Experiment 4: ARM C-Interfacing - Emulation of Switch LED and Stepper Motor Control

-KEERTHANA RACHURI
EE20B102

Brief outline of the target of the experiment:

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 Microsystem's ViARM 7238 development board. Due to ongoing pandemic, only emulated version of this experiment is intended here

Questions:

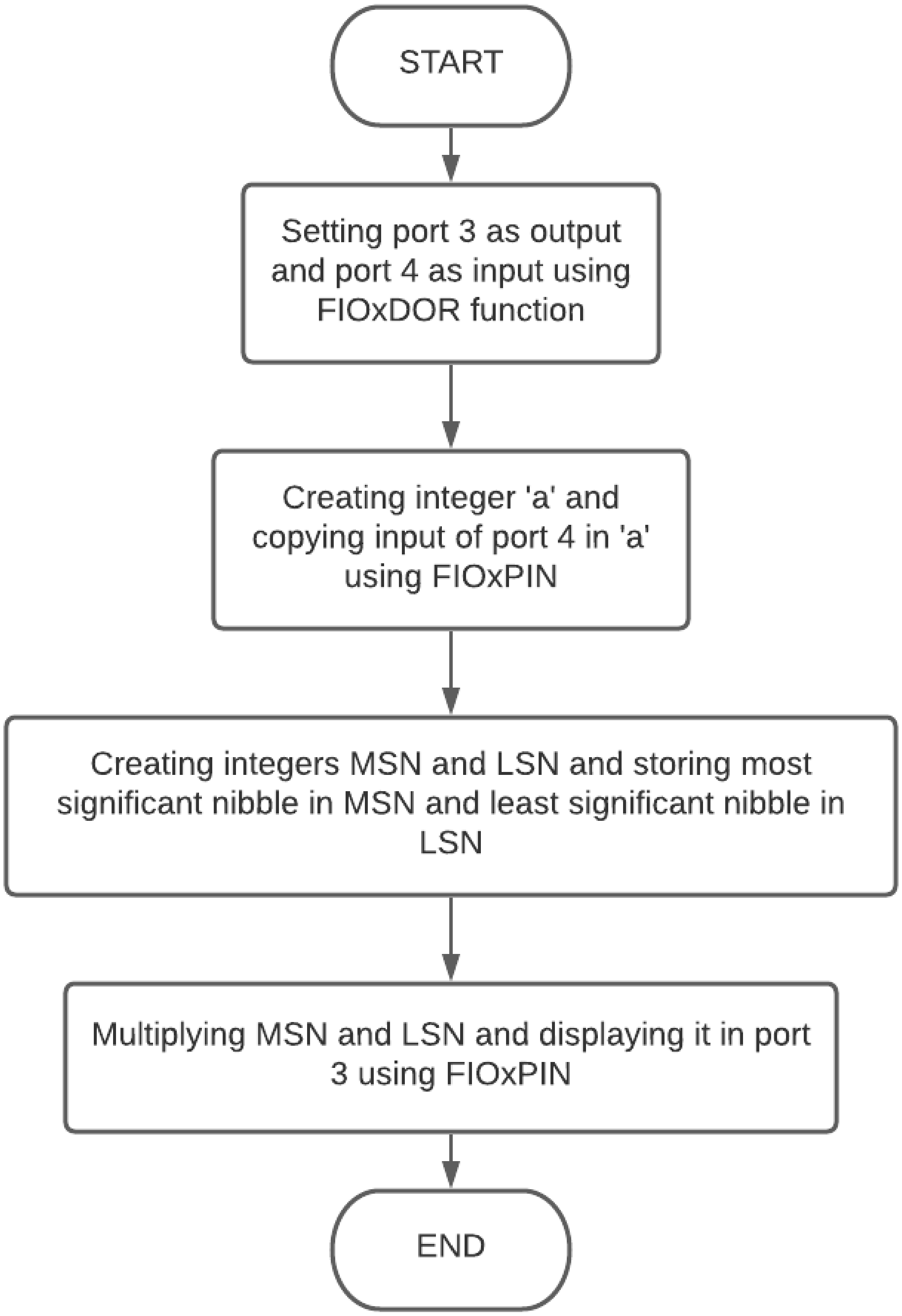
1. Write a program (in C) to dis-assemble a byte into two nibbles from the DIP switch states, multiply and display the product in the LED.

    a. **Code**

```
#include "LPC23xx.h"
int main()
{

   int Byte,HigherNibble,LowerNibble,Product;

   FIO3DIR = 0xFF; // making pin 3 as an output port
   FIO4DIR = 0x00;// making pin 4 as an input port

   while(1)
   {
      Byte = FIO4PIN; // copying input of pin 4 to Byte
                          LowerNibble = Byte & 0x0F; // Storing the least
significant nibble of Byte in LowerNibble
```

```
        HigherNibble = Byte& 0xF0; // Storing the most significant nibble of
Byte in HigherNibble
        HigherNibble = HigherNibble>> 4; // Shifting the HigherNibble by 4
bits right
        Product = HigherNibble*LowerNibble; // Storing the product of
HigherNIbble and LowerNibble in Product
                                  FIO3PIN = Product; // Displaying the value in Product
through Port 3
     }
    return 0;
}
```

b. Flow chart


START
Setting port 3 as output and port 4 as input using FIOxDOR function
Creating integer 'a' and copying input of port 4 in 'a' using FIOxPIN
Creating integers MSN and LSN and storing most significant nibble in MSN and least significant nibble in LSN
Multiplying MSN and LSN and displaying it in port 3 using FIOxPIN
END

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

   a. Code

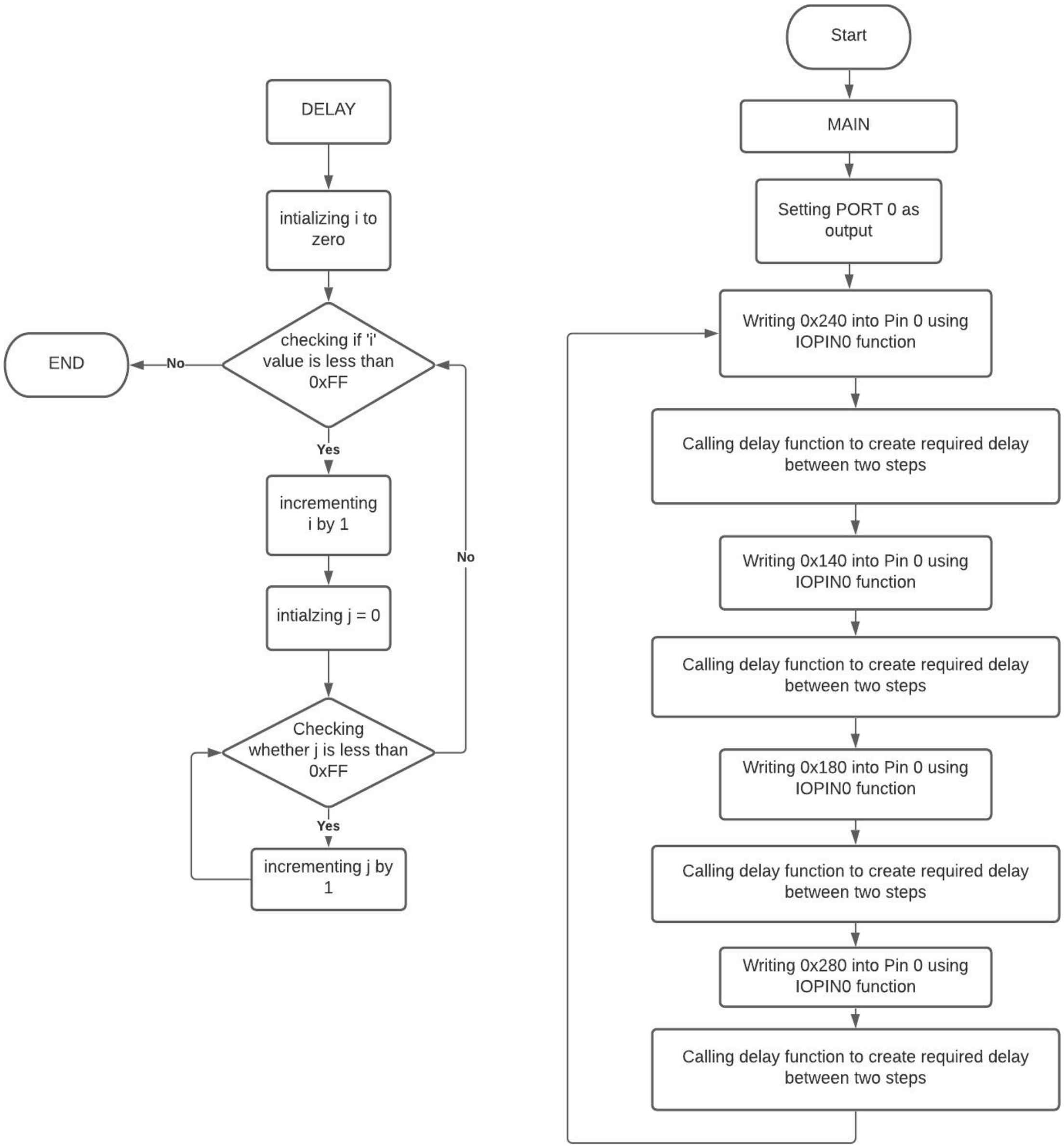
```
/* ARM C program to run Stepper Motor */
#include "LPC23xx.h"
void delay(void)
{ int i,j;
for(i=0; i<0xff;i++)
for(j=0; j<0XFF;j++);
}
int main(void)
{ IODIR0 = 0XFFFFFFFF; // making port0 as output port
while(1) // To rotate stepper motor in anti-clock wise direction
{ IOPIN0=0X00000240;
delay();
IOPIN0=0X00000140;
delay(); // creating delay between every 2 steps
IOPIN0=0X00000180;
delay();
IOPIN0=0X00000280;
delay();
}
return 0;
}
```

Explanation for the IOPINO values:

The stepper motor is connected to the 6,7,8,9 pins of the ARM kit . So to make the stepper motor rotate in a particular direction, the currents’ in the windings should be given in a particular direction. To control the current flow switches Q1,Q2,Q3,Q4 are used . For example, in one of the alienations Q1,Q2,Q3,Q4 must be 1,0,0,1 respectively . So the value that should be given to IOPINO is 001001000000 which in hexadecimal form is written as 0x240. Similarly when

Q1,Q2,Q3,Q4 values are (1,0,1,0) ,(0,1,1,0), (0,1,0,1) the hexadecimal values given to IOPIN0 are 0x280,0x180,0x140 respectively.

b . Flow chart

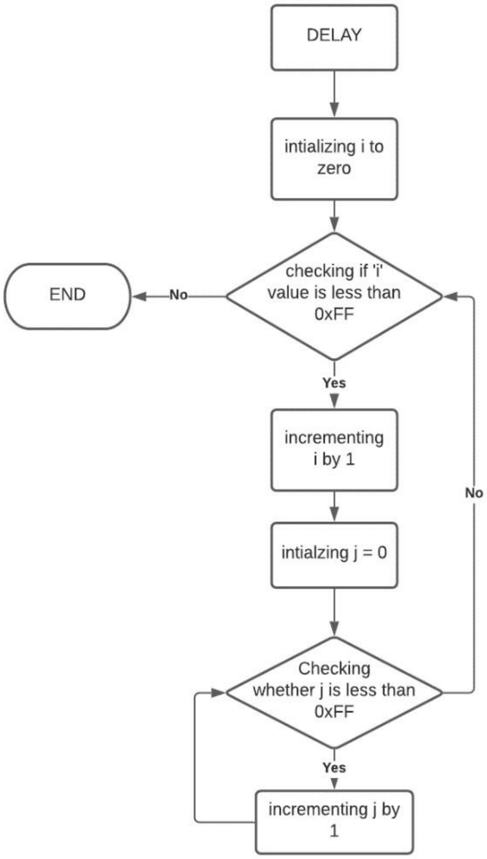

DELAY
intializing i to zero
checking if 'i' value is less than 0xFF
No
END
Yes
incrementing i by 1
intialzing j = 0
Checking whether j is less than 0xFF
No
Yes
incrementing j by 1
Start
MAIN
Setting PORT 0 as output
Writing 0x240 into Pin 0 using IOPIN0 function
Calling delay function to create required delay between two steps
Writing 0x140 into Pin 0 using IOPIN0 function
Calling delay function to create required delay between two steps
Writing 0x180 into Pin 0 using IOPIN0 function
Calling delay function to create required delay between two steps
Writing 0x280 into Pin 0 using IOPIN0 function
Calling delay function to create required delay between two steps

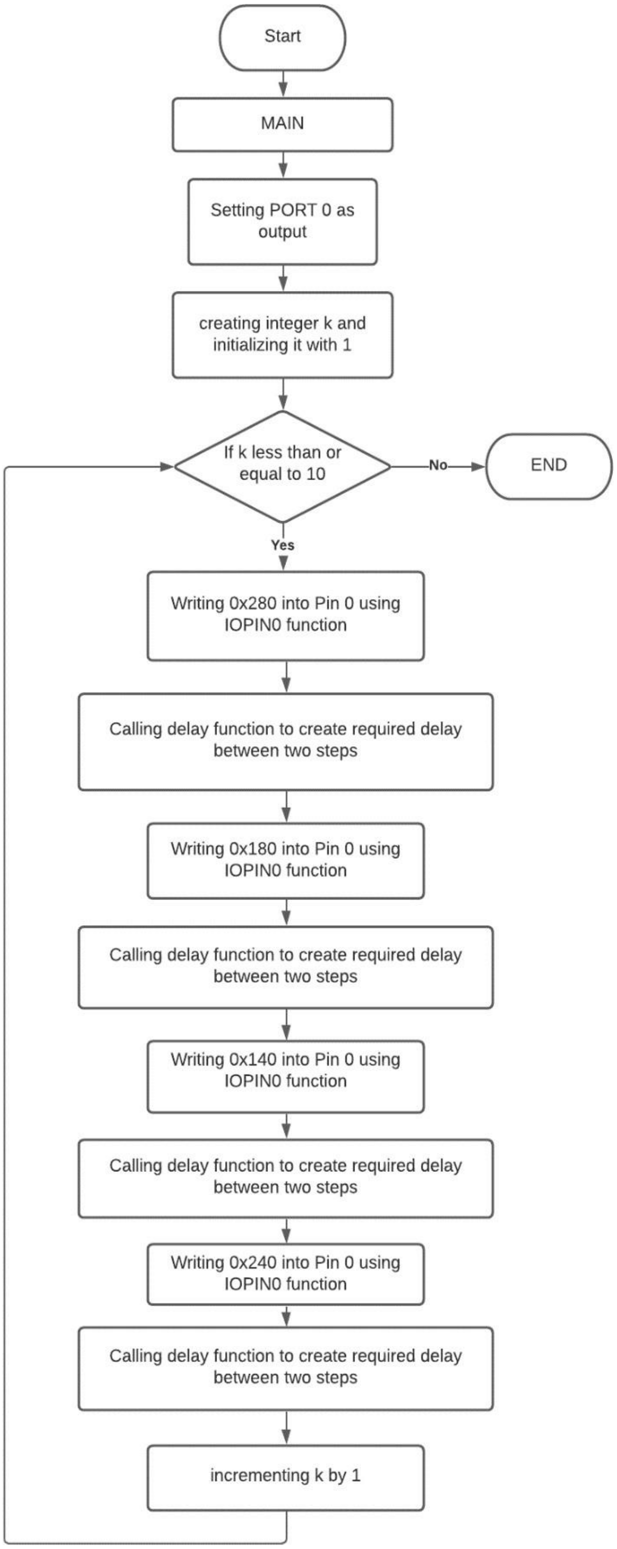
3. Modify the given demo code supplied to demonstrate the control of the stepper motor 80 degrees and stop, assuming the step angle as 2 degrees (motor rotates 2 degrees /step)

   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; // To make port 0 as output port
    int k=1; // initialise the counter by 1
while(k<=10)     // each step rotates by 2 degrees
                 // each cycle turns 4*2 = 8 degrees
               // total loops that should be created are 80/8 = 10 loops
{ IOPIN0=0X00000280;
delay();
IOPIN0=0X00000180;
delay(); // creating delay between 2 steps
IOPIN0=0X00000140;
delay();
IOPIN0=0X00000240;
delay();
    k++; // increments k by 1
}
return 0;
}
```

   b. **Flow chart**


DELAY
intializing i to zero
checking if 'i' value is less than 0xFF
No
END
Yes
incrementing i by 1
intialzing j = 0
No
Checking whether j is less than 0xFF
Yes
incrementing j by 1


Start
MAIN
Setting PORT 0 as output
creating integer k and initializing it with 1
If k less than or equal to 10
No
END
Yes
Writing 0x280 into Pin 0 using IOPIN0 function
Calling delay function to create required delay between two steps
Writing 0x180 into Pin 0 using IOPIN0 function
Calling delay function to create required delay between two steps
Writing 0x140 into Pin 0 using IOPIN0 function
Calling delay function to create required delay between two steps
Writing 0x240 into Pin 0 using IOPIN0 function
Calling delay function to create required delay between two steps
incrementing k by 1

Inferences:

- I learnt the features of ViARM-2378 development board and C-interfacing.
- I learnt about stepper motor and controlling it by connecting to ViARM-2378 development board.
- I have learnt how to use FIOxPIN function to get or send the data using ports.
- I have also learnt about how to set ports as output or input using the functions like FIOxDIR.