NANYANG POLYTECHNIC
EGR204 Microcontroller Applications

Laboratory Session 4

Course: Diploma in Robotics and Mechatronics

Module: EGR204 Microcontroller Applications

Experiment: 4

Title: Programming The 8051 Parallel Ports

Objective:

- The students will learn how to write 'C' program for the 8051 microcontroller to control the 8051 parallel port in order to display numbers on a 4-digit 7-segment LED display panel.

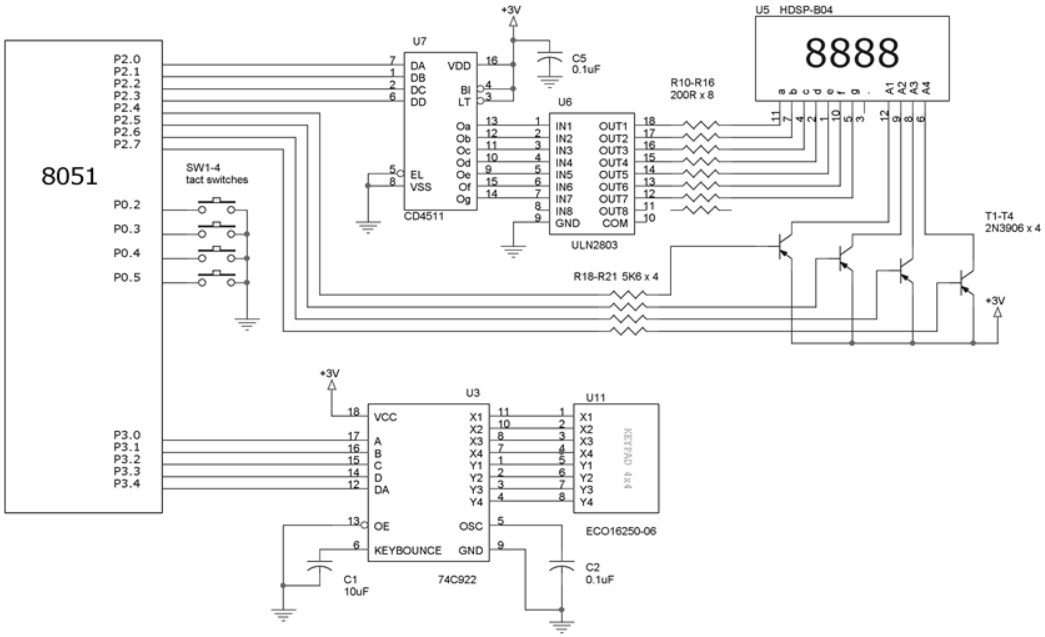
Learning Objectives:

- Recall from lab 3 how to write 'C' program to control the 4-digit 7-segment LED.
- Understand the logic of the routines mux_display( ) and display( ).
- Apply the routines mux_display( ) and display( ) to write 'C' program to display numbers on the 4-digit 7-segment LED.
- Write a stopwatch-like* program using 'C' (* *time accuracy is not important*).

1. Introduction

The 8051 port 2 (P2) is connected together with the CD4511 BCD-to-7 segment decoder to provide multiplexing operation to the 4-digit 7-segment LED. Figure 1-1 shows the circuit.

Figure 1-1
Multiplexing Circuit



2.1 Exercise 1: Using The mux_display( ) Subroutine

The program in listing 2-1 makes use of the mux_display( ) subroutine to display 4 digits on the 4-digit 7-segment LED. When the program is executed, the number "1", "2", "3" and "4" is display.

Analyse the mux_display( ) subroutine and make sure you understood it thoroughly.

Listing 2-1

```
#include <f200.h>

void delay(unsigned long duration)
{
   while((duration--)!=0);
}

void setSystem();

void mux_display(a,b,c,d)
unsigned char a,b,c,d;
{
   P2 = a | 0xE0;
   delay(100);
   P2 = b | 0xD0;
   delay(100);
   P2 = c | 0xB0;
   delay(100);
   P2 = d | 0x70;
   delay(100);
}

void main()
{
   setSystem();
    for (;;)
      mux_display(1,2,3,4);
}
```

2.2 Exercise 2: The display( ) Subroutine

Add the display( ) subroutine in listing 2-3 to the program in listing 2-1 and change the main routine in listing 2-1 to the one shown in listing 2-3.

Execute the program. What do you observe?

Listing 2-3

```
void display(unsigned int number)
{
   unsigned char a,b,c,d;

   a = number/1000;
   b = (number%1000)/100;
   c = (number%100)/10;
   d = (number%10);
   mux_display(a,b,c,d);
}

void main( )
{
   unsigned int k = 3468;
   setSystem();
   for(;;)
   {
      display(k);
   }
}
```

3.1 Assignment 1: Running Number

Make use to the display( ) subroutine to run a program so that the number on the 4-digit 7-segment LED will run from "0000" to "9999" and then repeat itself.

3.2 Assignment 2: Stopwatch-like timer

Complete the program in listing 3-1 to run in the following manners:

1. Display "0000" upon execution.
2. When START button is pressed, the display starts to run. If it hit "9999", it will repeat from "0000"
3. When STOP button is pressed, the display stop running.
4. When the RESET button is pressed, the display reset to "0000".

Note: There are four set of push buttons on your target board which are connected to P0. Refer to figure 1-1.

Listing 3-1

```
#define STOP_MODE   0
#define RUN_MODE    1

void main()
{
   unsigned char mode=STOP_MODE;
   unsigned int x=0;
   setSystem();

   for (;;)
   {
      if (P02==0) mode = RUN_MODE;
      if (P03==0) mode = STOP_MODE;
      if (P04==0) x = 0;

      if (mode==RUN_MODE)
      {
         // Increase x by 1.
         // Reset x to 0 if exceed 9999.
      }
      display(x);
   }
}
```

4. Program Analysis

Analyse the program in listing 4-1. What do you think the program do?

Note:
Program is incomplete. Recommended that you fill in all the necessary parts to complete the program, test and verify your analysis.

Listing 4-1

```
void main( )
{
   unsigned int number=0;
   unsigned char countdown=0;

   for(;;)
   {
      display(number);
      if (P37==0)
         number++;

      if (P36==0) && (number!=0)
         countdown = 1;

      if (countdown==1)
      {
         number--;
         if (number==0)
            countdown = 0;
      }
   }
}
```