NANYANG POLYTECHNIC
EGR204 Microcontroller Applications

Laboratory Session 9

Course:	Diploma in Robotics and Mechatronics
Module:	EGR204 Microcontroller Applications
Experiment:	9
Title:	Using the 8051 Interrupts

Objective:

- ❑ The students will learn how to write external and timer Interrupt for the 8051.

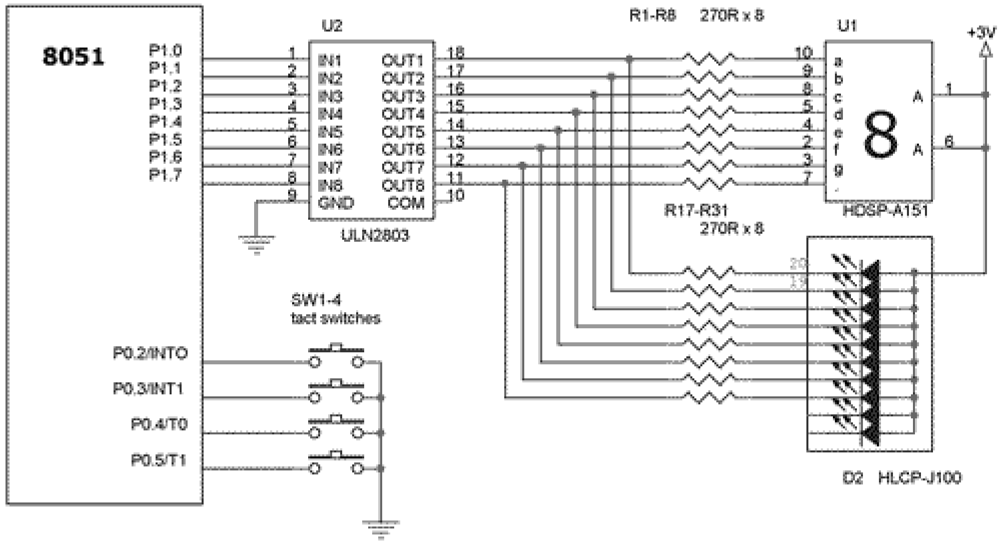
Learning Objectives:

- ❑ Recognise interrupt routines.
- ❑ Learn how to write the external and timer interrupt routines.

1. Introduction

Figure 1-1 shows how the two external interrupts can be connected to generate interrupts. The timer interrupts are internal interrupts and no hardware connection is needed.

Figure 1-1
External Interrupts Connection



2.1 Exercise 1: Blinking LED program.

Run the program in listing 2-1. Whenever the push button SW-1 (which is connected to P02) is pressed, the LEDs connected to Port 0 blinks. This happen because the program detects for P37 turning “0” through the instruction shown below:

```
if (P02==0)
    blinking_led();
```

Listing 2-1

```
#include <f200.h>

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

void setSystem();

void blinking_led()
{
   unsigned char x;

   for (x=0;x<20;x++)
   {
      P1 = ~P1;
      delay(10000);
   }
}

void main()
{
   setSystem();

   P1 = 0;     // Turn all LEDs off.

   for (;;)
   {
      if (P02==0)
         blinking_led();
   }
}
```

2.2 Exercise 2: Blinking LED using Interrupt

Run the program in listing 2-2. This is similar to program listing 2-1 where the SW-2 is pressed.

SW-2 is connected to P03 which is also the *external interrupt 1* input. SW-2 is a connected to P03 which is used as an simple I/O.

Do you notice any P03 detection program such as:

Listing 2-2

```
#include <f200.h>

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

void setSystem();

void blinking_led (void) interrupt 2
{
   unsigned char x;

   for (x=0;x<20;x++)
   {
      P1 = ~P1;
```

```
    if (P03==0)
        blink_led();
```

No? Then why the LEDs blink when SW-2 (P03) is toggle?

Note the declaration for the blinking_led() routine:

```
    void blinking_led (void) interrupt 2
```

blinking_led() is an interrupt routine. This routine will execute when an interrupt 2 occurs. Interrupt 2 is defined by the C compiler for external interrupt 1 (refer to your lecture notes).

Before you can use the interrupts, you have to set up the interrupts as follow:

```
  EA = 1;
  EX1 = 1;
```

```
      delay(10000);
   }
}

void main()
{
   setSystem();

   EA = 1;
   EX1 = 1;
   P1 = 0;     // Turn all LEDs off.

   for (;;);
}
```

2.3 Exercise 3: Timer Interrupt

Run the program in listing 2-3. You will observed that the LEDs start to blink. Time the blinking of the LEDs. At what rate is the LED blinking?

Look at the main program. Where can you find the 'C' instruction that makes the LEDs (which is connected to Port 0) blinks? However, there are instructions in the main program that setup timer 0 and start it running.

The timekeeper() routine is a timer 0 interrupt routine as defined by <u>interrupt 1</u> (check your notes). This routine will be executed whenever timer 0 overflows, that is, the TF0 flag is set. How often does timer 0 overflow?

Take note of the interrupt setup:

```
  EA = 1;
  ET0 = 1;
```

Listing 2-3

```
#include <f200.h>

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

void setSystem();

unsigned char count=0;

void timekeeper (void) interrupt 1
{
   TF0=0;
   TH0=0xDB;
   TL0=0xFF;
   count++;
   if (count==50)
   {
      count=0;
      P1 = ~P1;
   }
}

void main()
{
    setSystem();
    EA=1;
    ET0=1;
    TMOD=0x01;
    TR0=1;

    P1=0;
    for(;;);
}
```

3.1 Assignment 1: Using External Interrupt 0

There are two external interrupts for your 8051: 0 and 1. Switch SW1 is connected to Int0 and SW2 is connected to Int1. Modify program listing 2-2 so that the program will respond to an interrupt 0 (means that when you press SW1) instead of interrupt 1.

3.2 Assignment 2: Using Two External Interrupts Together

Continue from where you left off at assignment 1. Modify the program so that both external interrupt 0 and 1 (means that you can either press SW1 or SW2) will cause the LEDs to blink.

3.3 Assignment 3: Running a 1-sec Timer Display

Make use of the 4-digit LED display you have learned in earlier lab to display a running number at a rate of **1-sec**. Use the timer interrupt routine to keep track of the time.

The advantage of using the timer interrupt routine to take care of the time keeping and display is that the main program is free to perform other functions.

Listing 3-3

```
#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;
{
   …..
}

void display(unsigned int number)
{
   …..
}
void timeKeeper (void) interrupt 1
{
   TF0=0;
   TH0=0xDB;       // 10 ms overflow rate
   TL0=0xFF;

   // Complete the rest of the program here.
}
```

```
void main()
{
   setSystem();
   // Enable your timer 0 Interrupt.

   TMOD=0x01;
   TR0=1;          // Run timer 0.

   for(;;);
}
```