微算機實驗報告

Lab #11

姓名：**溫環華**

系級：**電機系**

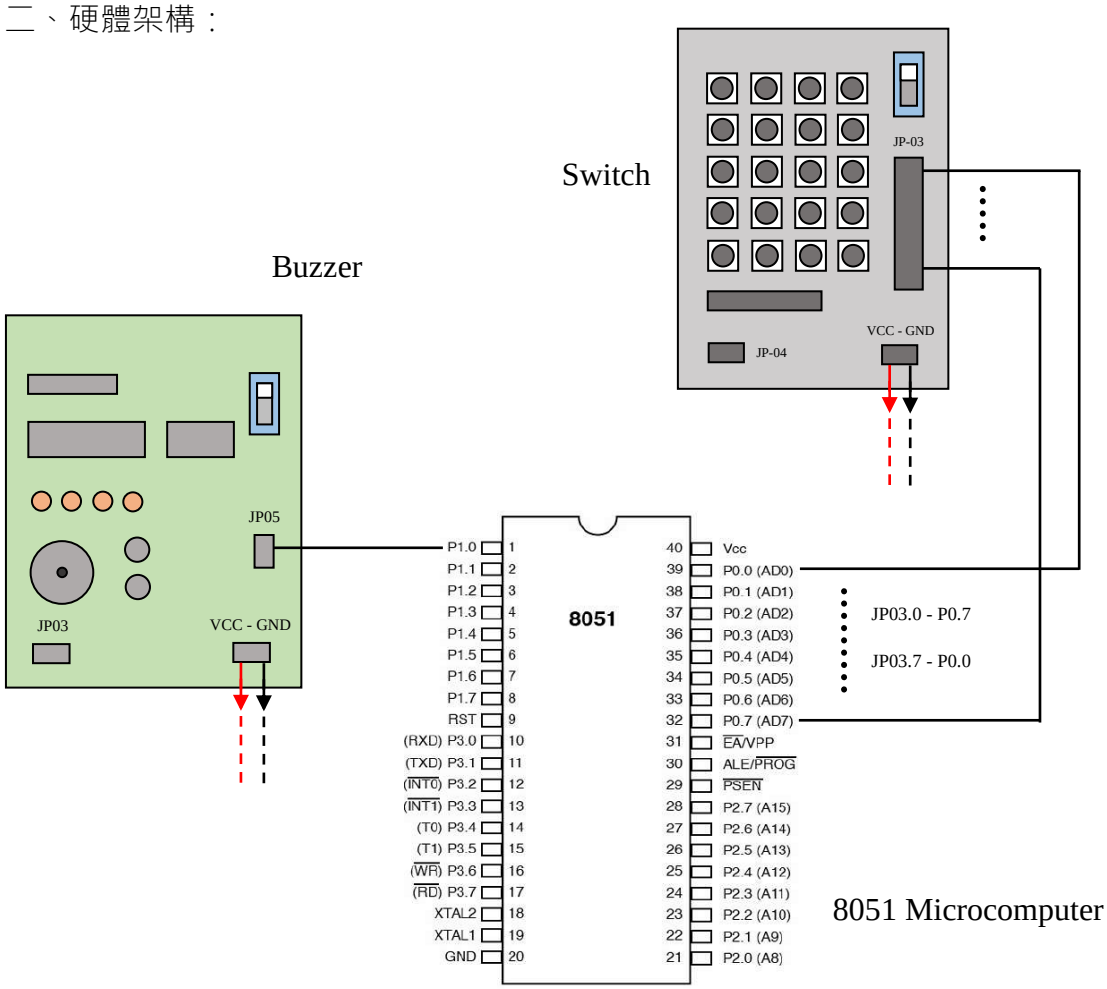
學號：**0710872**

上課時間：**2021-12-14**

一、實驗目的：

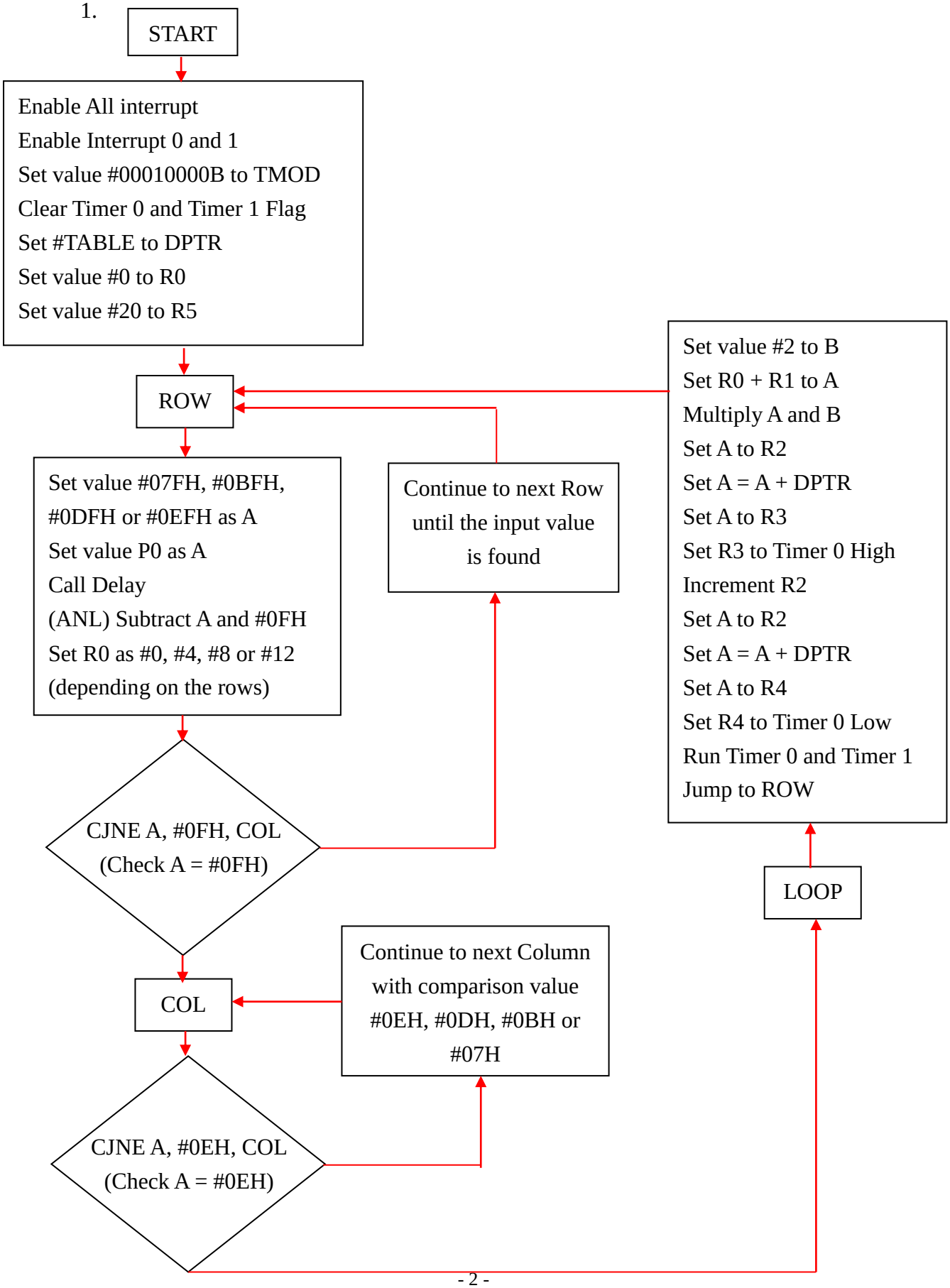
We learned the principle of the buzzer, the calculation method of the scale frequency and use the internal TIMER interrupt to control the square wave output and emit different scales.

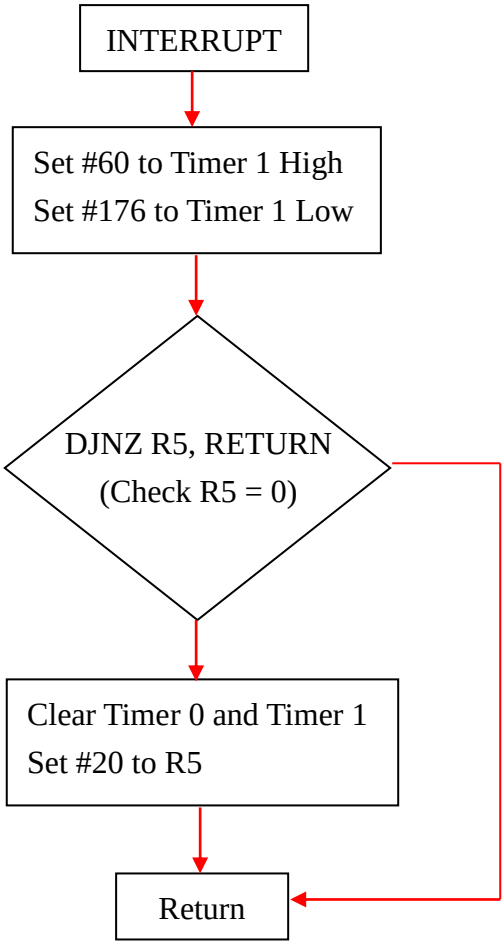
二、硬體架構：

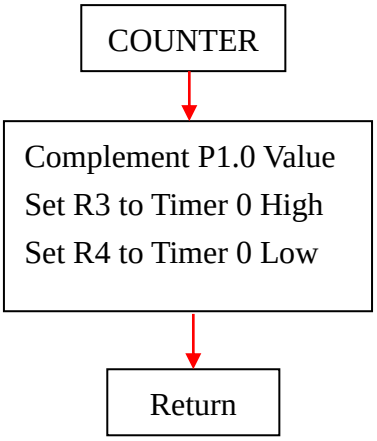


三、程式流程圖：

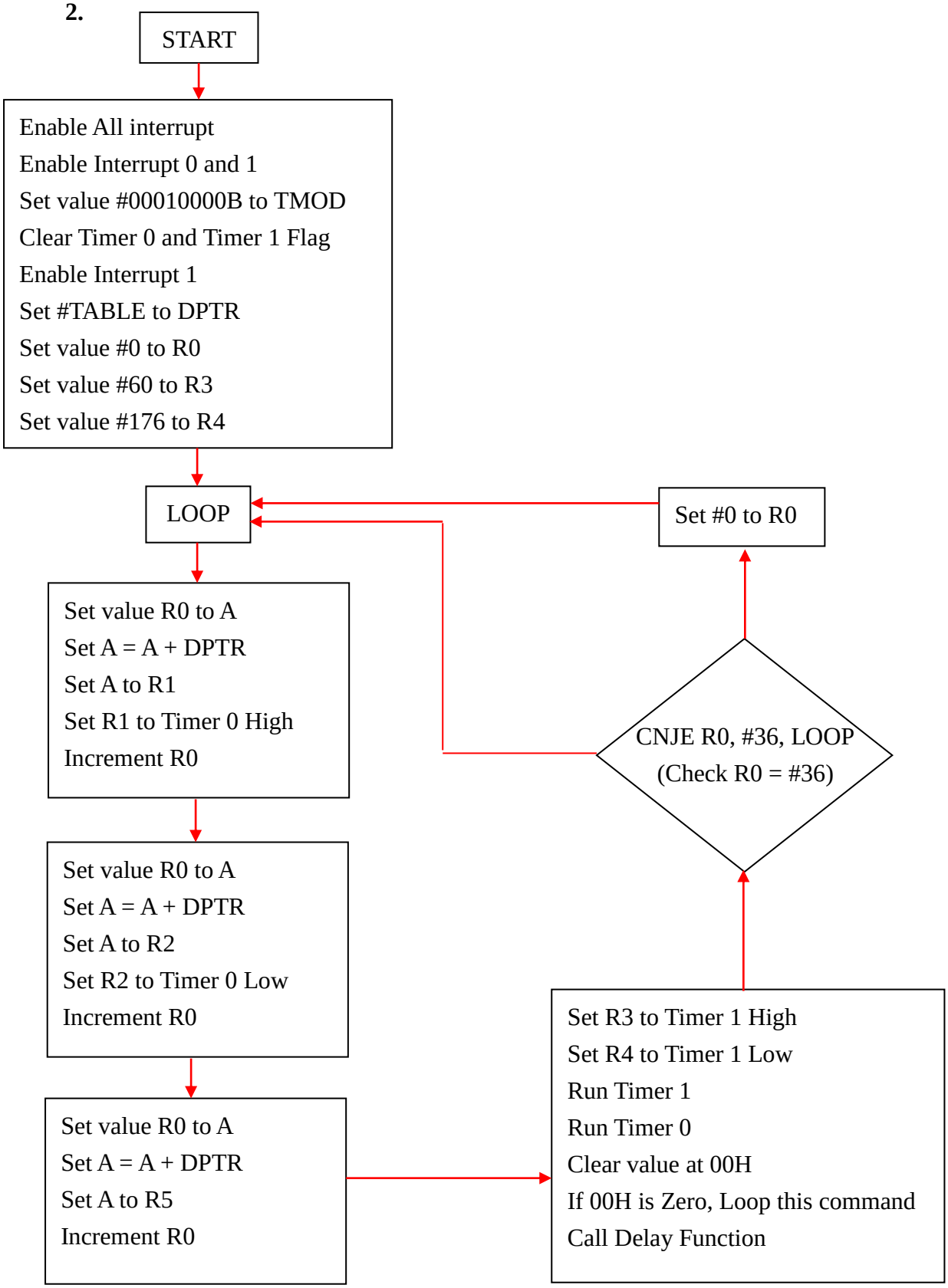
1.

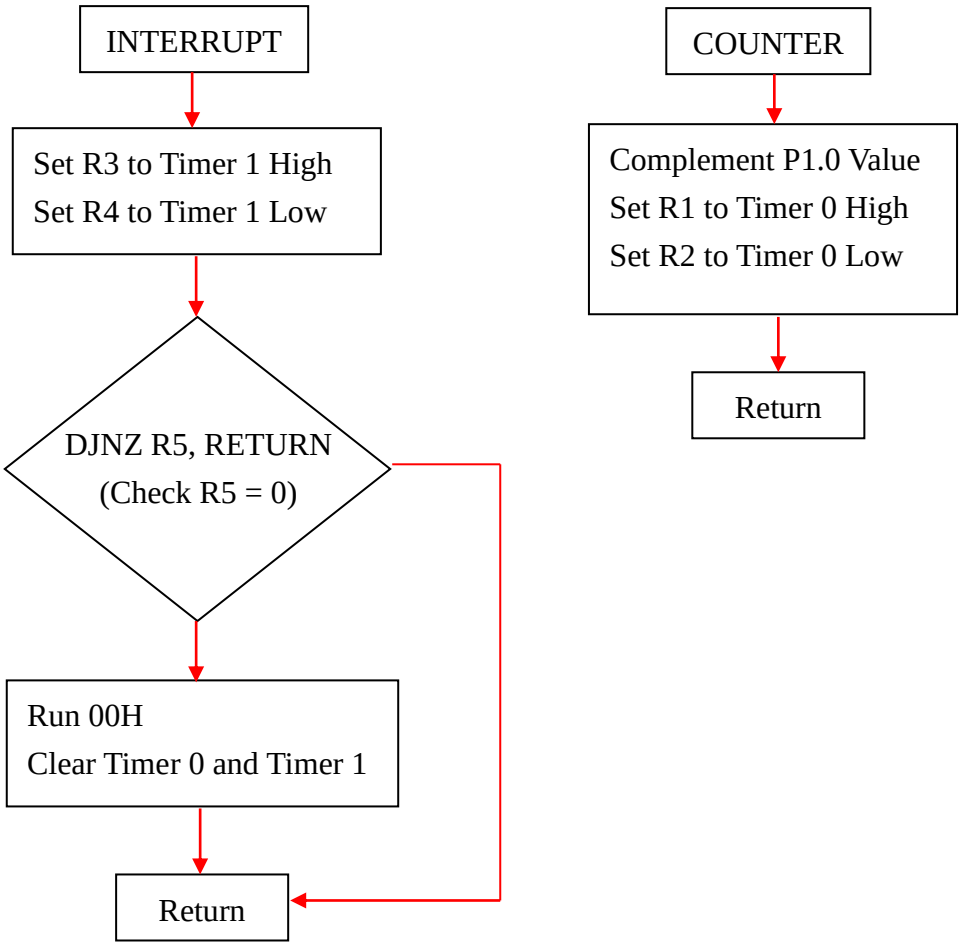

START
Enable All interrupt
Enable Interrupt 0 and 1
Set value #00010000B to TMOD
Clear Timer 0 and Timer 1 Flag
Set #TABLE to DPTR
Set value #0 to R0
Set value #20 to R5
ROW
Set value #07FH, #0BFH, #0DFH or #0EFH as A
Set value P0 as A
Call Delay
(ANL) Subtract A and #0FH
Set R0 as #0, #4, #8 or #12 (depending on the rows)
CJNE A, #0FH, COL (Check A = #0FH)
Continue to next Row until the input value is found
Set value #2 to B
Set R0 + R1 to A
Multiply A and B
Set A to R2
Set A = A + DPTR
Set A to R3
Set R3 to Timer 0 High
Increment R2
Set A to R2
Set A = A + DPTR
Set A to R4
Set R4 to Timer 0 Low
Run Timer 0 and Timer 1
Jump to ROW
LOOP
COL
Continue to next Column with comparison value #0EH, #0DH, #0BH or #07H
CJNE A, #0EH, COL (Check A = #0EH)


INTERRUPT
Set #60 to Timer 1 High
Set #176 to Timer 1 Low
DJNZ R5, RETURN
(Check R5 = 0)
Clear Timer 0 and Timer 1
Set #20 to R5
Return


COUNTER
Complement P1.0 Value
Set R3 to Timer 0 High
Set R4 to Timer 0 Low
Return

2.


START
Enable All interrupt
Enable Interrupt 0 and 1
Set value #00010000B to TMOD
Clear Timer 0 and Timer 1 Flag
Enable Interrupt 1
Set #TABLE to DPTR
Set value #0 to R0
Set value #60 to R3
Set value #176 to R4
LOOP
Set value R0 to A
Set A = A + DPTR
Set A to R1
Set R1 to Timer 0 High
Increment R0
Set value R0 to A
Set A = A + DPTR
Set A to R2
Set R2 to Timer 0 Low
Increment R0
Set value R0 to A
Set A = A + DPTR
Set A to R5
Increment R0
Set R3 to Timer 1 High
Set R4 to Timer 1 Low
Run Timer 1
Run Timer 0
Clear value at 00H
If 00H is Zero, Loop this command
Call Delay Function
CNJE R0, #36, LOOP
(Check R0 = #36)
Set #0 to R0


INTERRUPT
Set R3 to Timer 1 High
Set R4 to Timer 1 Low
DJNZ R5, RETURN
(Check R5 = 0)
Run 00H
Clear Timer 0 and Timer 1
Return
COUNTER
Complement P1.0 Value
Set R1 to Timer 0 High
Set R2 to Timer 0 Low
Return

四、問題與討論：

1. 試根據你所設計的程式架構，解說如何實現蜂鳴器的長短音功能。

   By using 2 timers which is timer 0 and Timer 1 we can control the length of the tone, the longer tone we can assign it to Timer 1 using the interrupt function. The value of Timer 1 must be longer than Timer 0.

2. 假設要更改音色，試問要如何更改?

   We can change the tone's frequency by changing the table value. The value of the table is calculated by the frequency and the clock of the microcomputer.

五、程式碼與註解：

1.

```
ORG 0000H               // Open 0000H address
JMP START               // Jump to START
ORG 00BH                // Open 000BH address
JMP COUNTER             // Jump to COUNTER
ORG 001BH               // Open 001BH address
JMP INTERRUPT           // Jump to INTERRUPT
ORG 0050H               // Open 0050H address

START:
    SETB EA             // Enable All interrupt
    SETB ET0            // Enable Interrupt 0
    SETB ET1            // Enable Interrupt 1
    MOV TMOD,#00010000B // Set value #00010000B to TMOD
    CLR TF0             // Clear Timer 0 Flag
    CLR TF1             // Clear Timer 1 Flag
    MOV DPTR,#TABLE     // Set #TABLE to DPTR
    MOV R0,#0           // Set value #0 to R0
    MOV R5,#20          // Set value #20 to R5

ROW1:
    MOV P0,#07FH        // Set value #07FH to P0
    CALL DELAY          // Call delay function
    MOV A,P0            // Set value P0 to A
    ANL A,#0FH          // Subtract value in A by #0FH
    MOV R0,#0           // Set value #0 to R0
    CJNE A,#0FH,COL1    // Compare if A = #0FH, jmp to COL1 if false, Continue if true
```

```
ROW2:
    MOV P0,#0BFH          // Set value #0BFH to P0
    CALL DELAY            // Call delay function
    MOV A,P0              // Set value P0 to A
    ANL A,#0FH            // Subtract value in A by #0FH
    MOV R0,#4             // Set value #4 to R0
    CJNE A,#0FH,COL1      // Compare if A = #0FH, jmp to COL1 if false, Continue if true

ROW3:
    MOV P0,#0DFH          // Set value #0DFH to P0
    CALL DELAY            // Call delay function
    MOV A,P0              // Set value P0 to A
    ANL A,#0FH            // Subtract value in A by #0FH
    MOV R0,#8             // Set value #8 to R0
    CJNE A,#0FH,COL1      // Compare if A = #0FH, jmp to COL1 if false, Continue if true

ROW4:
    MOV P0,#0EFH          // Set value #0EFH to P0
    CALL DELAY            // Call delay function
    MOV A,P0              // Set value P0 to A
    ANL A,#0FH            // Subtract value in A by #0FH
    MOV R0,#12            // Set value #12 to R0
    CJNE A,#0FH,COL1      // Compare if A = #0FH, jmp to COL1 if false, Continue if true
    JMP ROW1              // Jump to ROW1

COL1:
    CJNE A,#0EH,COL2      // Compare if A = #0EH, jmp to COL2 if false, Continue if true
    MOV R1,#0             // Set value #0 to R1
    JMP LOOP              // Jump to LOOP

COL2:
    CJNE A,#0DH,COL3      // Compare if A = #0DH, jmp to COL3 if false, Continue if true
    MOV R1,#1             // Set value #1 to R1
    JMP LOOP              // Jump to LOOP

COL3:
    CJNE A,#0BH,COL4      // Compare if A = #0BH, jmp to COL4 if false, Continue if true
    MOV R1,#2             // Set value #2 to R
    JMP LOOP              // Jump to LOOP

COL4:
    CJNE A,#07H,ROW1      // Compare if A = #07H, jmp to ROW1 if false, Continue if true
    MOV R1,#3             // Set value #3 to R1
    JMP LOOP              // Jump to LOOP

LOOP:
    MOV B,#2              // Set value #2 to B
    MOV A,R0              // Set R0 to A
    ADD A,R1              // Set R1 to A
    MUL AB                // Multiply A and B
    MOV R2,A              // Set A to R2
    MOVC A,@A+DPTR        // Set A = A + DPTR
    MOV R3,A              // Set A to R3
    MOV TH0,R3            // Set R3 to Timer 0 High
    INC R2                // Increment R2
    MOV A,R2              // Set A to R2
    MOVC A,@A+DPTR        // Set A = A + DPTR
    MOV R4,A              // Set A to R4
    MOV TL0,R4            // Set R4 to Timer 0 Low
    SETB TR0              // Run Timer 0
    SETB TR1              // Run Timer 1
    JMP ROW1              // Jump to ROW1

INTERRUPT:
    MOV TH1,#60           // Set #60 to Timer 1 High
    MOV TL1,#176          // Set #176 to Timer 1 Low
    DJNZ R5,RETURN        // Decrement R5, if zero jump to RETURN
    CLR TR0               // Clear Timer 0
    CLR TR1               // Clear Timer 1
    MOV R5,#20            // Set #20 to R5
```

```
RETURN:
    RETI                // Return

COUNTER:
    CPL P1.0            // Complement P1.0 Value
    MOV TH0,R3          // Set R3 to Timer 0 High
    MOV TL0,R4          // Set R4 to Timer 0 Low
    RETI                // Return

DELAY:
    MOV R6,#0FH
DELAY1:
    MOV R7,#01FH
DELAY2:
    DJNZ R7,DELAY2
    DJNZ R6,DELAY1
    RET

TABLE:
    DB 196,12
    DB 202,28
    DB 208,21
    DB 211,8
    DB 216,5
    DB 220,16
    DB 224,12
    DB 226,4

END
```

2.

```
ORG 0000H               // Open 0000H address
JMP START               // Jump to START
ORG 00BH                // Open 00BH address
JMP COUNTER             // Jump to COUNTER
ORG 001BH               // Open 001BH address
JMP INTERRUPT           // Jump to INTERRUPT
ORG 0050H               // Open 0050H address

START:
    SETB EA                 // Enable All interrupt
    SETB ET0                // Enable Interrupt 0
    MOV TMOD,#00010000B     // Set value #00010000B to TMOD
    CLR TF0                 // Clear Timer 0 Flag
    SETB ET1                // Enable Interrupt 1
    CLR TF1                 // Clear Timer 1 Flag
    MOV DPTR,#TABLE         // Set #TABLE to DPTR
    MOV R0,#0               // Set value #0 to R0
    MOV R3,#60              // Set value #60 to R3
    MOV R4,#176             // Set value #176 to R4

LOOP:
    MOV A,R0                // Set value R0 to A
    MOVC A,@A+DPTR          // Set A = A + DPTR
    MOV R1,A                // Set A to R1
    MOV TH0,R1              // Set R1 to Timer 0 High
    INC R0                  // Increment R0


    MOV A,R0                // Set value R0 to A
    MOVC A,@A+DPTR          // Set A = A + DPTR
    MOV R2,A                // Set A to R2
    MOV TL0,R2              // Set R5 to Timer 0 Low
    INC R0                  // Increment R0
```

```
    MOV A,R0                // Set value R0 to A
    MOVC A,@A+DPTR          // Set A = A + DPTR
    MOV R5,A                // Set A to R5
    INC R0                  // Increment R0

    MOV TH1,R3              // Set R3 to Timer 1 High
    MOV TL1,R4              // Set R4 to Timer 1 Low
    SETB TR1                // Run Timer 1

    SETB TR0                // Run Timer 0
    CLR 00H                 // Clear value at 00H

    JNB 00H,$               // if 00H is Zero, Loop this command
    ACALL DELAY             // Call Delay Function
    CJNE R0,#36,LOOP        // Check if R0 = #36. If true, continue. If false, jump to LOOP
    MOV R0,#0               // Set #0 to R0
    JMP LOOP                // Jump to LOOP

INTERRUPT:
    MOV TH1,R3              // Set R3 to Timer 1 High
    MOV TL1,R4              // Set R4 to Timer 1 Low
    DJNZ R5,RETURN          // Decrement R5, if zero jump to RETURN
    SETB 00H                // Run 00H
    CLR TR0                 // Clear Timer 0
    CLR TR1                 // Clear Timer 1

RETURN:
    RETI                    // Return

COUNTER:
    CPL P1.0                // Complement P1.0 Value
    MOV TH0,R1              // Set R1 to Timer 0 High
    MOV TL0,R2              // Set R2 to Timer 0 Low
    RETI                    // Return

DELAY:
    MOV R6,#07FH
DELAY1:
    MOV R7,#01FH
DELAY2:
    DJNZ R7,DELAY2
    DJNZ R6,DELAY1
    RET

TABLE:
    DB 220,16,5
    DB 211,8,5
    DB 196,12,5
    DB 211,8,5
    DB 216,5,5
    DB 226,4,10

    DB 216,5,5
    DB 220,16,5
    DB 216,5,5
    DB 196,12,5
    DB 211,8,5
    DB 211,8,10

END
```

六、心得：

1. We learned how to use the internal Timer as a delay. The internal structure of the Timer/Counter is also introduced. I feel that I can understand the teacher's explanation during the class. The next experiment should be no problem.

2. Timer/Counter has some settings to do, and I am not familiar with the whole system at the beginning, I am stuck for a long time, I don't know how to start, and the experiment last week has not been done yet. After reading the handout many times, I kept trying to write, and then asked the TAs to figure out the concept as much as possible, and finally I wrote it out.