**ARM C-Interfacing – Emulation of Switch LED and Stepper Motor Control**

* EE20B052

**Aim:** Using C interfacing to implement the following tasks

1. Read the status (binary position) of the switch and use the LEDs (8 LEDs are provided) to display the status of each of the 8-bit DIP switch
2. Control the Stepper motor rotation

**Tasks:**

1. Write a program (in C) to dis-assemble a byte into two nibbles from the DIP switch states, multiply and product the LED
2. **Flow chart**

Start

LOAD FI03DIR = 0xFF

LOAD FIO4DIR = 0x00

READ FI04PIN

Separate HIGH and LOW nibbles

Multiply Both

Store RESULT in FIO3PIN

End

1. **Code**

#include "LPC23xx.h"

int main()

{

    int highByte, lowByte;

    FIO3DIR = 0xFF;

    FIO4DIR = 0x00;

    int a = FIO4PIN;

    lowByte = a & 0x0F;

    highByte = a & 0xF0;

    highByte = highByte >> 4;

    FIO3PIN = highByte \* lowByte;

}

1. Rotate stepper motor in clockwise direction
2. **Flowchart**

Start

LOAD IODIR0 = 0Xffffffff

LOAD IOPINO 4 times

Delay

Repeat

End

1. **Code:**

#include <LPC23xx.h>

void delay()

{

    int i, j;

    for (int i = 0; i < 0xFF; i++)

        for (int j = 0; j < 0xFF; j++)

            ;

}

void rotateClockwise()

{

    while (1)

    {

        IOPIN0 = 0x00000240;

        delay();

        IOPIN0 = 0x00000140;

        delay();

        IOPIN0 = 0x00000180;

        delay();

        IOPIN0 = 0x00000280;

        delay();

    }

}

int main()

{

    IODIR0 = 0xFFFFFFFF;

    rotateClockwise();

    return 0;

}

1. Rotate Stepper Motor with step angle 5° by 180° in clockwise direction
2. **Flowchart**

Start

LOAD IODIR0 = 0xFFFFFFFF

Initialize step\_angle and required\_angle

Calculate step\_count and loop\_count

Rotate by required\_angle

End

1. **Code**

#include <LPC23xx.h>

void delay()

{

    int i, j;

    for (int i = 0; i < 0xFF; i++)

        for (int j = 0; j < 0xFF; j++)

            ;

}

void rotateClockwise(int n)

{

    for (int i = 0; i < n; i++)

    {

        IOPIN0 = 0x00000240;

        delay();

        IOPIN0 = 0x00000140;

        delay();

        IOPIN0 = 0x00000180;

        delay();

        IOPIN0 = 0x00000280;

        delay();

    }

}

int main()

{

    IODIR0 = 0xFFFFFFFF;

    float step\_angle = 5;

    float angle = 180;

    int step\_count = angle / step\_angle;

    int n = step\_count / 4.0;

    rotateClockwise(n);

    return 0;

}

**Inferences:**

* Data in I/O registers can be used for READ/WRITE by appropriately changing DIR Register appropriately
* A stepper motor rotates by a fixed angle when voltage is applied
* The precision of angle of rotation of a Stepper Motor can be increased by increasing the number of teeth
* The direction of rotation of Stepper Motor can be altered by reversing the order of input given to the I/O pins
* Bitmasks can be used to dis-assemble the bits in a word
* Loops can used for creating delays

- - - - - - - - - - - - - - - - - - -END- - - - - - - - - - - - - - - - -