EE2016: Microprocessor Lab 7

Introduction to the LPC2378 Microcontroller

B#44

ANANYA DAS EE21B016
SAILENDRA EE21B018

CONTENT

PART 1

PART 2

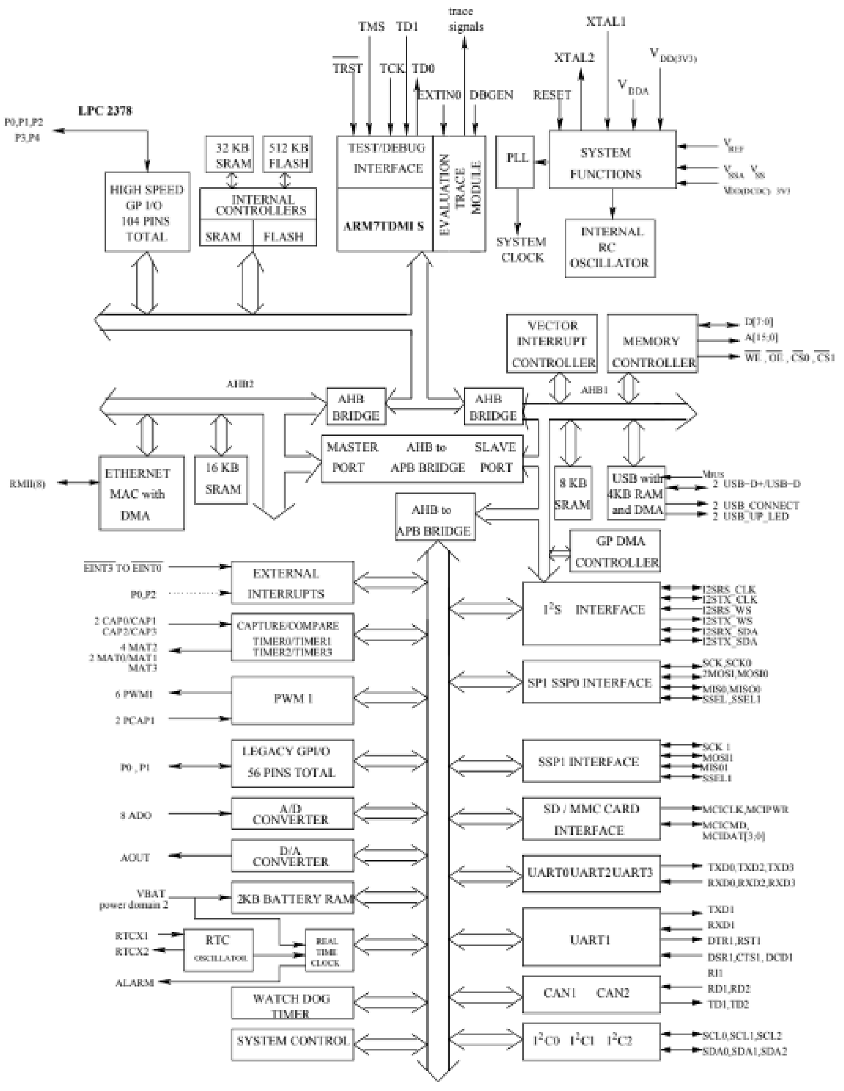
AIM:

Few experiments in this lab will use the LPC2378 microcontroller. The LPC2378 microcontroller is based on a 16-bit/32-bit ARM7TDMI-S CPU.

The LPC2378 Microcontroller:

The LPC2378 microcontroller to be used in this experiment is from NXP semiconductors. LPC stands for Low Power Consumption. The LPC2378 is based on the ARM7TDMI-S CPU so we will discuss aspects of ARM below.

Block Diagram of LPC2378



Tasks for the Experiment:

Task 1:

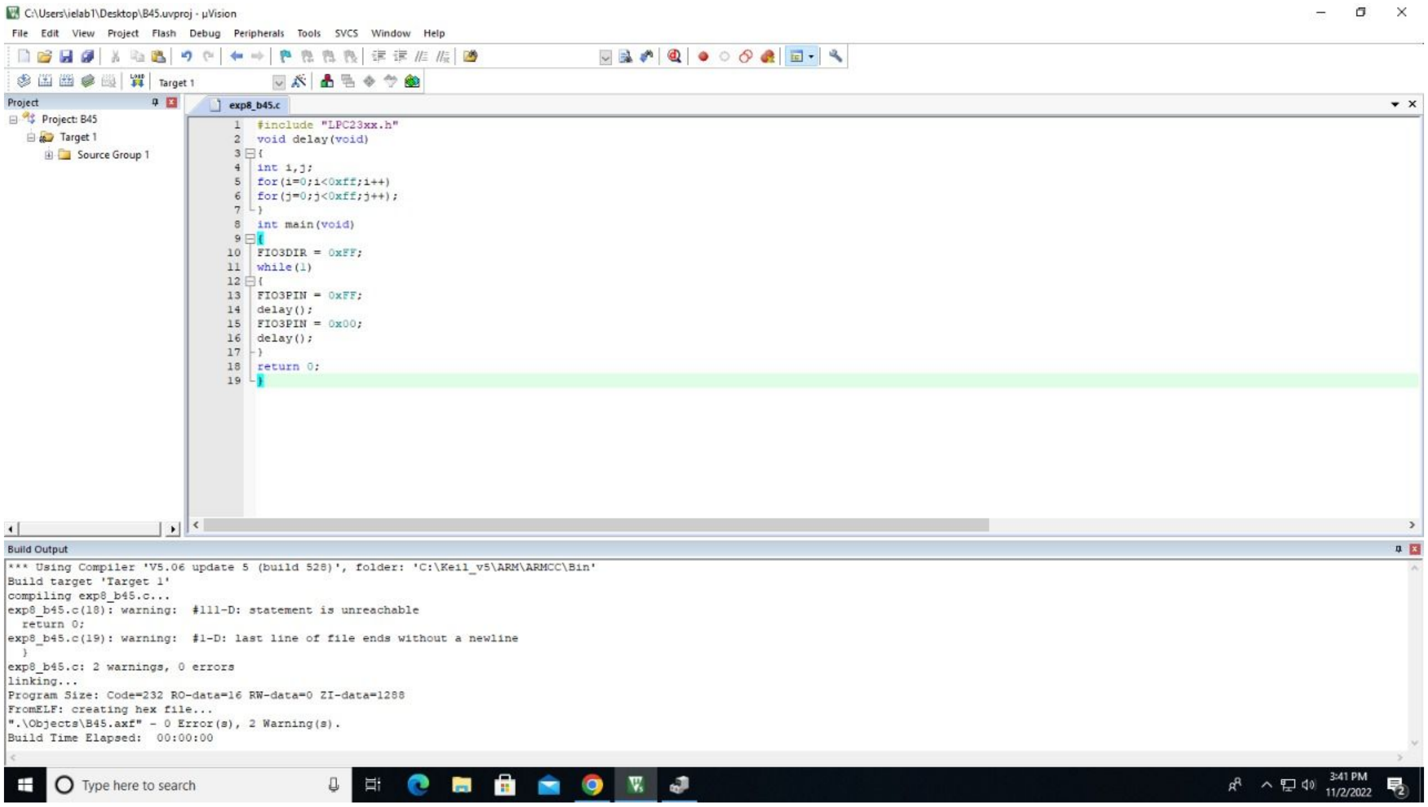
Complete the following program to cause the LEDs on the ARM-board to blink.

Code:

```
#include "LPC23xx.h"
void delay(void)
{
     int i,j;
     for(i=0;i<0xff;i++)
     {
          for(j=0;j<0xff;j++)
          {
          }
     }
}
int main(void)
{
     FIO3DIR = 0xFF;
     while(1)
     {
          FIO3PIN = 0xFF;
          delay();
          FIO3PIN = 0x00;
          delay();
     }

return 0;
}
```

The LEDs turn on and off after each subsequent delay and due to for loop they keep on blinking

Code Execution:


C:\Users\ielab1\Desktop\B45.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: B45
Target 1
Source Group 1
exp8_b45.c
Build Output
Type here to search
3:41 PM
11/2/2022

```
 1  #include "LPC23xx.h"
 2  void delay(void)
 3  {
 4  int i,j;
 5  for(i=0;i<0xff;i++)
 6  for(j=0;j<0xff;j++);
 7  }
 8  int main(void)
 9  {
10  FIO3DIR = 0xFF;
11  while(1)
12  {
13  FIO3PIN = 0xFF;
14  delay();
15  FIO3PIN = 0x00;
16  delay();
17  }
18  return 0;
19  }
```

```
*** Using Compiler 'V5.06 update 5 (build 528)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
compiling exp8_b45.c...
exp8_b45.c(18): warning:  #111-D: statement is unreachable
  return 0;
exp8_b45.c(19): warning:  #1-D: last line of file ends without a newline
  }
exp8_b45.c: 2 warnings, 0 errors
linking...
Program Size: Code=232 RO-data=16 RW-data=0 ZI-data=1288
FromELF: creating hex file...
".\Objects\B45.axf" - 0 Error(s), 2 Warning(s).
Build Time Elapsed:  00:00:00
```

Output:

~~https://drive.google.com/drive/folders/1_lkvJMq7fv5IkCSHldO1wmExooVFAjka~~

Inference:

From the video, we can see the blinking of LEDs.

Task 2:

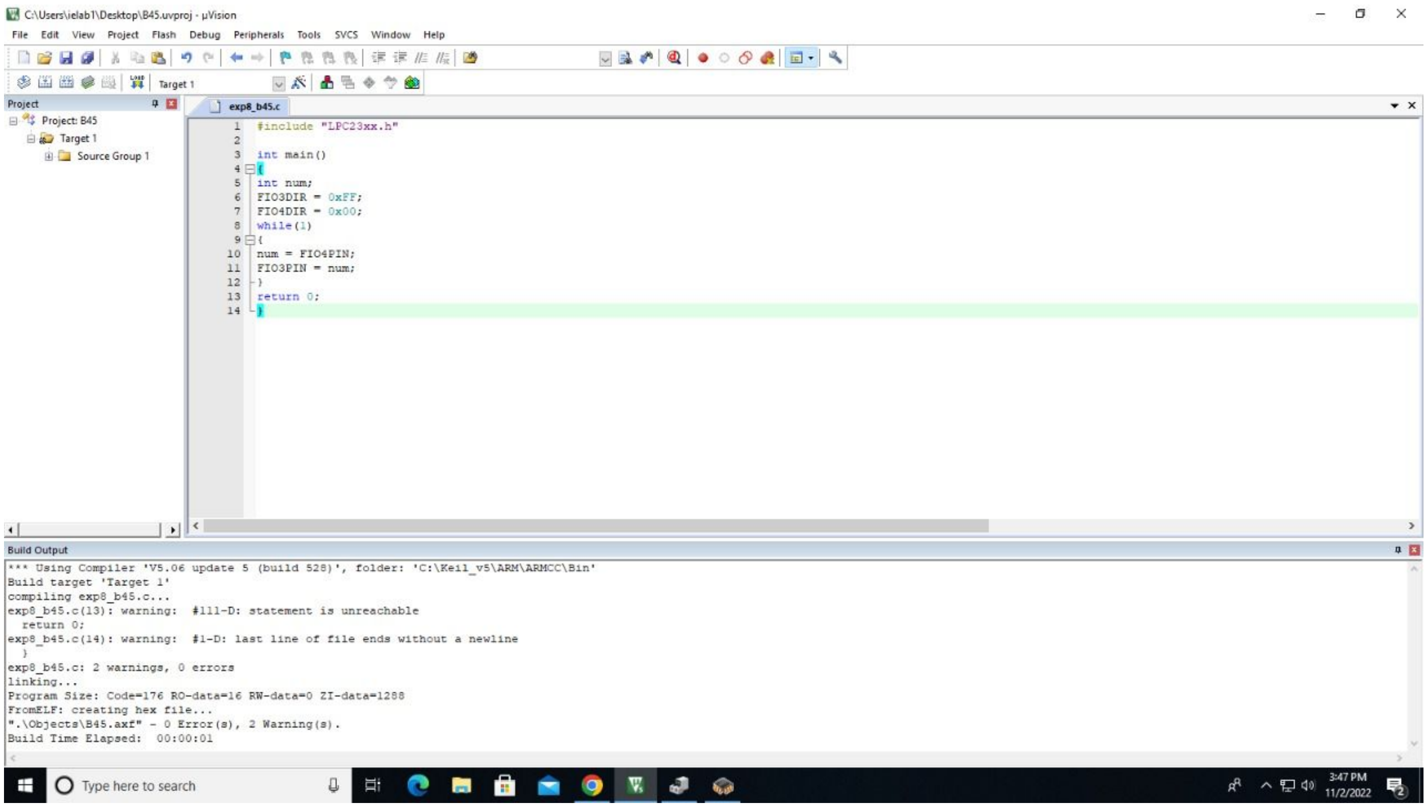
In the previous task, we focussed on output on the LEDs. In this task, we will take input. In particular, read the settings of an 8-way DIP (Dual Inline Package) switch and display it on the LEDs. Use FIO4DIR AND FIO4PIN for data input.

Code:

```
#include "LPC23xx.h"

int main()
{
     int num;
     FIO3DIR = 0xFF;
     FIO4DIR = 0x00;
     while(1)
     {
          num = FIO4PIN;
          FIO3PIN = num;
     }
     return 0;
}
```

First the LEDs are switched off, we read the inputs from the DIP switch and switch on the LEDs

Code Execution:


C:\Users\ielab1\Desktop\B45.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: B45
Target 1
Source Group 1
exp8_b45.c
Build Output
Type here to search
3:47 PM
11/2/2022

```c
#include "LPC23xx.h"

int main()
{
int num;
FIO3DIR = 0xFF;
FIO4DIR = 0x00;
while(1)
{
num = FIO4PIN;
FIO3PIN = num;
}
return 0;
}
```

```
*** Using Compiler 'V5.06 update 5 (build 528)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
compiling exp8_b45.c...
exp8_b45.c(13): warning:  #111-D: statement is unreachable
  return 0;
exp8_b45.c(14): warning:  #1-D: last line of file ends without a newline
  }
exp8_b45.c: 2 warnings, 0 errors
linking...
Program Size: Code=176 RO-data=16 RW-data=0 ZI-data=1288
FromELF: creating hex file...
".\Objects\B45.axf" - 0 Error(s), 2 Warning(s).
Build Time Elapsed:  00:00:01
```

Output:

https://drive.google.com/drive/folders/1_lkvJMq7fv5IkCSHldO1wmExooVFAjka

Inference:

We can see from the output that led's corresponding to DIP switch 2,3,4 are glowing.

Task 3:

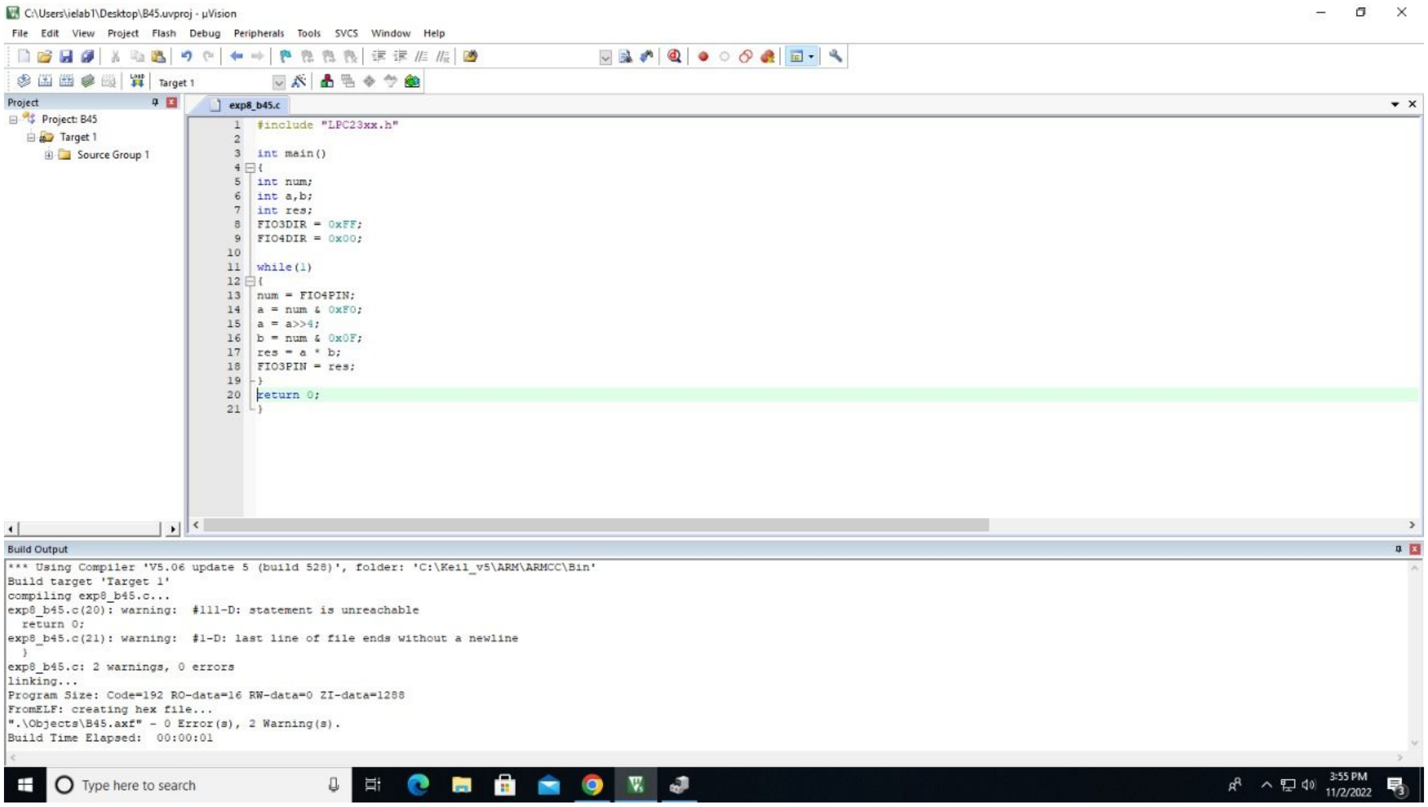
Write a C program to read a DIP switch, split into two nibbles (4 bits), multiply them and display the product on the LEDs.

Code:

```
#include "LPC23xx.h"

int main()
{
     int num;
     int a,b;
     int res;
     FIO3DIR = 0xFF;
     FIO4DIR = 0x00;

 while(1)
 {
          num = FIO4PIN;
          a = num & 0xF0;
          a = a>>4;
          b = num & 0x0F;
          res = a * b;
          FIO3PIN = res;
 }
 return 0;
}
```

First we will take input from the DIP switches, 1-4 DIP switches will correspond to 1st nibble and 5-8 DIP switches will correspond to 2nd nibble. We will perform multiplication on the nibbles. The output will be displayed by the leds.

Code Execution:



C:\Users\ielab1\Desktop\B45.uvproj - µVision

exp8_b45.c

```c
 1  #include "LPC23xx.h"
 2
 3  int main()
 4  {
 5  int num;
 6  int a,b;
 7  int res;
 8  FIO3DIR = 0xFF;
 9  FIO4DIR = 0x00;
10
11  while(1)
12  {
13  num = FIO4PIN;
14  a = num & 0xF0;
15  a = a>>4;
16  b = num & 0x0F;
17  res = a * b;
18  FIO3PIN = res;
19  }
20  return 0;
21  }
```

Build Output

```
*** Using Compiler 'V5.06 update 5 (build 528)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
compiling exp8_b45.c...
exp8_b45.c(20): warning:  #111-D: statement is unreachable
  return 0;
exp8_b45.c(21): warning:  #1-D: last line of file ends without a newline
  }
exp8_b45.c: 2 warnings, 0 errors
linking...
Program Size: Code=192 RO-data=16 RW-data=0 ZI-data=1288
FromELF: creating hex file...
".\Objects\B45.axf" - 0 Error(s), 2 Warning(s).
Build Time Elapsed:  00:00:01
```

Output:

https://drive.google.com/drive/folders/1_lkvJMq7fv5IkCSHldO1wmExooVFAjka

Inference:

Two input nibbles are (0110 and 1010 which in decimal are 6 and 10). We can see this from DIP switches. Switches from 1-4 correspond to 0110 and from 5-8 correspond to 1010.

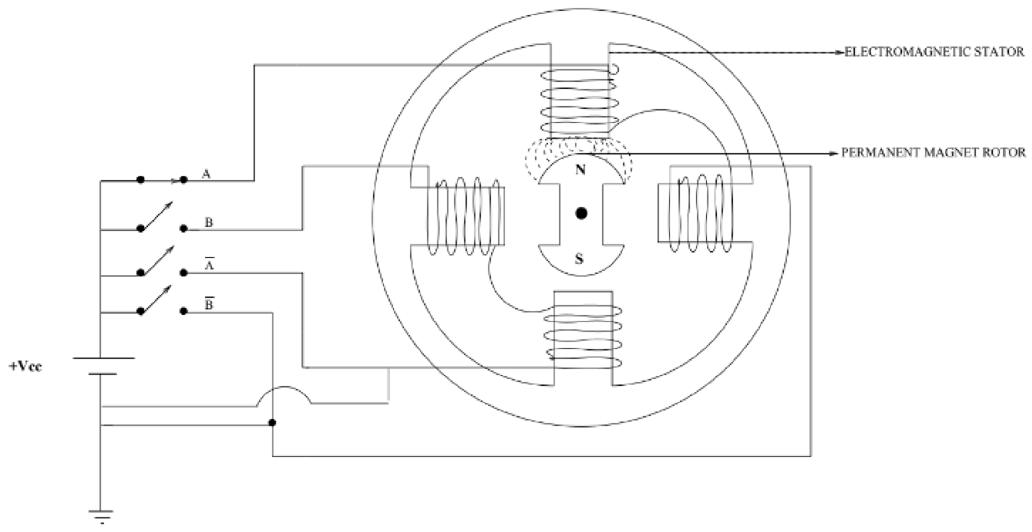
Product of 0110 and 1010 = 00111100 which in decimal is equal to 60 (6*10).

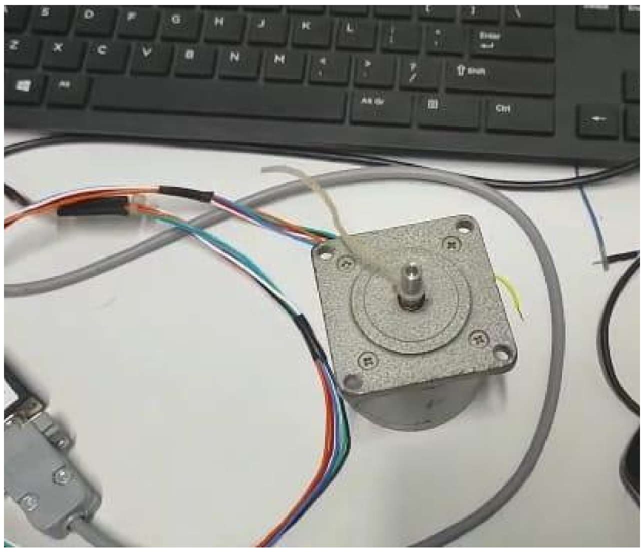
Part 2

AIM:

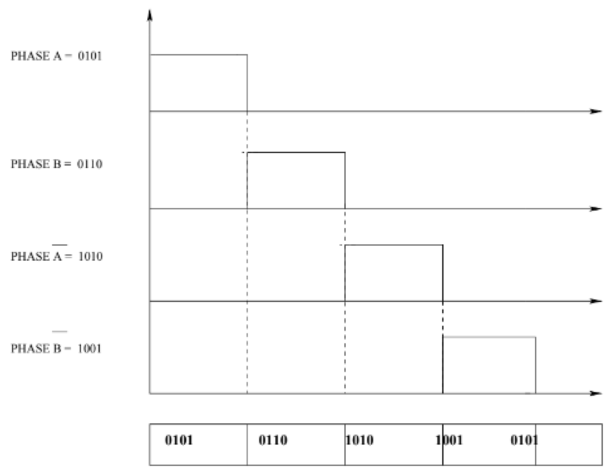
In this experiment, we will interface a stepper motor to the LPC2378 microcontroller. We will begin by describing how a stepper motor works.

Stepper Motor:

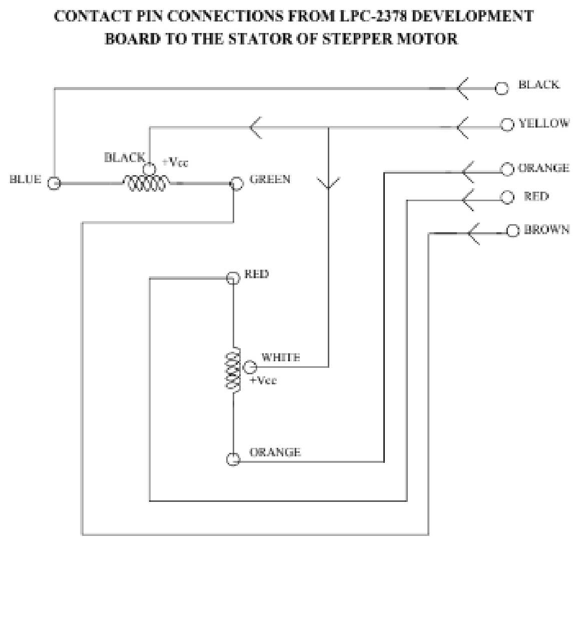




Wave Drive Control:



Unipolar Connection:


CONTACT PIN CONNECTIONS FROM LPC-2378 DEVELOPMENT BOARD TO THE STATOR OF STEPPER MOTOR
BLACK
YELLOW
ORANGE
RED
BROWN
BLUE
BLACK
+Vcc
GREEN
RED
WHITE
+Vcc
ORANGE

Tasks for the Experiment:

Task 1:

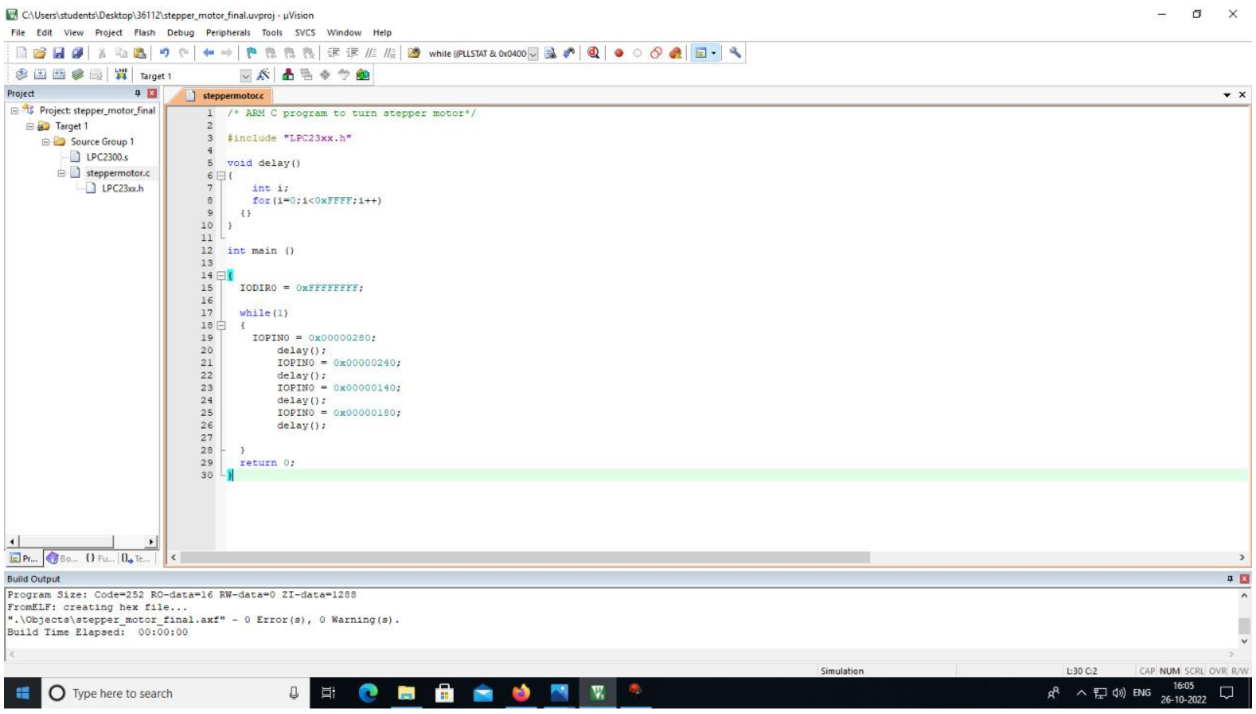
The first task in this experiment involves completion of the program given below to make the motor rotate in a specific direction at a fixed speed.

Code:

```
#include "LPC23xx.h"

void delay()
{
    int i;
    for(i=0;i<0xFFFF;i++)
}

int main ()

{
  IODIR0 = 0xFFFFFFFF;

  while(1)
  {
    IOPIN0 = 0x00000280;
        delay();
        IOPIN0 = 0x00000240;
        delay();
        IOPIN0 = 0x00000140;
        delay();
        IOPIN0 = 0x00000180;
        delay();

  }
  return 0;
}
```

In the delay function we are setting values to control the speed of rotation. Using IODIR0 we configure the output pins. Using IOPIN0 we are exciting different phases one after one to complete one revolution. The code in the main is in a while loop to keep rotating forever until some interrupt.

Code Execution:


C:\Users\students\Desktop\36112\stepper_motor_final.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
while ((PLLSTAT & 0x0400
Target 1
Project
Project: stepper_motor_final
Target 1
Source Group 1
LPC2300.s
steppermotor.c
LPC23xx.h
steppermotor.c
Build Output
Simulation
L:30 C:2
Type here to search

```
/* ARM C program to turn stepper motor*/

#include "LPC23xx.h"

void delay()
{
    int i;
    for(i=0;i<0xFFFF;i++)
  {}
}

int main ()

{
  IODIR0 = 0xFFFFFFFF;

  while(1)
  {
    IOPIN0 = 0x00000280;
        delay();
        IOPIN0 = 0x00000240;
        delay();
        IOPIN0 = 0x00000140;
        delay();
        IOPIN0 = 0x00000180;
        delay();

  }
  return 0;
}
```

```
Program Size: Code=252 RO-data=16 RW-data=0 ZI-data=1288
FromELF: creating hex file...
".\Objects\stepper_motor_final.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

Output:

https://drive.google.com/drive/folders/1_lkvJMq7fv5IkCSHldO1wmExooVFAjka

Inference: From the video we can see that the motor is rotating in a specific direction at a fixed speed.

Task 2:

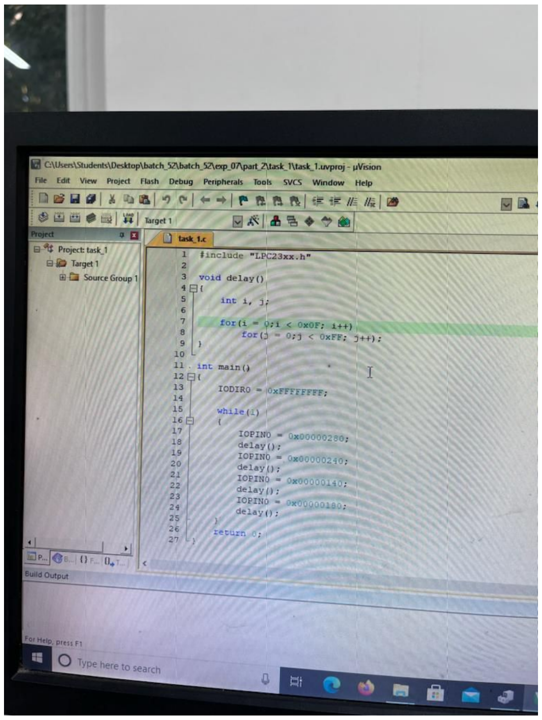
Modify the program given in Task 1 to cause rotation of the stepper motor in both clockwise and anti-clockwise directions. That is, the motor should make a few rotations in clockwise direction, stop and then make a few rotations in the anti-clockwise direction.

Code:

```
/* ARM C program to turn stepper motor*/

#include "LPC23xx.h"

void delay()
{
    int i;
    for(i=0;i<0x0FFF;i++)
    {}
}

int main ()



{
  IODIR0 = 0xFFFFFFFF;

  while(1)
  {

          for(int j=0;j<0x30;j++)
   { IOPIN0 = 0x00000280;
         delay();
         IOPIN0 = 0x00000240;
         delay();
         IOPIN0 = 0x00000140;
         delay();
         IOPIN0 = 0x00000180;
         delay();
       }
       for(int j=0;j<0x11;j++)
       {
       delay();
           }
       for(int j=0;j<0x30;j++)
       {
       IOPIN0 = 0x00000180;
         delay();
         IOPIN0 = 0x00000140;
         delay();
         IOPIN0 = 0x00000240;
         delay();
         IOPIN0 = 0x00000280;
         delay();
       }
       for(int j=0;j<0x11;j++)
       {
       delay();
           }

  }
   return 0;
 }
```

Changing the order of IODIR0 changes the direction of rotation.

Code Execution:



C:\Users\Students\Desktop\batch_52\batch_52\exp_07\part_2\task_1\task_1.uvproj - µVision

```c
#include "LPC23xx.h"

void delay()
{
    int i, j;

    for(i = 0;i < 0x0F; i++)
        for(j = 0;j < 0xFF; j++);
}

int main()
{
    IODIR0 = 0xFFFFFFFF;

    while(1)
    {
        IOPIN0 = 0x00000280;
        delay();
        IOPIN0 = 0x00000240;
        delay();
        IOPIN0 = 0x00000140;
        delay();
        IOPIN0 = 0x00000180;
        delay();
    }
    return 0;
}
```

Output:

https://drive.google.com/drive/folders/1_lkvJMq7fv5IkCSHldO1wmExooVFAjka

Inference:

From the video, we can see that the motor rotates in one direction first, then stops and again rotates in the opposite direction.

Task 3:

The program in Task 1 causes the motor to rotate at approximately 90 rpm. Write a program which will allow the motor to rotate at four different speeds. That is, it should rotate at (say) 30 rpm for one complete revolution, then at (say) 50 rpm for another revolution, then at 70 rpm and finally at 90 rpm for the last revolution.

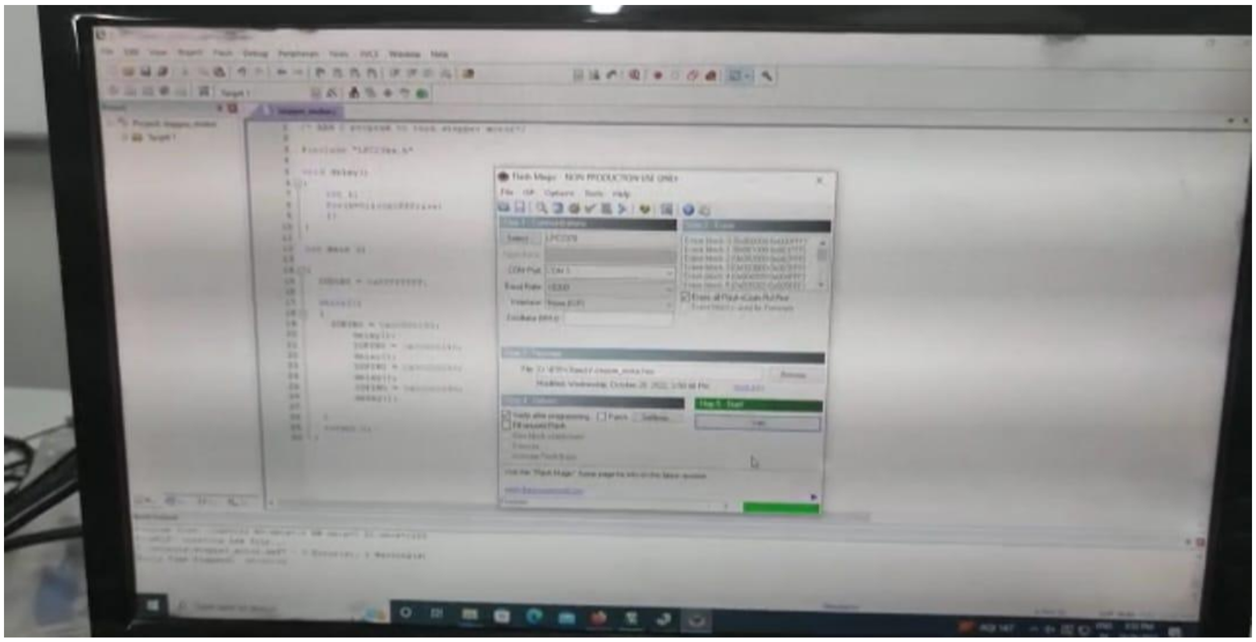
Code:

```
/* ARM C program to turn stepper motor*/

#include "LPC23xx.h"

void delay4()
{
    int i;
    for(i=0;i<0x02FF;i++)
          {}
}
void delay3()
{
    int i;
    for(i=0;i<0x05FF;i++)
          {}
}
void delay2()
{
    int i;
    for(i=0;i<0x0AFF;i++)
          {}
}
void delay1()
{
    int i;
    for(i=0;i<0x0FFF;i++)
          {}
}

int main ()
```

```
{
  IODIR0 = 0xFFFFFFFF;

  while(1)
  {

          for(int j=0;j<0x34;j++)
   { IOPIN0 = 0x00000280;
        delay1();
        IOPIN0 = 0x00000240;
        delay1();
        IOPIN0 = 0x00000140;
        delay1();
        IOPIN0 = 0x00000180;
        delay1();
      }
      for(int j=0;j<0x10;j++)
      {
      delay1();
          }
      for(int j=0;j<0x34;j++)
   { IOPIN0 = 0x00000280;
        delay2();
        IOPIN0 = 0x00000240;
        delay2();
        IOPIN0 = 0x00000140;
        delay2();
        IOPIN0 = 0x00000180;
        delay2();
      }
            for(int j=0;j<0x10;j++)
      {
      delay1();
          }
       for(int j=0;j<0x34;j++)
   { IOPIN0 = 0x00000280;
        delay3();
        IOPIN0 = 0x00000240;
        delay3();
        IOPIN0 = 0x00000140;
        delay3();
        IOPIN0 = 0x00000180;
        delay3();
      }
            for(int j=0;j<0x10;j++)
      {
      delay1();
          }
       for(int j=0;j<0x34;j++)
   { IOPIN0 = 0x00000280;
        delay4();
        IOPIN0 = 0x00000240;
        delay4();
        IOPIN0 = 0x00000140;
        delay4();
        IOPIN0 = 0x00000180;
        delay4();
      }
            for(int j=0;j<0x10;j++)
      {
      delay1();
          }




 }
  return 0;
}
```

The four delay functions produces different rotation speeds

Code Execution:



Output:

https://drive.google.com/drive/folders/1_lkvJMq7fv5IkCSHldO1wmExooVFAjka

Inference:

From the video, we can see that the motor first rotates at 30 rpm for one revolution then speed increases to 50 rpm for one revolution and finally increases to 70 rpm for another revolution.