EE2016 Micropprocessor Lab & Theory July-Nov 2024

Dr. R. Manivasakan, EE Department, IIT, Madras.

Experiment 7: ARM C-Interfacing - Interfacing of Switch, LED and Stepper Motor and their Control

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 Equipments, Hardwares / Softwares Required

The list of equipments, components required are:

1. Sun Tech's 2148 embedded development board and accessaries
2. RS-232 cable
3. Keil microvision 5
4. USB - serial converter (this is a must when the PC loaded with keil doesnt have a serial port).
5. flash magic
6. Burn o-mat
7. Stepper motor

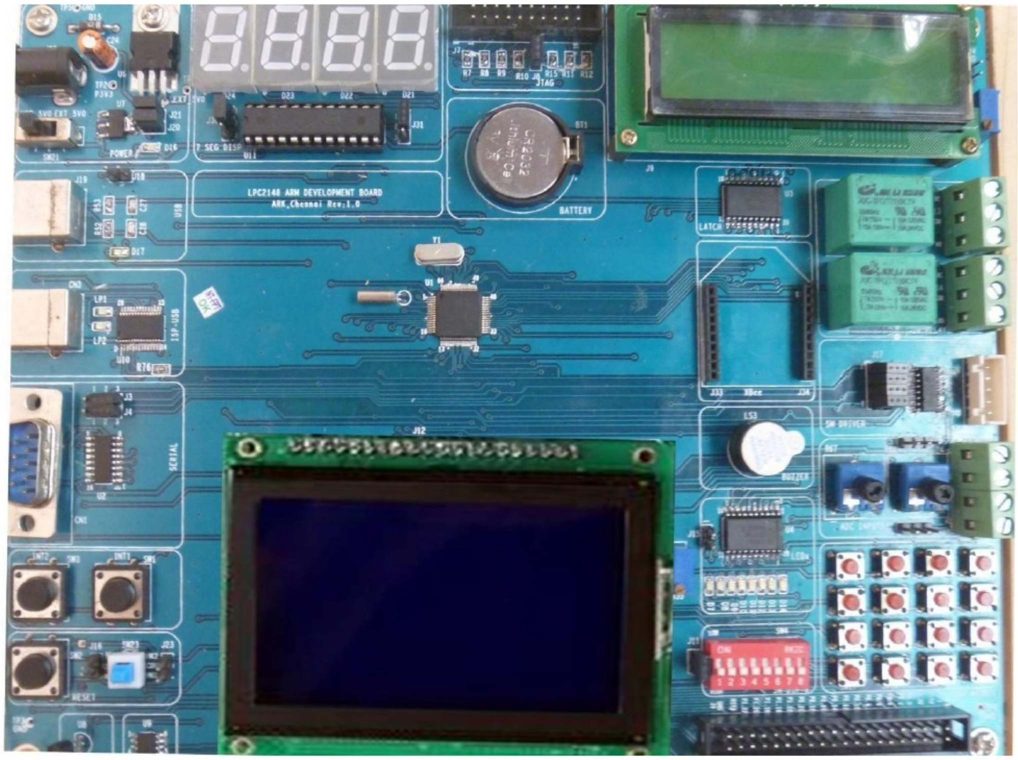
The hardware components are given here just not to loose the context of the experiment. Otherwise it is a purely emulation based experiment (due to the ongoing pandemic).

3 Background Information

In this Section we would discuss the following background information which are very much essential to do the above experiment

1. ARM-2148 Development Board
2. LED Interfacing in LPC 2378 ARM processor
3. Stepper motor control by LPC 2378 ARM processor
4. Demo programs (which could be used to understand and write the code for the tasks defined at the end).

3.1 ARM-2148 Microcontroller - Anatomy



The microcontroller which was used in the embedded development board (which we bought) is from NXP semiconductors and is LPC 2148. LPC stands for Low Power Consumption. The LPC2148 is based on the ARM7TDMI-S CPU so we will discuss aspects of ARM below.

3.1.1 The ARM and ARM7TDMI-S

In this experiment, we will use an ARM7-based processor, namely LPC2378. The ARM core here is called the ARM7TDMI-S (where T stands for Thumb Instructions, D for on-chip debugging support, M for multiplier, I for embedded In circuit emulation hardware and S for the synthesizable option based on RTL provided). ARM7TDMI was the first of a range of processors introduced in 1995 by ARM. ARM7TDMI is the first core to include the Thumb instruction set. The Thumb instruction set contains 16-bit instructions. These serve as a 'compact shorthand' for a subset of the 32-bit instructions of the ARM. For 32-bit systems, ARM instructions are recommended for higher performance. Thumb instructions are appropriate for memory constrained systems.

3.1.2 Block Diagram of LPC2148

The LPC2148 block diagram is shown in Fig. ??. The instructions for the board to be used in the experiments are available as a separate pdf file.

3.1.3 Brief Description of the Software Environment

In this laboratory experiment, programming will be done in C. The instructions for writing C language programs in Keil µVision are as follows.

1. Open Keil uVision and Click project > New uVision Project
2. Select the 'Legacy device Database' under device dropdown. Search for LPC2148 and select it
3. Choose Yes when prompted to copy LPC2100.s Startup file

4. Right Click on Target 1, choose 'Options for Target 1'.
5. Under the Target tab tick 'Use Microlib'. Under Output tab tick 'Create Hex File'
6. Right click Source Group1 under Target in the left side of the keil window. Select Add new Item to group ..
7. Select C file in the list, give a filename and save.
8. Write your program.
9. Go to Project tab and click on Build Target (or press F7).

3.1.4 How to Upload the Program to the Flash on LPC2378 in the Board

The instructions for uploading the program on to the flash memory on the LPC2378 are as follows.

1. Run FlashMagic.
2. Select LPC2378.
3. Choose the correct COM port to which the USB to Serial converter is connected. If you do not know the COM Port, Use Device Manager to find out.
4. Set baud rate to 19200 and Oscillator to 12 MHz.
5. Choose the Hex file to be transferred.
6. On the board, put the "Program/Execute" slide Switch to "Program" mode (slide it down)
7. On FlashMagic click 'Start' to start programming.
8. Once programming is done, put the "Program/Execute" slide Switch to "Execute" mode (slide it up) and press reset to execute the program.

3.2 LEDs and DIP switches in LPC-2148 Board

LEDs and DIP switches are most common in development boards. here, DIP switches are the red ones in bottom right (just before the key pad button matrix).

3.2.1 LEDs

Light Emitting Diodes are the most commonly used components usually for displaying Pin's digital state. LPC-2148 has 8 LED's that are connected to the microcontroller port line. One has to close the jumper J38 to use LEDs.

Used port lines: LED0 - LED7 : P3.0 - P3.7

3.2.2 Switches

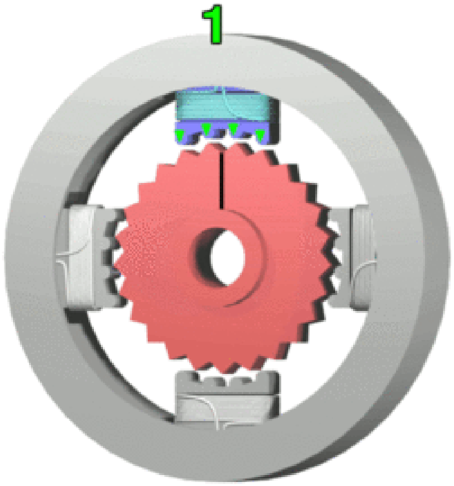
Switches are devices that have two positions - ON and OFF, which have a toggle to establish or break a connection between two contacts. The LPC-2148 development board has one 8-way Dip switch.

Used port lines: SW0 - SW7 : P4.0 - P4.7

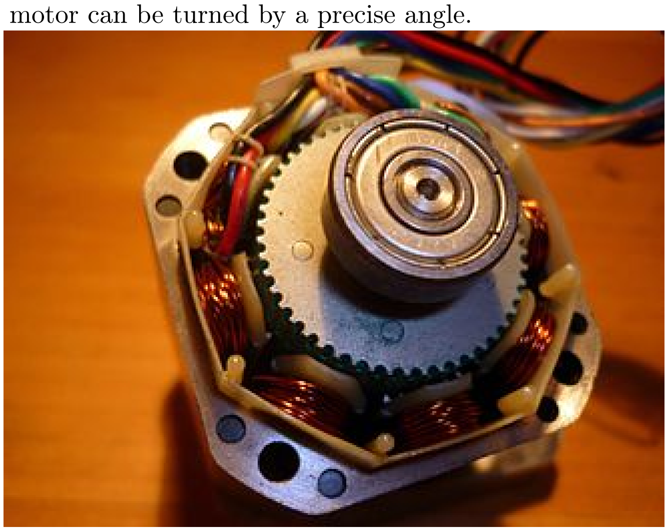
3.3 Stepper Motor

A stepper motor simply rotates by a fixed angle when the voltage is applied.

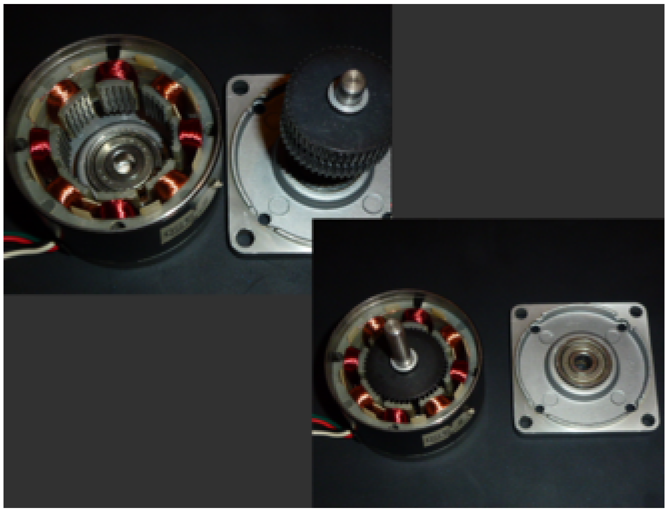
A stepper motor is a brushless electric DC motor with a large number of poles (on both rotor and stator), such that a full rotation is divided into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed.



Stepper motors effectively have multiple "toothed" electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external driver circuit or a micro controller. To make the motor shaft turn, first, one electromagnet is given power, which magnetically attracts the gear's teeth. When the gear's teeth are aligned to the first electromagnet, they are slightly offset from the next electromagnet. This means that when the next electromagnet is turned on and the first is turned off, the gear rotates slightly to align with the next one. From there the process is repeated. Each of those rotations is called a "step", with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise angle.



The circular arrangement of electromagnets is divided into groups, each group called a phase, and there is an equal number of electromagnets per group. The number of groups is chosen by the designer of the stepper motor. The electromagnets of each group are interleaved with the electromagnets of other groups to form a uniform pattern of arrangement. For example, if the stepper motor has two groups identified as A or B, and ten electromagnets in total, then the grouping pattern would be ABABABABAB.



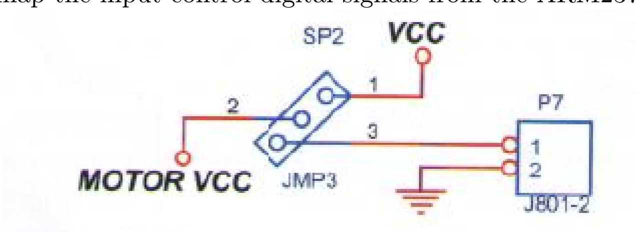
Electromagnets within the same group are all energized together. Because of this, stepper motors with more phases typically have more wires (or leads) to control the motor.

3.3.1 Used Port lines

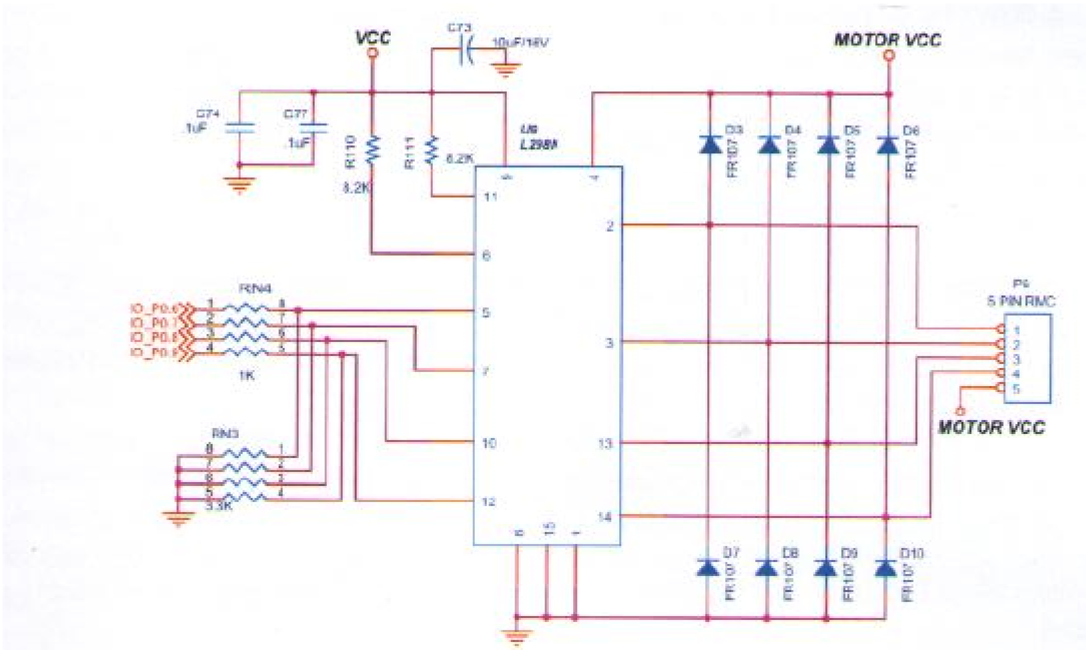
P3.24 & P3.26

3.3.2 Stepper Motor Driver in ViARM-2378

In the ViARM-2378 development board, there are many peripherals and stepper motor driver is one of them. The stepper motor driver itself is a peripheral interface controller (PIC - as mentioned in the class). The connection diagram for the IC (which is incorporated in ViARM-2378 development board) is given below. One of the main function of the driver is to map the input control digital signals from the ARM2378 into controlling power signal to the motor.



3.3.3 Jumper Position



Closed 1 & 2 - Internal voltage for stepper motor
Closed 2 and 3 - External voltage for stepper motor

3.4 Demo Programs

These programs are presented here to enable the students write their own program independently, given an engineering problem of similar type.

3.4.1 DIP Switch & LEDs

<generate .png file of the .c file>

3.4.2 Stepper Motor Control

The demo program (StpprMtrCntrl.c) is uploaded in the moodle, controls the stepper motor (angle to which it has to be turned etc).

```
/* 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;
        while(1)
        { IOPIN0=0X00000280;
            delay();
            IOPIN0=0X00000180;
            delay();
            IOPIN0=0X00000140;
            delay();
            IOPIN0=0X00000240;
            delay();
        }
```

```
        return 0;
}
```

4 Problem Definitions

Following are the tasks you need to take up for this lab session:

4.1 Tasks Involving LED

4.1.1 Task 1: Only LED

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

```
#include "LPC23xx.h"
int main ()
{
    while(1)
    {
        FIO3DIR=0x--; // LEDs are connected to lower 8 bits of
                      // Fast IO port 3 (FIO3PIN). Hence set the lower 8 bits
                      // as output in the corresponding direction
                      // register (FIO3DIR)
        FIO3PIN=0x--; Port to which actual data is to be written
        delay_code ; to be written
        FIO3PIN=0x--
        delay_code; to be written
    }
    return 0;
}
```

4.1.2 Task 2: DIP Switch and LED

Read the DIP positions and display the byte in LED ('0' by OFF & '1' for 'ON')

4.1.3 Task 3: Sum

Write a C program to read a DIP switch, split into two nibbles (4 bits), display the sum in the LEDs.

4.1.4 Task 4: Product

Write a C program to read a DIP switch, split into two nibbles (4 bits), display the sum in the LEDs.

4.2 Tasks Involving Stepper Motor

1. Modify the demo code (StpprMtrCntrl.c) supplied to demonstrate the control of stepper motor to rotate in opposite direction.
2. Gauge the speed of the stepper motor
3. What is the max speed with which the stepper motor can rotate?
4. How many steps are required for 360 deg rotation

5 Procedure

Stepper Control

1. Write a C program, which could control the stepper motor
2. Edit the above program file in Keil software. In Keil software one can edit, recompile and run etc. Compile it in Keil platform
3. Connect the serial cable from ARM LPC2148 and PC
4. Dump it in ARM LPC2148
5. Connect the stepper motor [Lab staffs / TAs would have done all these including jumper wire settings, so that you need not worry about these].
6. Run the program on the development kit by resetting the ViARM-2378.

For emulation you only need to do the following:

1. Write a C program, which could control the stepper motor
2. Edit the above program file in Keil software. In Keil software one can edit, recompile and run etc. Compile it in Keil platform
3. Identify the (digital) signal and demonstrate it to the TA

6 Results

1. Run the program and ask the TA to see the output
2. Take a snapshot using your mobile, write a report and submit