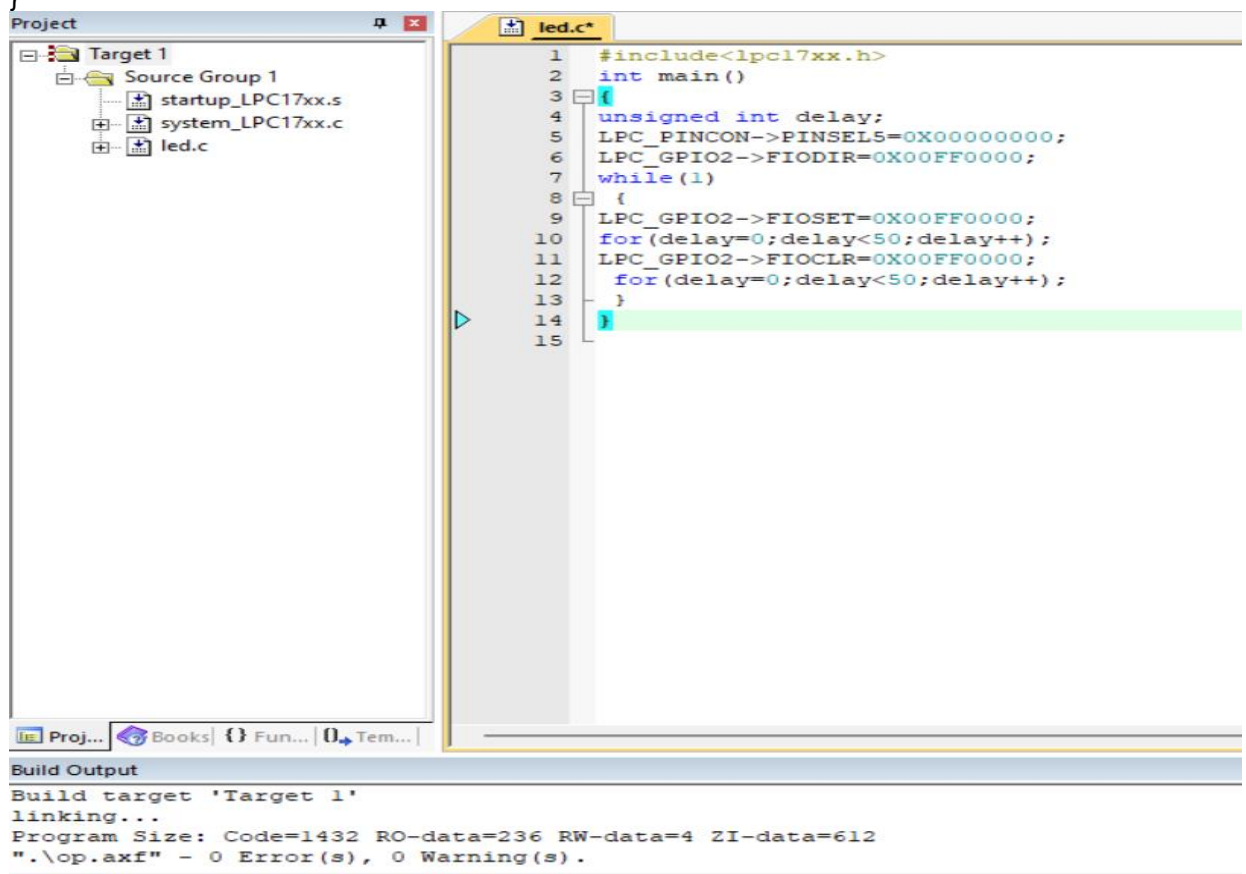

Fig-1 code to toggle the LED with some delay

General Purpose Input/Output 2 (GPIO 2) – Fast Interface

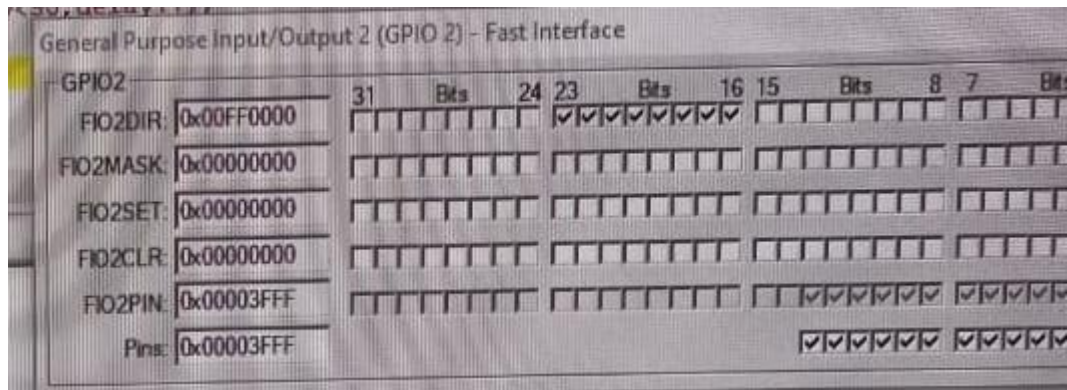| GPIO2		31	Bits	24	23	Bits	16	15	Bits	8	7	Bit
FIO2DIR:	0x00FF0000				☑☑☑☑☑☑☑☑							
FIO2MASK:	0x00000000											
FIO2SET:	0x00000000											
FIO2CLR:	0x00000000											
FIO2PIN:	0x00003FFF							☑☑☑☑☑	☑☑☑☑☑			
Pins:	0x00003FFF							☑☑☑☑☑☑	☑☑☑☑☑			

Fig1.2-output obtained

2. write a program to toggle the alternate LED

Program:

```c
#include<lpc17xx.h>
int main()
{
unsigned int delay;
LPC_PINCON->PINSEL5=0X00000000;
LPC_GPIO2->FIODIR=0X00FF0000;
while(1)
 {
LPC_GPIO2->FIOSET=0X00550000;
for(delay=0;delay<50;delay++);
LPC_GPIO2->FIOCLR=0X00550000;
 for(delay=0;delay<50;delay++);

LPC_GPIO2->FIOSET=0X00AA0000;
for(delay=0;delay<50;delay++);
LPC_GPIO2->FIOCLR=0X00AA0000;
 for(delay=0;delay<50;delay++);
 }

}
```
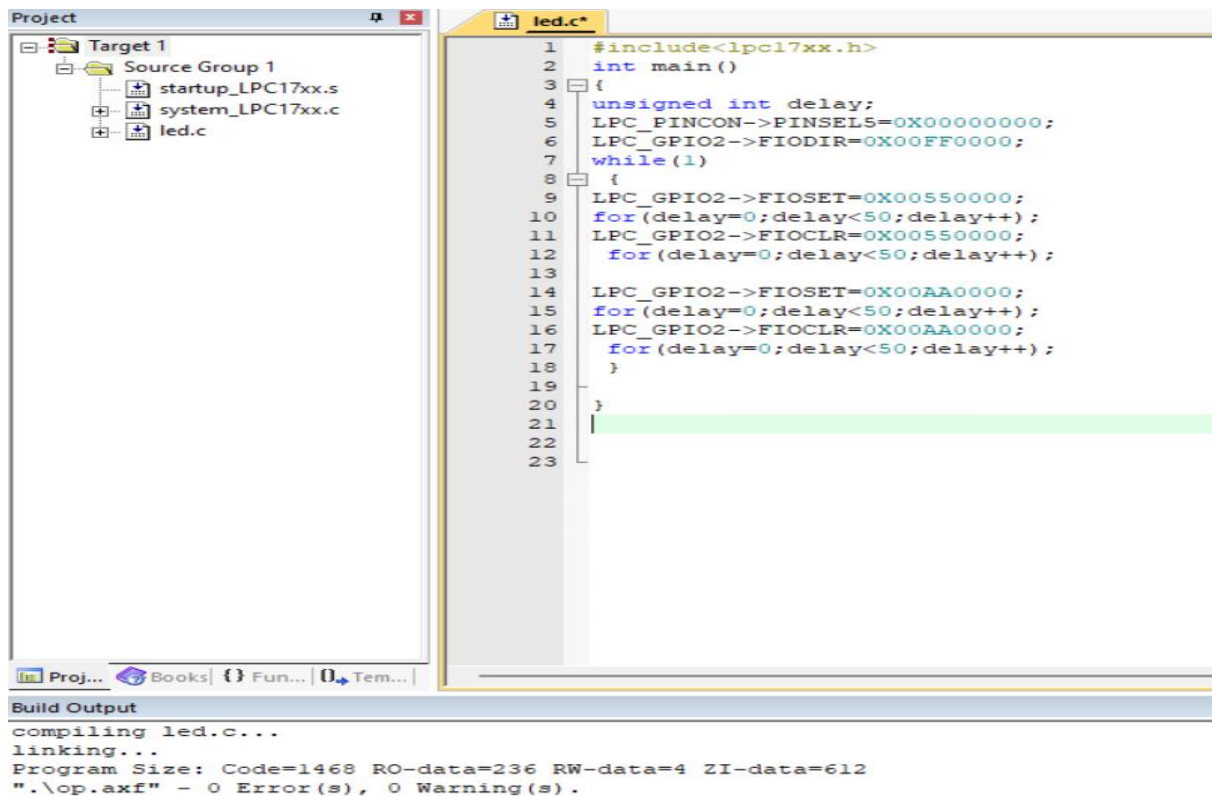
Fig 2-program to toggle the alternate LED

3. Write a program to turn ON and the LED from left to right if switch S1 is ON and right to left if switch S2 is ON (S1-P2.25, S2-P2.26)

Program:

```c
#include<LPC17XX.H>

int main(void){
    unsigned int i,delay;
    LPC_PINCON->PINSEL5=0X00000000;
    LPC_GPIO2->FIODIR=0X000FF000;
    while(1){
        if(!(LPC_GPIO2->FIOPIN & 0X02000000))
        {
            for(i=23;i>=16;i--){
                LPC_GPIO2->FIOSET=(1<<i);
                for (delay=0;delay<50;delay++);
            }
        }
        if(!(LPC_GPIO2->FIOPIN & 0X04000000))
        {
            for(i=16;i>=23;i++){
                LPC_GPIO2->FIOSET=(1<<i);
                for (delay=0;delay<50;delay++);
            }
        }
    }
```

}


```
led.c
1    #include<LPC17XX.H>
2
3    int main(void){
4        unsigned int i,delay;
5        LPC_PINCON->PINSEL5=0X00000000;
6        LPC_GPIO2->FIODIR=0X000FF000;
7        while(1){
8            if(!(LPC_GPIO2->FIOPIN & 0X02000000))
9            {
10               for(i=23;i>=16;i--){
11                   LPC_GPIO2->FIOSET=(1<<i);
12                   for (delay=0;delay<50;delay++);
13               }
14           }
15           if(!(LPC_GPIO2->FIOPIN & 0X04000000))
16           {
17               for(i=16;i>=23;i++){
18                   LPC_GPIO2->FIOSET=(1<<i);
19                   for (delay=0;delay<50;delay++);
20               }
21           }
22       }
23   }
24
25
26
27
28
29
```

```
Build Output
compiling led.c...
linking...
Program Size: Code=1468 RO-data=236 RW-data=4 ZI-data=612
".\op.axf" - 0 Error(s), 0 Warning(s).
```
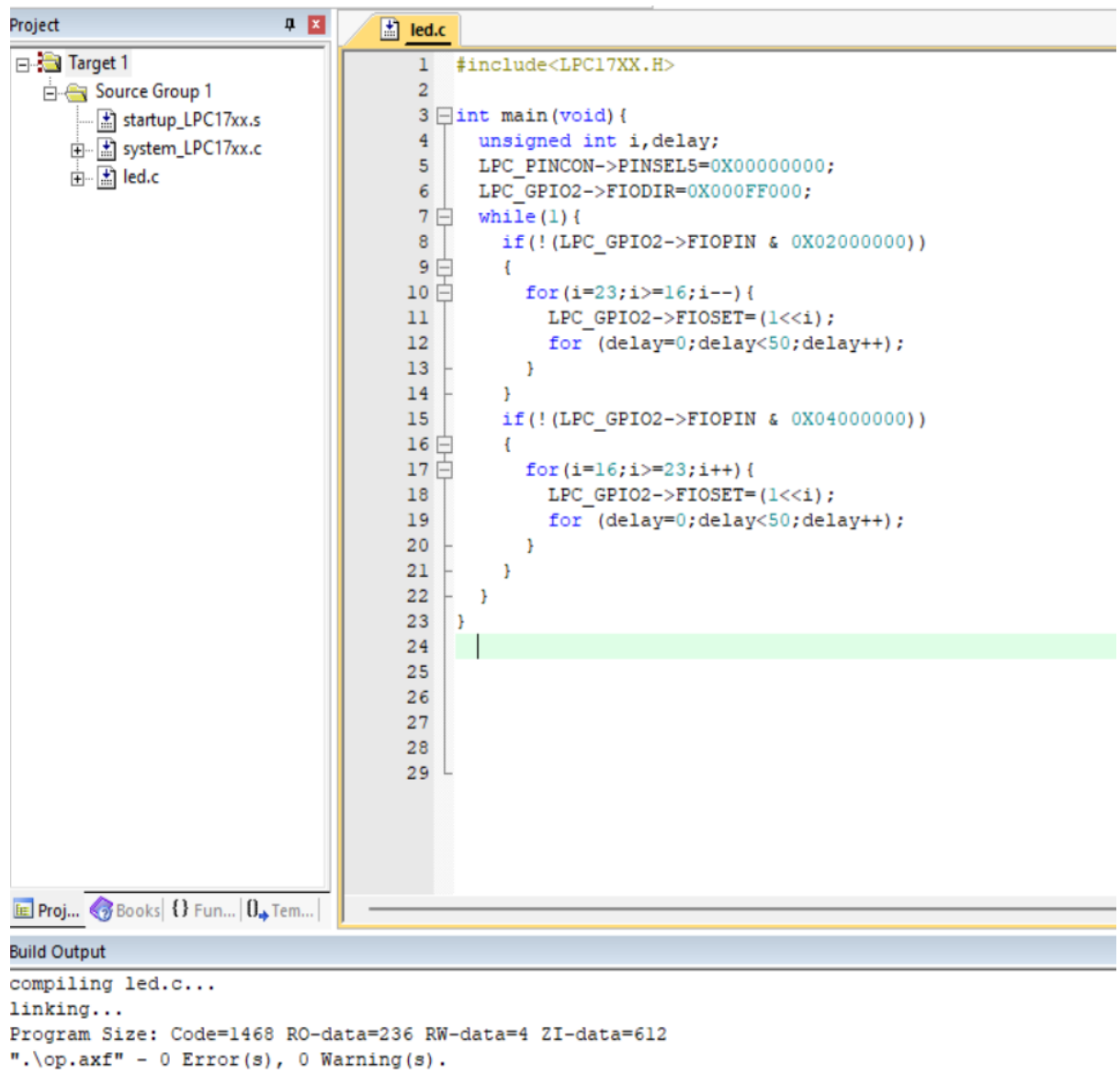
Fig 3- turn ON and the LED from left to right if switch S1 is ON and right to left if switch S2 is ON

4. Write a program to ON and OFF the buzzer when EXT1(Switch 12) P2.11 is ON

Program:

```
#include<LPC17XX.H>
int main(void)
{
LPC_PINCON->PINSEL1=0X00000000;
LPC_GPIO0->FIODIR=0X03000000;
while(1)
{
if(!(LPC_GPIO2->FIOPIN & 0X00000800))
{
LPC_GPIO0->FIOSET=0X03000000;
```
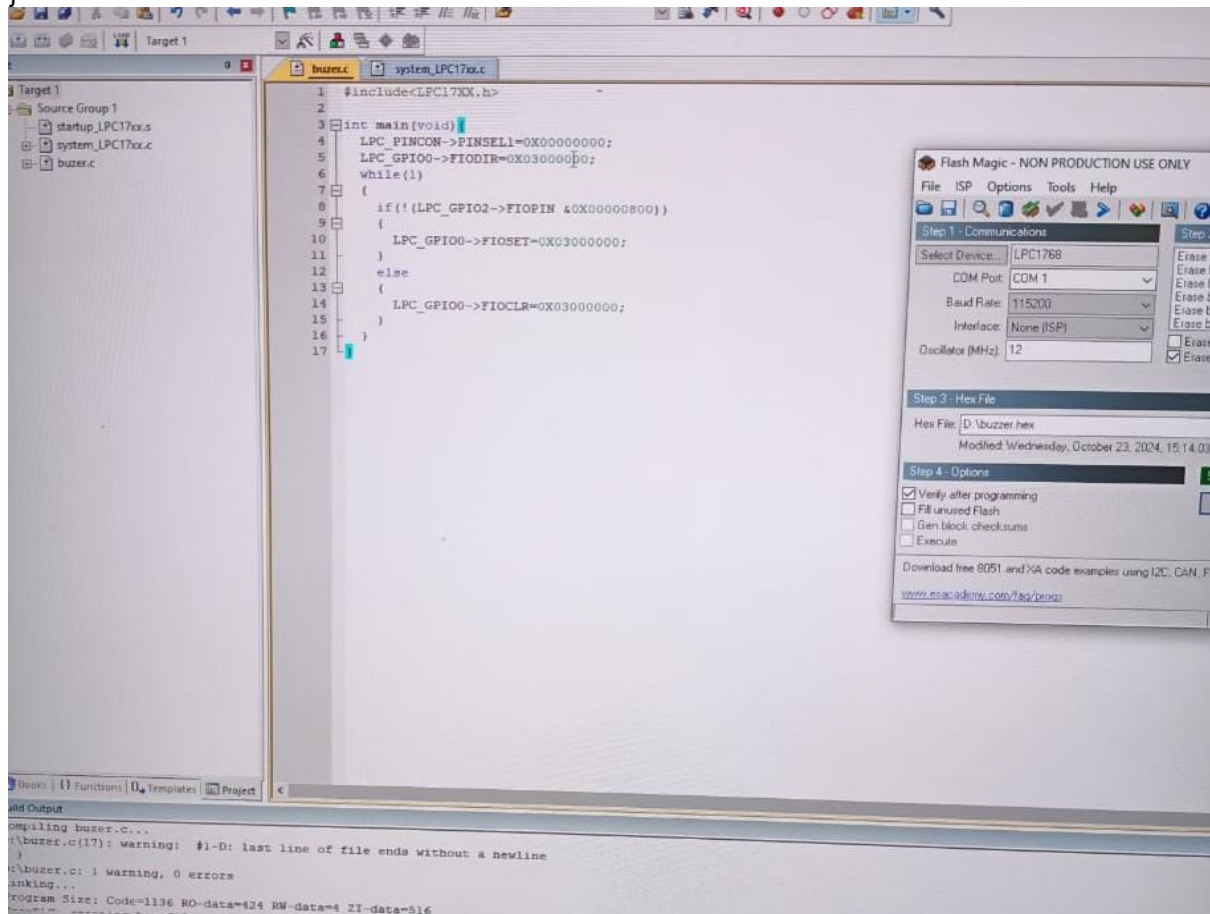
```
}
else
{
LPC_GPIO0->FIOCLR=0X03000000;
}
}
}
```


Fig 4- program to ON and OFF the buzzer