MICROCONTROLLER APPLICATONS /

ENGINEERING DESIGN & BUSINESS PROJECT II

2017/2018 SEMESTER TWO MID-SEMESTER TEST

SAS code:

MST

ET1010
Diploma in Aerospace Electronics (DASE)
Diploma in Energy Systems and Management (DESM)
Diploma in Computer Engineering (DCPE)
Diploma in Electrical & Electronic Engineering (DEEE)
Diploma in Mechatronics and Robotics (DMRO)

ET1216
Diploma in Engineering with Business (DEB)

2nd Year Full-Time

Time Allowed: 1.5 Hours

Instructions to Candidates

1. The Singapore Polytechnic examination rules are to be complied with.
2. This paper consists of TWO sections:

   Section A - 10 Multiple Choice Questions, 3 marks each.

   Section B - 5 Questions, 14 marks each.
3. ALL questions are COMPULSORY.
4. All questions are to be answered in the Answer Booklet. Start each question in Section B on a new page.
5. This paper consists of 10 pages (including 2 pages in the Appendix).

SECTION A

MULTIPLE CHOICE QUESTIONS [3 marks each]

- Please tick your answers in the MCQ box provided on the second page of the answer booklet.
- No marks will be deducted for wrong answers.

A1. Which one of the following shows the correct sequence of steps in programming a microcontroller?

◆ Write C program

◘ Create project

O Download hex file into microcontroller

★ Compile C program

(a) ◆ → O → ★ → ◘
(b) ◘ → ◆ → O → ★
(c) ◘ → O → ★ → ◆
(d) ◘ → ◆ → ★ → O

A2. A compiler _____

(a) converts an assembly language program into machine code.
(b) converts a high level language e.g. C-language program into machine code.
(c) allows a program to be simulated on a PC before it is run on a microcontroller.
(d) converts machine code into a high level language e.g. C-language program.

A3. Which of the codes below can be used to set RA3 (Port A, Bit 3) as an input pin?

(a) PORTAbits.RA3 = 1;
(b) PORTAbits.RA3 = 0;
(c) TRISAbits.TRISA3 = 1;
(d) TRISAbits.TRISA3 = 0;

A4. The code while (PORTAbits.RA3 == 0); _____

(a) waits for RA3 to become low i.e. logic 0, before executing the next instruction.
(b) waits for RA3 to become high i.e. logic 1, before executing the next instruction.
(c) writes a logic 0 to RA3.
(d) writes a logic 1 to RA3.

A5. Which line of code below can be used to write a logic 1 to RB5 (Port B, Bit 5), without affecting other bits of Port B?

(a) PORTB = 0xFF;
(b) PORTBbits.RB5 = 0;
(c) PORTB = 0b00100000;
(d) PORTB = PORTB | 0b00100000;

A6. The advantage of an LCD operating in 8-bit mode is _____

(a) more microcontroller IO pins can be used.
(b) less microcontroller IO pins will be needed.
(c) data can be written to the LCD at a higher rate.
(d) data can be written to the LCD at a lower rate.

A7. The 10-bit result of an analogue to digital conversion is 0b0101000111. When left-justified, _____

(a) ADRESH = 0b01010001 and ADRESL = 0b11000000.
(b) ADRESH = 0b01010001 and ADRESL = 0b11111111.
(c) ADRESH = 0b00000001 and ADRESL = 0b01000111.
(d) ADRESH = 0b11111101 and ADRESL = 0b01000111.

A8. An 8-bit ADC accepts an analogue input voltage from 0 to 5 volt. A digital output of binary 10000000 is likely to correspond to an analogue input of _____

(a) 5 volt.
(b) 0 volt.
(c) 1 volt.
(d) 2.5 volt.

A9. How many times does the loop *for (count = 0; count < 10; count++) …* execute?

(a) 0
(b) 9
(c) 10
(d) forever, i.e. it is an infinite loop

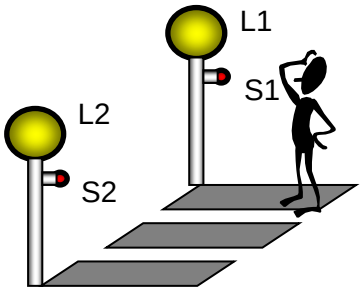
A10. What is achieved by the code *PORTD = temp << 1;* ?

(a) The variable temp is left-shifted by 1 bit and then sent to Port D.
(b) Port D is read and then stored in the variable temp.
(c) Port D is checked to see if it reads a logic 1.
(d) The variable temp is incremented by 1 bit and then sent to Port D.

SECTION B

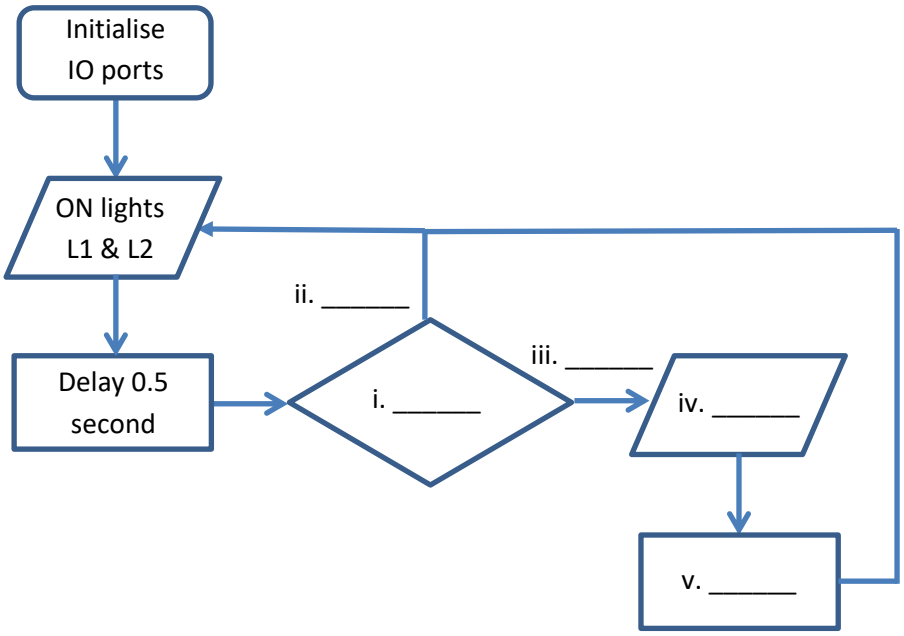
SHORT QUESTIONS [14 marks each]

B1. At the zebra crossing below, the lights (L1 and L2) will light up continuously when no pedestrian is crossing. The lights will blink at 0.5 second intervals when a pedestrian is detected by the sensors (S1 or S2). S1 and S2 are wide angle sensors that can detect a person on the crossing. A PIC18F4550 microcontroller is used and it reads the sensor S1 and S2, and controls the lights L1 and L2.



(a) Draw a block diagram for the design. (4 marks)

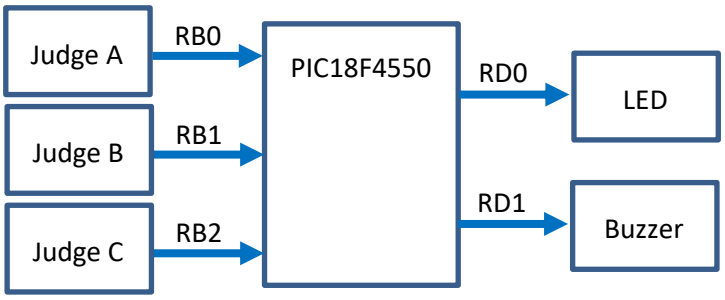
(b) Complete the flow chart for the design below. The following words should appear in the flow chart: pedestrian detected? OFF lights L1 & L2 Delay 0.5 second yes no (10 marks)



Give your answers in the answer booklet as (b) i. *your answer* etc.

B2. The preliminary round of a singing contest is judged by 3 judges A, B and C. If at least 2 of the 3 judges think that a singer is good enough, the singer can go on to the next round.

The block diagram of the electronic gadget used is shown below:



The push button switches for the judges are "active-low" i.e. when a judge presses his or her button (to say that the singer is good enough), it will give a logic 0 to the PIC18 microcontroller.

The microcontroller reads the buttons, does some computation and then EITHER lights up an active-high LED (if the singer is good enough to go on to the next round), OR to turn on an active-high buzzer (if the singer is not good enough to go on to the next round).

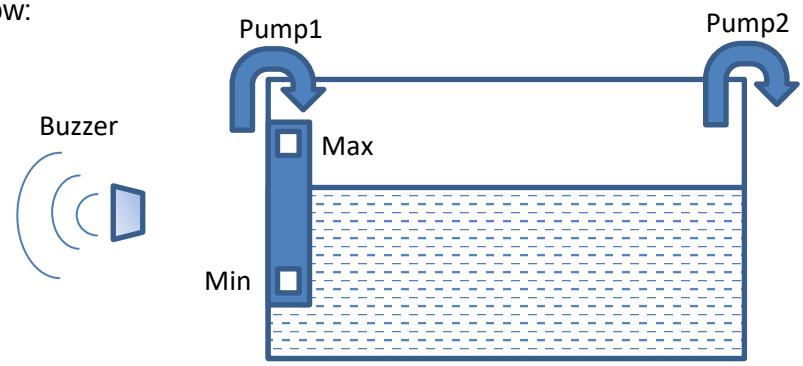
(a) Draw the circuit diagram for an active-low push button switch. (3 marks)

(b) Complete the PIC18 program outlined below. Give your answers in the answer booklet as (b) i. *your answer* etc. (8 marks)

main ( )		if (PORTBbits.RB1 == 1)
{		count = count - 1;
unsigned char count;		_____ // ii. ____
… // other lines not shown		_____ // iii. ____
…		if (count >= 2)
…		PORTD = 0b00000001;
count = _____; // i.____		else
if (PORTBbits.RB0 == 1)		_____ // iv. ____
count = count - 1;		… // other lines not shown

(c) Suggest a suitable output device that can be added to the design so that the audience knows how many judges have given their approval for the singer. (3 marks)

B3. A PIC18 microcontroller-based "water level monitoring" system is used in a fish tank, as shown below:



When the water level is too high (i.e. Max == 1), Pump2 is turned on (Pump2 = 1) to pump away the excess water. When the water level is too low (i.e. Min == 0), Pump1 is turned on (Pump1 = 1) to add water to the tank. Whenever Pump1 or Pump2 is turned on, the buzzer will also be activated, to alert the owner.

The microcontroller pins used for connecting to the I/O devices are shown below:

Max (sensor)	RA3	Active high
Min (sensor)	RA4	Active high
Pump1	RD1	Active high
Pump2	RD2	Active high
Buzzer	RD0	Active high

The water level sensors (Max & Min) can be created from 2 metal strips placed very close to one another, so that the presence of water will short the strips, resulting in a closed circuit between points 1 and 2.

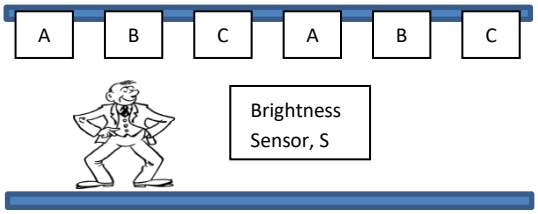
(a) Draw the circuit diagram of a water level sensor. Include Vcc, Ground and a 10k resistor in your circuit. (4 marks)

(b) Write the codes to configure Port A bits 3 and 4 as inputs, and Port D bits 0, 1 and 2 as outputs. (4 marks)

(c) Write the codes (in your answer booklet) to check the water level and to switch the pumps and buzzer on/off (as appropriate), for the "water level monitoring" described above. (6 marks)

```
while(1) { // loop forever
                                // if water level too high
                                //   on Pump2, off Pump1, on Buzzer
                                // else if water level too low
                                //   …
                                // else
                                // …
}
```

B4. An LDR-based brightness sensor (S) is used to detect the level of brightness along a corridor. The sensor output is amplified and then connected to the AN0 input of a PIC18F4550 microcontroller. The corridor lights (set A, set B & set C) are connected to RB2, RB1 & RB0 respectively.



The PIC microcontroller periodically converts the brightness level (an analogue input) into a 10-bit binary code, and right justifies it in the two registers ADRESH:ADRESL. The two most significant bits of the result (i.e. bits 1 & 0 of ADRESH) are then used to control the lights along the corridor, as follows:

ADRESH	Condition	Action to take	C Code
00000011	Very bright	Off all lights	PORTB = 0b00000000;
00000010	Bright	On set A	PORTB = 0b00000100;
00000001	Dark	On sets A & B	PORTB = 0b00000110;
00000000	Very dark	On sets A, B & C	PORTB = 0b00000111;

The program running in the microcontroller is as follows:

```
main (void)
{
   … // other lines of code not shown
   // configure A/D converter module & switch it on
   ADCON0 = 0 b 0 0 ? ? ? ? 0 1; // select AN0 for conversion
   ADCON1 = 0 b 0 0 ? ? 1 1 1 0; // use Vss (0V) as Vref-, and Vdd (5V) as Vref+
   ADCON2 = 0 b ? 0 0 1 0 1 1 0; // right justify 10 bit result
   while (1)
   {
      ADCON0bits.GO = 1;    // starts A/D conversion
      _?_                   // wait here for A/D completion
      _?_                   // use brightness reading to control corridor lights – few lines here
   }
}
```

(a) What binary bit pattern should be put into bits <5:2> Of ADCON0 in order to use AN0 as the analogue input channel? You may need to refer to the Appendix. (2 marks)

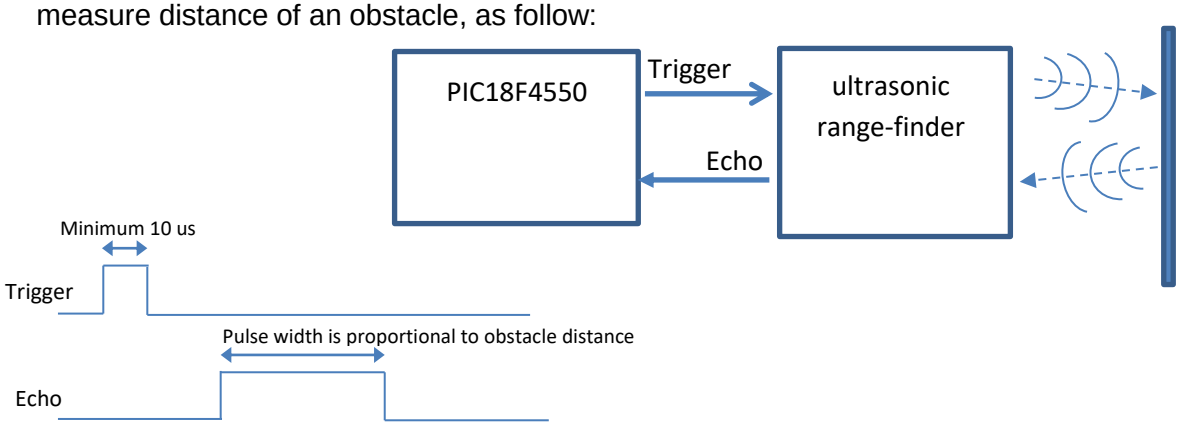
(b) What binary bit pattern should be put into bits <5:4> of ADCON1 in order to use Vss (i.e. 0 volt) as Vref- and Vdd (i.e. 5 volts) as Vref+? (2 marks)

(c) What value (0 or 1) should be put into bit <7> of ADCON2 in order to right-justify the 10 bit conversion result? (2 marks)

(d) Write the C code to wait for the A to D conversion to be completed. (3 marks)

(e) Write the C codes to use the brightness reading to control the corridor lights. (5 marks)

B5. An "ultrasonic range finder" can be connected to a PIC18F4550 microcontroller to measure distance of an obstacle, as follow:



The PIC microcontroller must "trigger" the range-finder by sending it a high pulse that is 10 us long. Assume that RB0 has been made an output pin to be used as Trigger output.

The width of the "echo" pulse from the range-finder to the PIC microcontroller indicates the obstacle distance. Every 58 us of pulse width represents an obstacle distance of 1 cm. Assume that RD1 has been made an input pin to be used as Echo input.

(a) Write the lines of code to produce a 10 us pulse at RB0. You can use the function call delay_us(10); to introduce a delay of 10 us. (5 marks)

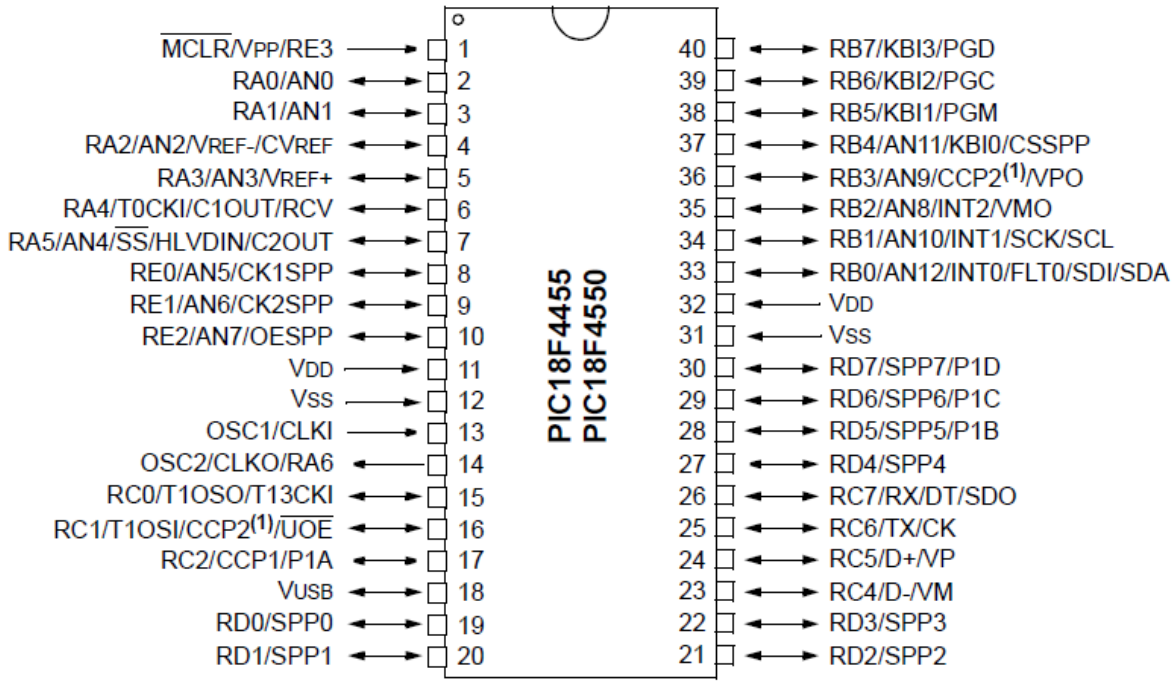
(b) Based on the outline given below, write the lines of code to measure the echo pulse width, at RD1, in multiple of 58 us.

Note: The result should be stored in an unsigned integer variable Count. So at the end, Count = 10 means the echo pulse width is 580us and the obstacle distance is 10cm. (6 marks)

```
Count = ____; // initialise Count
while (_____________); // wait for Echo pulse
while (_____________) // while Echo is high
  {
  _____________  // delay 58us, as each Count = a pulse duration of 58 us
  _____________ // increment Count
  }
```

(c) Suggest how a buzzer can be used with the set-up above, to alert a visually handicapped person of an obstacle less than 50cm away. (3 marks)

APPENDIX - PIC18F4550 – 40-pin PDIP – pin diagram



PIC18F4550 – Analogue to Digital Converter

ADCON1 - The ADCON1 register configures the **voltage references** and the **functions of the port pins**.

U-0	U-0	R/W-0	R/W-0	R/W-0(1)	R/W(1)	R/W(1)	R/W(1)
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **VCFG1:** Voltage Reference Configuration bit (VREF- source)
1 = VREF- (AN2)
0 = VSS

bit 4 **VCFG0:** Voltage Reference Configuration bit (VREF+ source)
1 = VREF+ (AN3)
0 = VDD

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7(2)	AN6(2)	AN5(2)	AN4	AN3	AN2	AN1	AN0
0000(1)	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111(1)	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog input D = Digital I/O

ADCON0 - The ADCON0 register controls the **operation of the A/D module**.

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-2 **CHS3:CHS0:** Analog Channel Select bits
0000 = Channel 0 (AN0)
0001 = Channel 1 (AN1)
0010 = Channel 2 (AN2)
0011 = Channel 3 (AN3)
0100 = Channel 4 (AN4)
0101 = Channel 5 (AN5)
0110 = Channel 6 (AN6)
0111 = Channel 7 (AN7)
1000 = Channel 8 (AN8)
1001 = Channel 9 (AN9)
1010 = Channel 10 (AN10)
1011 = Channel 11 (AN11)
1100 = Channel 12 (AN12)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
When ADON = 1:
1 = A/D conversion in progress
0 = A/D Idle

bit 0 **ADON:** A/D On bit
1 = A/D converter module is enabled
0 = A/D converter module is disabled

ADCON2 - The ADCON2 register configures the **A/D clock source**, **programmed acquisition time** and **justification**.

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified
0 = Left justified

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT2:ACQT0:** A/D Acquisition Time Select bits
111 = 20 TAD
110 = 16 TAD
101 = 12 TAD
100 = 8 TAD
011 = 6 TAD
010 = 4 TAD
001 = 2 TAD
000 = 0 TAD

bit 2-0 **ADCS2:ADCS0:** A/D Conversion Clock Select bits
111 = FRC (clock derived from A/D RC oscillator)
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC (clock derived from A/D RC oscillator)
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

- End of Paper -