

Micro-processor Systems

LAB-5

Group No.: 4 Name: 石楚城 Id: 411085025
Date: 2024/11/29

Objectives: To familiarize the LY-51S development board, the 8051 I/O ports, and the C programming language.

References: Mazidi and McKinlay, “The 8051 Microcontroller……,” Chapter 4~7

Materials: The LY-51S development board and Keil μ Vision.

Grading policy:

For submission before the deadline: $(score) = (correctness\ score) \cdot$

$(timing\ ratio)$, where $(timing\ ratio) = 1.0 - \frac{0.2}{7} \times$

$(days\ after\ the\ lab\ announced)$.

After the submission deadline, 30 points will be deducted every day.

Complete the following activities and save the results as a PDF file. **Compress the report PDF file with the project directories (including all the files within) of the activities into a single ZIP file. Upload the ZIP file to Digital Academy 3.0 to submit the lab assignment.** Each group only needs to submit one report. Reports with plagiarized content will receive zero scores.

我有開一個 [Github Repo](#)，如果要測試我寫的 code，可以在上面複製下來 link: [Github repo](#)

Activity 1: (60%)

Using the LY-51S development board, connect P1 to the eight LEDs (J9 socket on the LY-51S board) and P2 to the eight independent buttons K1~K8 (J26 socket on the LY-51S board). Write a C program to perform the following functions and download it to the development board for execution.

- When the program runs, all LEDs on P1 should be off and remain off if no button is pressed.
- The program should continuously read the status of the independent buttons K1~K8 and convert the pressed button's number to its ASCII code, lighting up the corresponding LEDs. For example, pressing the K1 switch should display '00110001' on the LEDs on P1, where '1' means on and '0' means off;

similarly, pressing the K2 switch and the LED should display '00110010'; and so on, pressing the K8 switch should display '00111000'.

(Note : 因為 LED 是共陽極，記得輸出要輸入 0 才會亮)

- c. Button K8 has the highest priority, followed by K7, and so on. The program should prioritize using rotation or bitwise logic operations to determine the button status, avoiding multiple if-else or switch statements as much as possible.

Paste the complete program here, **including comments with the group number and member student IDs.**

C Program :

```
1  #include <reg51.h>
2
3  // 411085025
4  // group 4
5
6  #define INPUT P2
7  #define OUTPUT P1
8
9  extern unsigned char _listen(void);
10 void displayAsciiN(unsigned char);
11 void displayNone();
12 unsigned char numToAscii(unsigned char);
13 void MSDelay(unsigned int);
14
15 void main(void)
16 {
17     INPUT = 0xFF;
18     displayNone();
19     for(;;)
20     {
21         unsigned char x;
22         x = _listen();
23         if(x == 0)
24         {
25             displayNone();
26         }
27         else
28         {
29             displayAsciiN(x);
30         }
31     }
32 }
33
```

```

34 void displayAsciiN(unsigned char target)
35 {
36     OUTPUT = ~numToAscii(target);
37     MSDelay(1000);
38     displayNone();
39 }
40
41 unsigned char numToAscii(unsigned char target)
42 {
43     return 0x30 | target;
44 }
45
46 void displayNone()
47 {
48     OUTPUT = 0xFF;
49 }
50
51 void MSDelay(unsigned int itime)
52 {
53     unsigned int i, j;
54     for(i=0;i<itime;i++)
55     for(j=0;j<113;j++);
56 }

```

Assembly Subroutine :

```

1  ?PR?_listen?listenINPUT SEGMENT CODE
2      PUBLIC _listen
3
4      RSEG ?PR?_listen?listenINPUT
5  _listen:
6      MOV     R7, #8
7      SETB    C
8      MOV     A, P2
9  LOOP:   RLC     A
10         JNC     RETURN
11         DJNZ    R7, LOOP
12 RETURN: RET ; R7 is return value
13 END

```

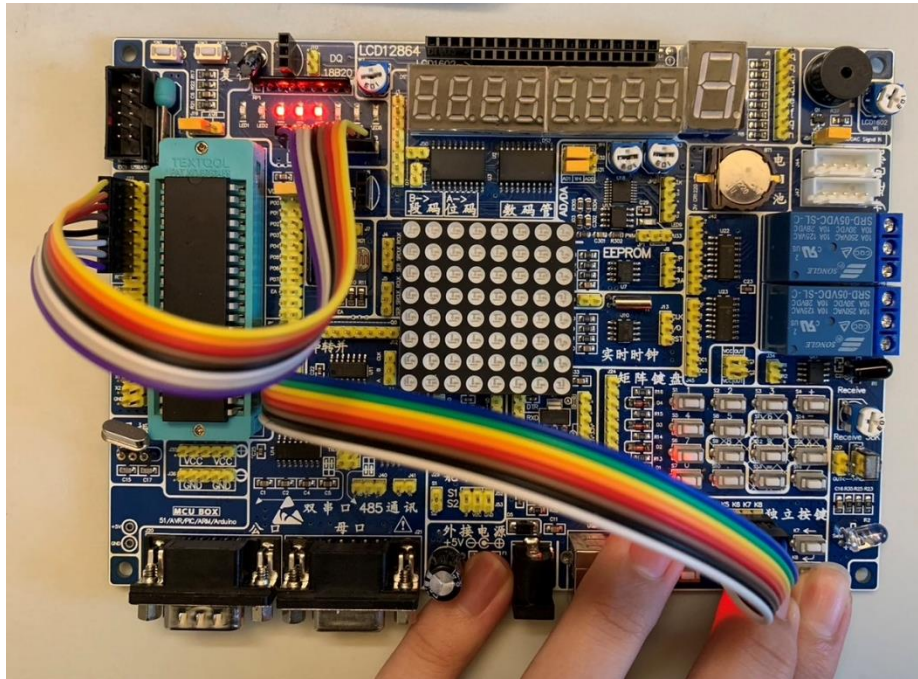
File Structure :

```

Project: lab_5_activity_1
├── Target 1
│   └── Source Group 1
│       ├── lab_5_activity_1.c
│       ├── reg51.h
│       └── listenINPUT.a51

```

After pressing any independent button, please take a photo of the eight LEDs' status and paste it here.



(同時按 K6、K8，K8 優先亮)

Activity 2: (40%)

Using the LY-51S development board, connect P1 to the single-digit 7-segment LED display (J6 socket on the board) and P2 to the eight independent buttons K1~K8 (J26 socket on the board). Write a C program to perform the following functions and download it to the development board for execution.

- When the program runs, all segments of the LED display on P1 should be off.
- After pressing the Kn button, the 7-segment LED display on port 1 should light up and show the number from 0 to n sequentially according to a one-second count. For example, pressing button K2 should display $0 \rightarrow 1 \rightarrow 2$ on the 7-segment LED display; pressing button K8 should display $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 7 \rightarrow 8$.
- After displaying all the numbers, return to the state described in part a and continue waiting and checking the buttons.
- Button K8 has the highest priority, followed by K7, and so on. The program should prioritize using rotation or bitwise logic operations to determine the button status, avoiding multiple if-else or switch statements as much as possible.

Paste the complete program here, **including comments with the group number and member student IDs.**

```

1  #include <reg51.h>
2
3  // 411085025
4  // group 4
5
6  #define INPUT P2
7  #define OUTPUT P1
8
9  void displayNToZero(unsigned char);
10 void displayN(unsigned char);
11 void displayNone();
12 void MSDelay(unsigned int);
13
14 void main(void)
15 {
16     INPUT = 0xFF;
17     for(;;)
18     {
19         unsigned char loop = 8;
20         unsigned char enable_bit = 0x80;
21         while(loop > 0)
22         {
23             if((~INPUT) & enable_bit)
24             {
25                 displayNToZero(loop);
26                 break;
27             }
28             enable_bit = enable_bit >> 1;
29             loop--;
30         }
31     }
32 }
33
34 void displayNToZero(unsigned char target)
35 {
36     unsigned char count = 0;
37     while(count <= target)
38     {
39         displayN(count);
40         MSDelay(1000);
41         count++;
42     }
43     displayNone();
44 }
45
46 void displayN(char target)
47 {
48     switch(target)
49     {
50         case 0:
51         {
52             OUTPUT = 0xC0;
53             break;
54         }
55         case 1:
56         {
57             OUTPUT = 0xF9;
58             break;
59         }
60         case 2:
61         {
62             OUTPUT = 0xA4;
63             break;
64         }
65         case 3:
66         {
67             OUTPUT = 0xB0;
68             break;
69         }

```

```

70     case 4:
71     {
72         OUTPUT = 0x99;
73         break;
74     }
75     case 5:
76     {
77         OUTPUT = 0x92;
78         break;
79     }
80     case 6:
81     {
82         OUTPUT = 0x82;
83         break;
84     }
85     case 7:
86     {
87         OUTPUT = 0xD8;
88         break;
89     }
90     case 8:
91     {
92         OUTPUT = 0x80;
93         break;
94     }
95     case 9:
96     {
97         OUTPUT = 0x90;
98         break;
99     }
100 }
101 }
102
103 void displayNone()
104 {
105     OUTPUT = 0xFF;
106 }
107
108 void MSDelay(unsigned int itime)
109 {
110     unsigned int i, j;
111     for(i=0;i<itime;i++)
112         for(j=0;j<113;j++);
113 }

```

After pressing any independent button, please take a photo of the single-digit 7-segment LED display and paste it here.

