ARM C-interfacing- Emulation of Switch LED and Stepper Motor Control

Aneesh Kandi EE20B009

27 October 2021

Contents

1 Aim **2**

2 Program Codes **2**
2.1 Problem 1 2
2.1.1 Code 2
2.1.2 Code Explained 2
2.1.3 Result 4
2.2 Problem 2 5
2.2.1 Code 5
2.2.2 Code Explained 5
2.2.3 Result 7
2.3 Problem 3 7
2.3.1 Code 7
2.3.2 Code Explained 8
2.3.3 Result 8

3 Inferences and Learnings **8**

4 Link to all Codes **8**

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

2 Program Codes

2.1 Problem 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.

2.1.1 Code

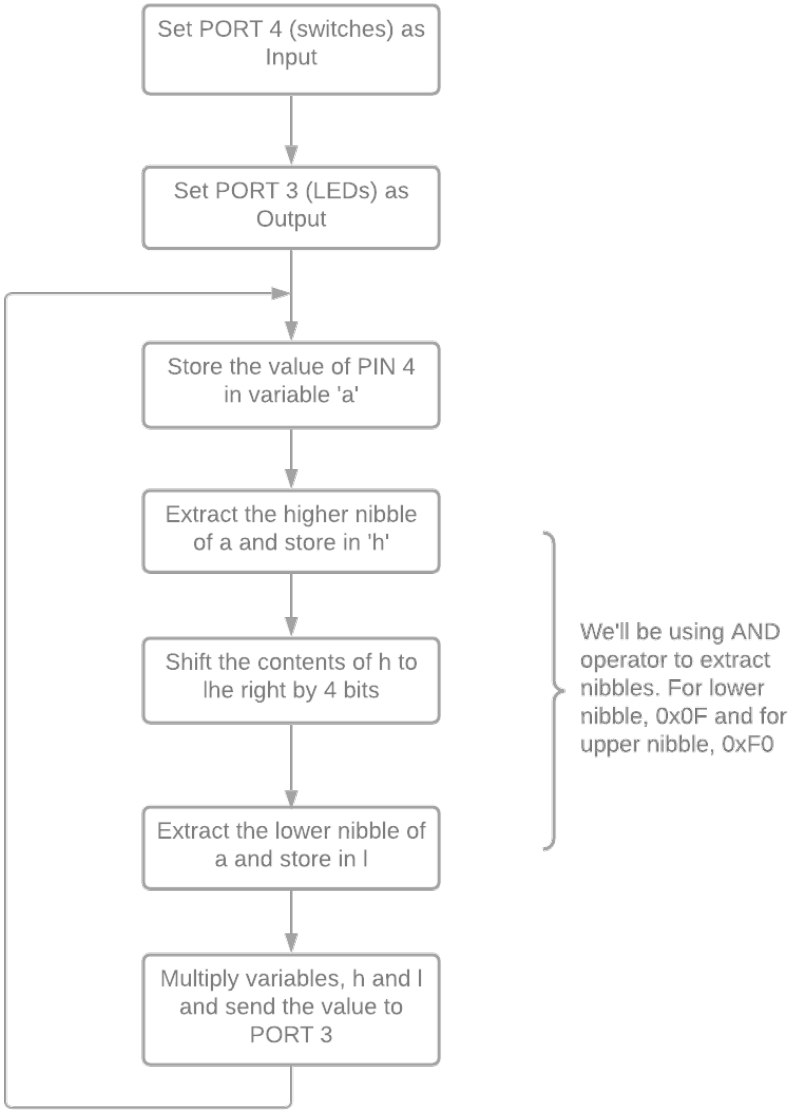
```
#include "LPC23xx.h"

int main(void)
{
  int h, l, a, m;
  FIO4DIR = 0x00000000;    // Setting PORT 4 as Input
  FIO3DIR = 0X000000FF;    // Setting PORT 3 as Output
  while(1)
  {
    a = FIO4PIN;
    h = a & 0x000000F0;    // Extracting the Higher Nibble
    h = h >> 4;    // Shifting the number by a nibble towards right
    l = a & 0x0000000F;    // Extracting the Lower Nibble
    m = h*l;               // Multiplying both the nibbles
    FIO3PIN = m;         // Sending the Multiplied values to PORT 3
  }
  return 0;
}
```

2.1.2 Code Explained

In this problem, we're interfacing switches and LEDs present in the ViARM Development board. The flowchart below will explain the logic.

We are using Fast IO here because of the superior speed it provides. Both the Input and Output are external peripherals and it takes time for the data transfer to occur. So, we use Fast IO.

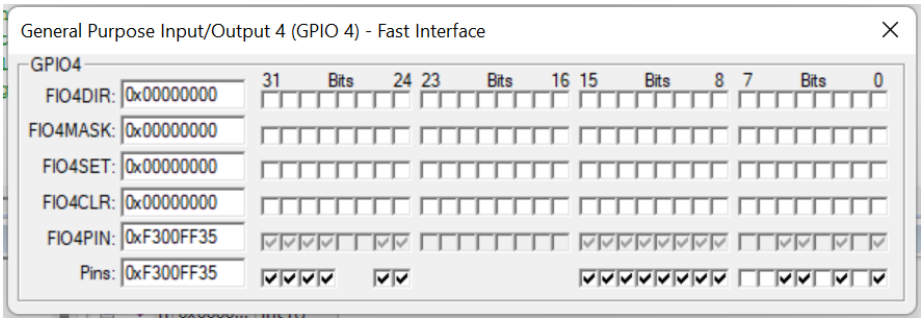


Interfacing LEDs and Switches

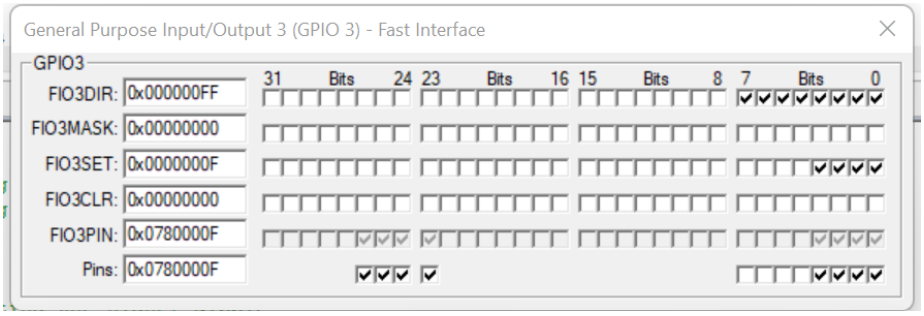
2.1.3 Result

According to the figures:

a = 0x35
h = 0x03
l = 0x05
m = 0x0F



PORT 4 - 0x00110101



PORT 3 - 0x0000000F

2.2 Problem 2

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

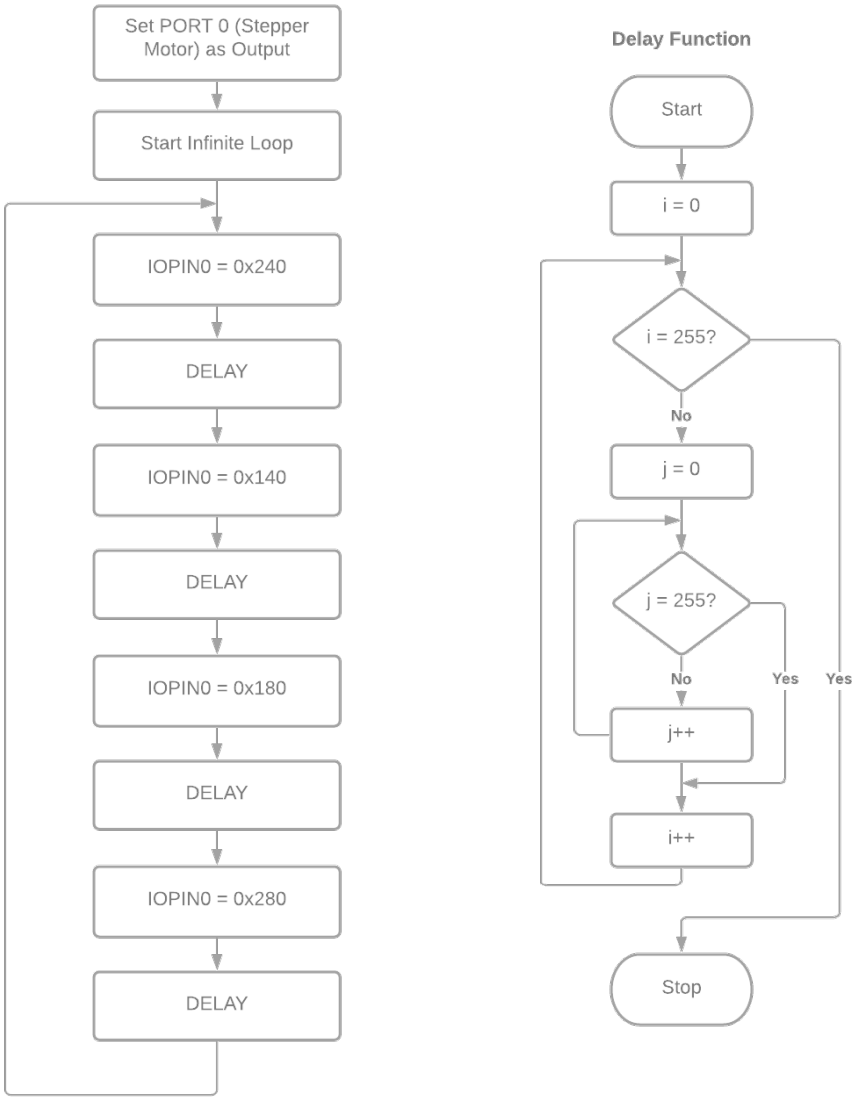
2.2.1 Code

```
#include "LPC23xx.h"
void delay(void)
{
  int i,j;
  for(i=0; i<0xff;i++)
  for(j=0; j<0XFF;j++);
}
int main(void)
{
  IODIR0 = 0XFFFFFFFF;
  // The motor is configured such that each step is 2 degrees
  while(1)                  // Continuous rotation
  {
    IOPIN0=0X00000240;
    delay();
    IOPIN0=0X00000140;
    delay();
    IOPIN0=0X00000180;
    delay();
    IOPIN0=0X00000280;
    delay();
  }
    return 0;
}
```

2.2.2 Code Explained

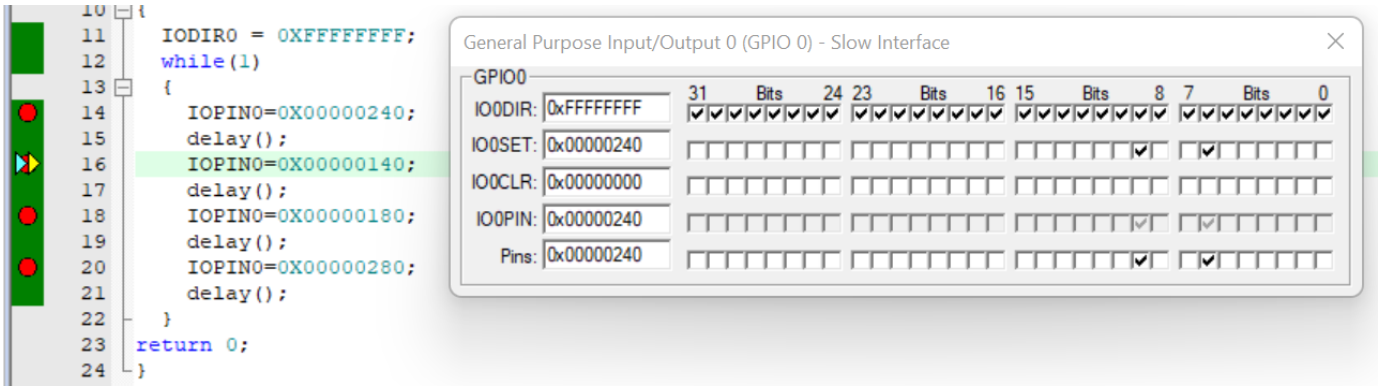
The Stepper Motor is connected to ViARM board using the pins 6-9 of PORT 0. The motor has 4 coils and each coil is activated in a particular order for Clockwise or Anti-clockwise direction.

This motor is configured in such a way that, each step = 2°. So one loop is equal to 4 steps which means, the blade of the motor rotates by 8°. To reverse the direction of spin, we need to reverse the order of activating the coils.



Flowchart for the program

2.2.3 Result



PIN value = 0x240

2.3 Problem 3

Modify the Stepper Motor code to rotate the shaft of the motor by specific angle.

2.3.1 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)
{
  IODIR0 = 0XFFFFFFFF;
  // Each loop rotates the wheel by 8 degrees
  // So for 80 degree rotation we iterate it for 10 times
  for(int i = 0; i< 10; i++)
  {
    IOPIN0=0X00000240;
    delay();
    IOPIN0=0X00000140;
    delay();
    IOPIN0=0X00000180;
    delay();
    IOPIN0=0X00000280;
    delay();
  }
    return 0;
}
```

2.3.2 Code Explained

This code is very similar to the 2nd code but the loop is iterated for a certain number of times. This code rotates the shaft of the motor by 80°. So for that we iterate it for 10 times (As we know that each iteration = 4 steps and each step = 2°).

2.3.3 Result

Same Results as the previous problem

3 Inferences and Learnings

- Interfacing C with ARM in Keil
- Learnt about working of the Stepper Motor and the various terminology involved
- Explored the ViARM Development Board and learned about the working and circuits of various peripherals present
- Learnt about concepts like Fast IO
- Learnt how to use breakpoints in Keil

4 Link to all Codes

https://drive.google.com/file/d/15hRowtnua5UKMi8SPPVCgfeBhtNt5I1i/view?usp=sharing
This drive link consists of a zipped file containing all the files related to the codes.

—THANK YOU—