

Question	T1	P1	P2	P3	Q1	Q2	Q3	Q4	Q5	Total	Name
Your score											St. ID
Full score	4	10	15	15	10	12	10	10	14	100	Seat

EE3046 Microcomputers Theory and Laboratory, **Spring Semester 2022**  
**Midterm Exam (8051)** Scope: edsim51 Time: 18:00~20:30, 2022.05.02

查看期中繳交記錄：<https://docs.google.com/spreadsheets/d/10iAwLlMrI9tBttWnMPN4ZYp0ZlUz8eiRm6inOR1NJg/edit?usp=sharing>

Your name will be shown right after you upload your answer to P1, P2, P3, or Q part (i.e., this WORD file).

T1: (4%)

The performance in uploading your answering files.

For example, if your filename doesn't conform to the requested format, the score (max. 4%) will be reduced.

**P1: (10%) ASCII code test**

This template program contains a subroutine called "lowercaseTest". The subroutine tests whether the accumulator contains the ASCII code of a lowercase letter (i.e., 'a', 'b', ..., or 'z'). Before return (i.e., RET), the subroutine sets the carry flag if it is a lowercase letter; otherwise, it clears the carry flag.

Download the template program at <http://www.ee.ncu.edu.tw/~jztsai/EE3046/mmmQQQQ/>

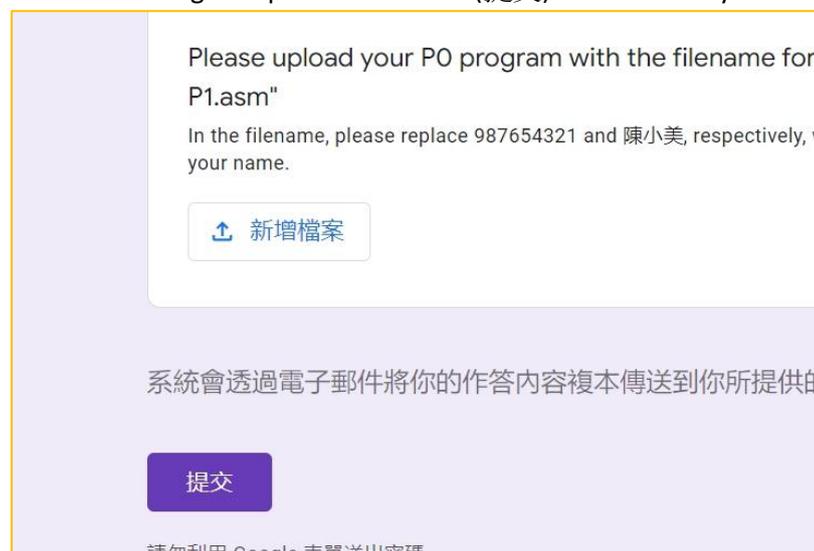
Note:

When you upload your program for P1, the file name must be " stdID yourName P1.asm ", for example, "987654321 陳小美 P1.asm ".

The link for uploading your program for P1 :

[https://docs.google.com/forms/d/e/1FAIpQLSfnb62gyeHGdAEN5qFeLYDqG8g5yX4f\\_8Qft5wmDg3DmTkktQ/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfnb62gyeHGdAEN5qFeLYDqG8g5yX4f_8Qft5wmDg3DmTkktQ/viewform?usp=sf_link)

Note: Don't forget to press the submit (提交) button after you have selected your files for uploading.



**P2: (20%) Sample and display**

In Example 6 of edsim51, the analog input voltage is sampled and displayed on the Scope every 50 μs. Now, the P2 program will display the voltage difference between the newly sampled input analog voltage and the input analog voltage sampled 250 μs ago. That is to say, the P2 program will display  $V_{in}(t) - V_{in}(t - 250 \mu s)$ , where t is a sampled time instance.

Note:

When you upload your program for P2, the file name must be " stdID yourName P2.asm ", for example,

“987654321 陳小美 P2.asm “.

The link for uploading your program for P2 :

[https://docs.google.com/forms/d/e/1FAIpQLSdqFuizzLEg\\_2c2C4f4bqF6QuXNkGWajZBcqI9sitzt1Uh4EA/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSdqFuizzLEg_2c2C4f4bqF6QuXNkGWajZBcqI9sitzt1Uh4EA/viewform?usp=sf_link)

### P3: (15%) Motor rotation

In this program, the motor rotates for a specified number of revolutions.

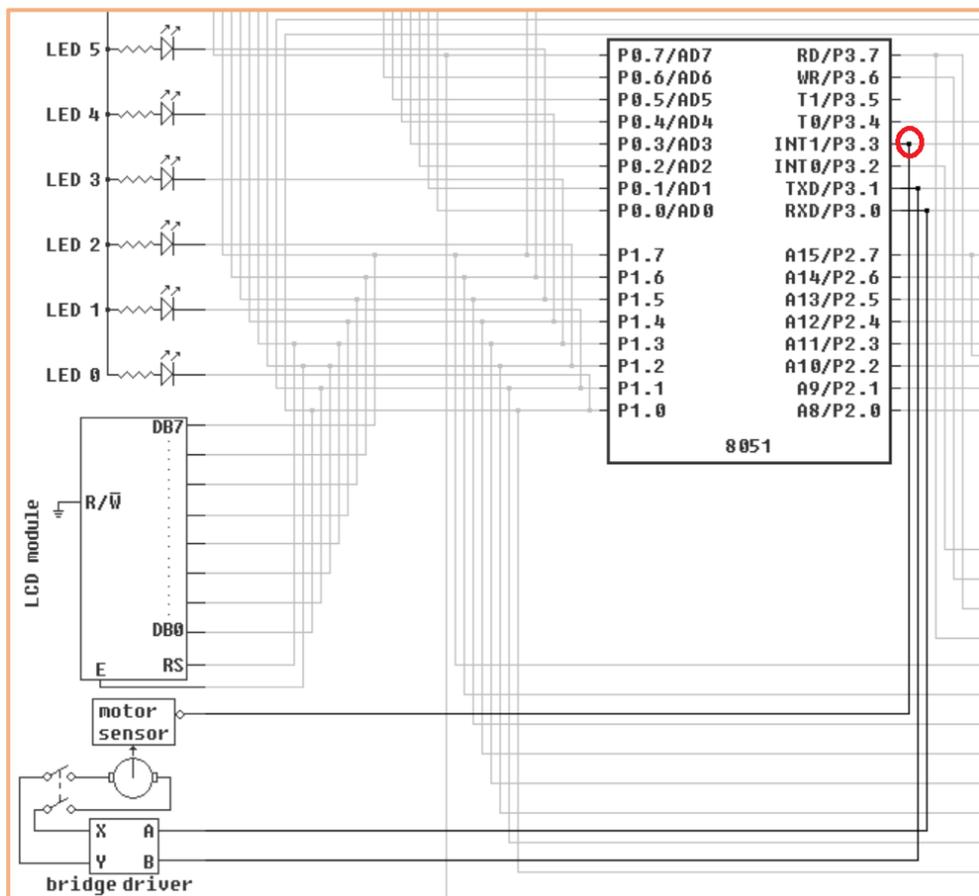
The function of this program is described in the following items:

- 1 Before the program starts, the register R7 stores a number that specifies the number of revolutions for motor rotation.
- 2 After the program starts, the motor starts to rotate for the specified number of revolutions.
- 3 During the time of motor rotation, the LED keeps showing the “decreasing pattern” continuously. The “decreasing pattern” is implemented as follows:

decreasingLED:

```
DEC R1
MOV P1,R1
JMP decreasingLED
```

- 4 The motor sensor must be connected to P3.3, as show in the figure below. That is to say, the external interrupt 1 must be used for counting the number of revolutions.



Note:

When you upload your program for P3, the file name must be " stdID yourName P3.asm ", for example, “987654321 陳小美 P3.asm “.

The link for uploading your program for P3 :

[https://docs.google.com/forms/d/e/1FAIpQLScDlx\\_TvjPH2zaU962AFOCEGIqBS78I2IYQ-8ZkI7zGPL9IqA/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLScDlx_TvjPH2zaU962AFOCEGIqBS78I2IYQ-8ZkI7zGPL9IqA/viewform?usp=sf_link)

Q1: (10%)

The following program is executed. The content of the data memory is observed while running a NOP instruction in the subroutine called **sub2**, and its snapshot is shown in the figure below. Indeed, the stack content is observed.

```
LJMP main
ORG ???H ; ← The number is masked intentionally.
main:
    MOV SP,#10H
    MOV R3,#0AFH
    MOV A,36H
    MOV 7FH,#0C8H
    PUSH 7FH
    LCALL sub1 ; Find the address of this instruction
    CPL C
    NOP
    NOP
    POP 7FH
    JMP $

ORG ???H ; ← The number is masked intentionally.
sub1:
    SUBB A,#33H
    LCALL sub2 ; Find the address of this instruction
    DA A
    RET

ORG ???H ; ← The number is masked intentionally.
sub2:
    PUSH AR3
    DEC R3
    NOP
    NOP ; Observe the data memory at this instance.
    NOP
    MOV 78H,R3
    POP AR3
    RET
```

addr	0x00	0x00	value
0	00	00	00 00
10	00	C8	52 62 D2 76 AF 00 00 00 00 00 00 00 00
20	00	00	00 00 00 00 00 00 00 00 00 00 00 00 00 00
30	00	00	00 00 00 00 00 00 00 00 00 00 00 00 00 00
40	00	00	00 00 00 00 00 00 00 00 00 00 00 00 00 00

Please find out the addresses of the two LCALL instructions.

(A) The address of LCALL sub1 =	624F H
(B) The address of LCALL sub2 =	76CF H

Q2: (12%) Branch destination

The following table shows an instruction sequence, the machine code of the branch instructions, and the code memory address where the machine code is stored.

In the "Destination" column, please write down the destination for each branch instruction.

For example, the branch instruction at LB66 is **LJMP LB11**. The destination for the long jump instruction is **LB11**.

Address	Machine code	Assembly Instruction	Destination
		<b>org 1234H</b>	
1234		LB11: INC A	
1235		LB12: MOV R3,AR5	
1237		LB13: DEC @R1	
1238		LB14: RL A	
1239		LB15: ADD A,R3	
		<b>org 2811H</b>	
2811		LB21: INC R2	
2812	02 3A 8C	LB22: LJMP LB??	<b>LB63 (只寫 63 也可)</b>
2815		LB23: MUL AB	
2816		LB24: DEC A	
2817		LB25: POP 20H	
		<b>org 3628H</b>	
3628		LB31: PUSH 70H	
362A		LB32: ANL C,65H	
362C		LB33: SWAP A	
362D	E1 4A	LB34: AJMP LB??	<b>LB54</b>
362F		LB35: RR A	
		<b>org 36CCH</b>	
36CC	80 05	LB41: SJMP LB??	<b>LB45</b>
36CE		LB42: INC R4	
36CF	40 75	LB43: JC LB??	<b>LB52</b>
36D1		LB44: ORL A,#98H	
36D3		LB45: SETB C	
		<b>org 3744H</b>	
3744		LB51: ANL A,#75H	
3746		LB52: DA A	
3747	B4 25 87	LB53: CJNE A,#25H,LB??	<b>LB44</b>
374A		LB54: XCH A,R6	
374B	C1 2A	LB55: AJMP LB??	<b>LB32</b>
		<b>org 3A88H</b>	
3A88		LB61: CPL 56H	
3A8A		LB62: CLR 76H	
3A8C		LB63: SUBB A,R7	
3A8D		LB64: RRC A	
3A8F	02 12 34	LB66: LJMP LB11	<b>LB11</b>

**Q3: (10%) Number system**

(A) Convert a signed binary number to a sign-magnitude binary number, and vice versa.

Sign-magnitude		Signed binary
1010	→	1...10 (10, 110, 1110, ...)
10...010011 (110011, 1010011, 10010011, ...)	←	11111101101

(B) You are given two signed binary numbers: 100110 and 01101.

Do the subtraction.

Express the result in 8-bit signed binary number.

Express the result in signed binary number with the minimal bit number.

		1	0	0	1	1	0
—			0	1	1	0	1
=	1	1	0	1	1	0	1
=		1	0	1	1	0	1

(Format: 8-bit signed binary number, 8-bit)

(Format: signed binary number, minimal bit number)

**Q4. (10%) Addressing mode**

The following are addressing modes for 8051 instructions :

①Immediate ②Register ③Direct ④Indirect ⑤Indexed ⑥Relative ⑦Absolute ⑧Long

Please determine the addressing modes for the underlined operands of the following instructions:

Addressing mode	Instruction
②	ADD A, <u>R4</u>
3	CJNE A, <u>65H</u> , LB12
4	MOV <u>@R1</u> , A
3	PUSH <u>Acc</u>
6	DJNZ <u>44H</u> , <u>-12</u>
5	MOV A, <u>@A+PC</u>



**Upload the WORD file containing your answers for Q1~Q5 to**

**[https://docs.google.com/forms/d/e/1FAIpQLSeoh1Hs49bEnvAQ53ZyZMYn9bedYmeKpYSYG7ZpRIDBFekvRA/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSeoh1Hs49bEnvAQ53ZyZMYn9bedYmeKpYSYG7ZpRIDBFekvRA/viewform?usp=sf_link)**