SUHAS C

EE20B132

AIM:

Using C-interfacing, use C-programming, to implement the following tasks:

(i) 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.

(ii) Stepper motor control using Vi Microsystems ViARM 7238 development. Due to ongoing pandemic, only emulated version of this experiment is intended here.

QUESTION – 1:

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

a) Code:

#include "LPC23xx.h"

int main()

{

FIO3DIR = 0xFF ; //to make FIO3 as output

FIO4DIR = 0x00 ; //to make FIO4 as input.

while(1) // to run the loop continuously.

{

int c1,c2,mul;

int a = FIO4PIN & 0xFF ; // enable only the last 8 bits.

c1 = a & 0x0F ;

c2 = a & 0xF0 ;

c2 = c2 >> 4 ; // right shift the top 4 bits.

mul = c1\*c2 ;

FIO3PIN = mul ;

}

return 0;

}

FLOWCHART –

Diagram

Description automatically generated

QUESTION – 2:

Modify the demo code (StpprMtrCntrl.c) supplied to demonstrate the control of stepper motor to rotate in opposite

direction

a) Code:

/\* ARM C program to run Stepper Motor \*/

#include "LPC23xx.h"

void delay(void)

{

int i,j;

for(i=0; i<0xff;i++)

for(j=0; j<0XFF;j++);

}

int main(void)

{

IODIR0 = 0XFFFFFFFF;

int i=0;

while(i)

{

IOPIN0=0X00000240;

delay();

IOPIN0=0X00000140;

delay();

IOPIN0=0X00000180;

delay();

IOPIN0=0X00000280;

delay();

}

return 0;

**}**

b) FLOW CHART:

A picture containing text, whiteboard

Description automatically generated

QUESTION – 3:

Modify the demo code (StpprMtrCntrl.c) supplied to demonstrate the control of the stepper motor to rotate 80 degrees and stop, assuming the step angle as 2 degrees ( motor rotates 2 degrees/step).

1. Code:

/\*ARM C Programming for stepper motor to stop at certain angle \*/

#include "LPC23XX.h"

void delay (void)

{

int i,j ;

for(i=0 ; i<0xFF ; i++ )

{

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

{}

}

}

int main(void)

{

IODIR0 = 0xFFFFFFFF;

int angle = 180; // stop at 180 degrees

for (int i = 0; i < 45 ;i++ ) // 45 loops because each value set takes 1 degree and each loop takes 4 degrees.

{

IOPIN0 = 0x00000240 ;

delay();

IOPIN0 = 0x00000140 ;

delay();

IOPIN0 = 0x00000180 ;

delay();

IOPIN0 = 0x00000280 ;

delay();

}

return 0;

}