

MICROCONTROLLER APPLICATIONS /

ENGINEERING DESIGN & BUSINESS PROJECT II

2018/2019 SEMESTER ONE 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)

2nd Year Full-Time

ET1216

Diploma in Engineering with Business (DEB)

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 statements is true for PIC18F4550?

- (a) A 12 volt power supply can be used.
- (b) There are 32 bytes of flash memory for the program.
- (c) In a “dual in line” package, there should be 10 pins on each side of the IC.
- (d) Analogue input signal can be digitised.

A2. Which one of the following C-codes increments a variable called “count” by 2?

- (a) `count += 1;`
- (b) `count++;`
- (c) `count = count + 2;`
- (d) `count = count << 1;`

A3. An active low push button is connected to RA1 pin of PORTA. Which statement waits for the push button to be released?

- (a) `while (PORTAbits.RA1 == 1);`
- (b) `while (PORTAbits.RA1 == 0);`
- (c) `while (PORTAbits.RA1 = 1);`
- (d) `while (PORTAbits.RA1 = 0);`

A4. An active high 7-segment display unit is connected to PORTD (segment “a” to RD0, segment “b” to RD1... decimal point to RD7). Which command will display a digit 7?

- (a) `PORTD = 0b11100000;`
- (b) `PORTD = 0b00011111;`
- (c) `PORTD = 0b11111000;`
- (d) `PORTD = 0b00000111;`

A5. Which one of the following requires an Analogue-to-digital converter?

- (a) Measuring the level of noise in a library.
- (b) Checking if a button has been pressed.
- (c) Detecting if water in a drain has reached a pre-defined level.
- (d) Turning on an LED.

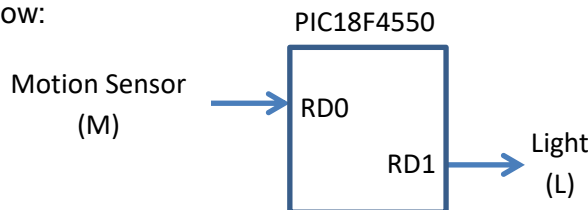
- A6.** Which packaging is suitable for prototyping a microcontroller-based circuit on a bread-board? Why?
- (a) Dual-In-Line package (DIP), because the size is usually smaller.
 - (b) Thin-Quad-Flat-Pack (TQFP), because the size is usually smaller.
 - (c) Dual-In-Line package (DIP), because it can be inserted onto a bread-board.
 - (d) Thin-Quad-Flat-Pack (TQFP), because it is cheaper.
- A7.** Which one of the following is NOT an input device?
- (a) An infra-red detector.
 - (b) A push button.
 - (c) A moisture sensor.
 - (d) A bi-directional motor.
- A8.** Which one of the following regarding the PIC18F4550 ADC is NOT true?
- (a) The result can be left- or right- justified.
 - (b) The digital result is 12-bit.
 - (c) The reference voltages can be set.
 - (d) The acquisition time can be set.
- A9.** Which one of the following is not a main part of a microcontroller?
- (a) CPU (Central Processing Unit).
 - (b) LED display.
 - (c) Memory.
 - (d) I/O (Input/Output) Ports.
- A10.** In a PIC18-based circuit, which of the following connections should NOT be made?
- (a) 5 V should be connected to Vss and ground should be connected to Vdd.
 - (b) A crystal oscillator should be connected to OSC1 & OSC2.
 - (c) The IO ports can be connected to the input/output devices required.
 - (d) An active low button should be connected to MCLR.

SECTION B

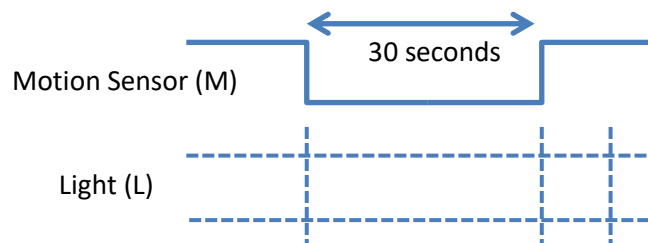
SHORT QUESTIONS [14 marks each]

- B1.** A “sensor light” installed in a public toilet works as follows - When a person is detected (M equals 0), the light is turned ON (L set to 1) immediately. The light is turned OFF (L set to 0) ten seconds after the person leaves (M equals 1).

The PIC18 microcontroller is connected to the Motion Sensor (M) and the Light (L) as shown below:



- (a) From the description, is the Motion Sensor active low or active high? (2 marks)
- (b) Give the C-code to configure the port D pins RD1 and RD0. (3 marks)
- (c) A person enters the toilet, washes his hands and then leaves after 30 seconds. Copy and complete the following waveforms in your answer booklet. (3 marks)



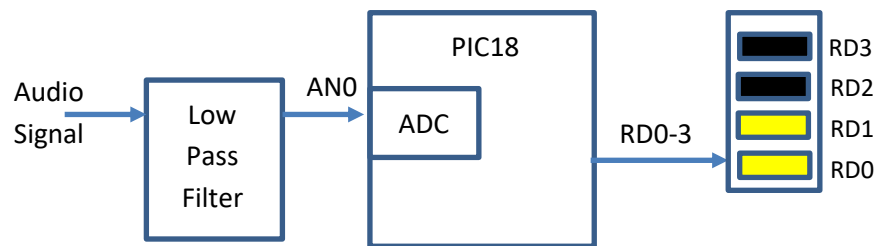
- (d) Arrange the following lines of code in the correct sequence for the PIC18 microcontroller used in the “sensor light”: (4 marks)

<pre>// C-code to configure the port D pins put here PORTDbits.RD1 = 0; // off light at the beginning while(1) { // loop forever if __ (i) __ __ (ii) __ else { __ (iii) __ __ (iv) __ } // else } // while</pre>	Codes to use:
	A. Delay10Seconds; // a function to delay 10 sec
	B. PORTDbits.RD1 = 1; // on light
	C. (PORTDbits.RD0==0) // if person detected
	D. PORTDbits.RD1 = 0; // off light

Note: You can indicate the answers in your answer booklet as (i) A etc.

- (e) Which one of the following sensors is suitable for motion detection: PIR sensor, tilt switch, solenoid? (2 marks)

- B2.** In the set up below, the audio signal (after low pass filtering) is sampled and A-to-D converted by the PIC18 microcontroller.



The result is then displayed on the LED bar connected to PORTD, as follows:

ADC Digital Result (2 most significant bits)	Which LED(s) to light up
00	All OFF
01	LED connected to RD0
10	LEDs connected to RD0 & RD1
11	All ON

- (a) The two most significant bits of the AD conversion result are used to determine which LED(s) is/are to turn on. Should the 10-bit AD conversion result be left or right justified? (2 marks)
- (b) Assuming Vss (0V) and Vdd (5V) are used as the reference voltages and AN0 is the only analogue input pin. What should be the setting for ADCON1? (2 marks)
- (c) Fill in the blanks in the following code segment written for the PIC18 microcontroller. You can indicate the answers in your answer booklet as i. your answer etc.

```

...
while(1) {
    ADCON0bits.GO = 1;
    while (ADCON0bits.GO==1);
    if (ADRESH==0b00000000)
        PORTD = 0b00000000; // All OFF
    else if (ADRESH==0b____i.____)
        PORTD = 0b____ii.____; // LED connected to RD0 on
    else if (ADRESH==0b____iii.____)
        PORTD = 0b____iv.____; // LEDs connected to RD0 & RD1 on
    else
        PORTD = b00001111; // All ON
    // delay...
}

```

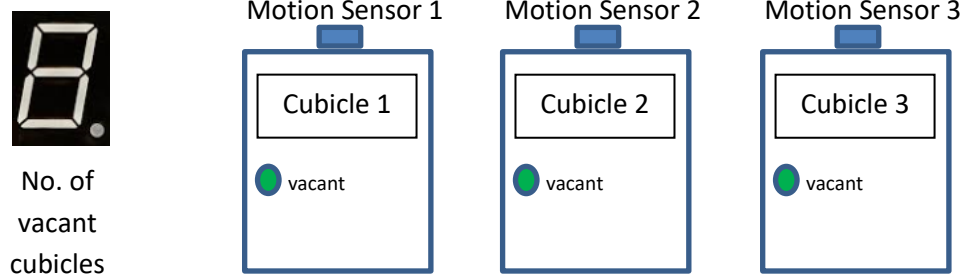
(4 marks)

- (d) What is the purpose of the line `while(ADCON0bits.GO==1);` above? (2 marks)
- (e) If the reference voltages are 5V and 0V and the audio signal (after filtering) is 2.8V. Determine which LED(s) will be turned on.

Digital result = $(V_{in} / 5) \times 1023$, where V_{in} is the voltage at AN0.

(4 marks)

- B3.** In a public toilet, 3 motion sensors, 3 green LEDs and a 7-segment display unit are used to show the availability of toilet cubicles.



The motion sensor in each cubicle gives logic '1' when a cubicle is occupied, and '0' otherwise. The three motion sensors are connected to RA1, RA2 and RA3 of a PIC18 microcontroller. The 7-segment display unit is connected to PORTD (segment 'a' to RD0, segment 'b' to RD1... decimal point to RD7).

An outline of the PIC18 program is given below:

1	while(1) {	7	_____ // ans of part (c)
	// count number of vacant cubicles	8	PORTD = 0b01011011; // show 2
2	num = Count_vacancies ();	9	else // three vacant
3	if (num == 0) // all occupied	10	PORTD = 0b01001111; // show 3
4	PORTD = 0b00111111; // show 0,	11	// other lines here....
5	else if (num == 1) // one vacant	12	}
6	_____ // ans of part (b)		

- (a) Draw a circuit diagram to show how the common cathode 7-segment display unit can be connected to PORTD of the PIC18 microcontroller. Include any resistors needed. (5 marks)
- (b) Write the code to show '1' on the 7-segment display unit (line 6). (2 marks)
- (c) Write the missing statement (line 7), including an appropriate comment. (2 marks)
- (d) Write the function Count_vacancies (in your answer booklet) to determine the number of vacant cubicles. You can use the outline below: (5 marks)

```

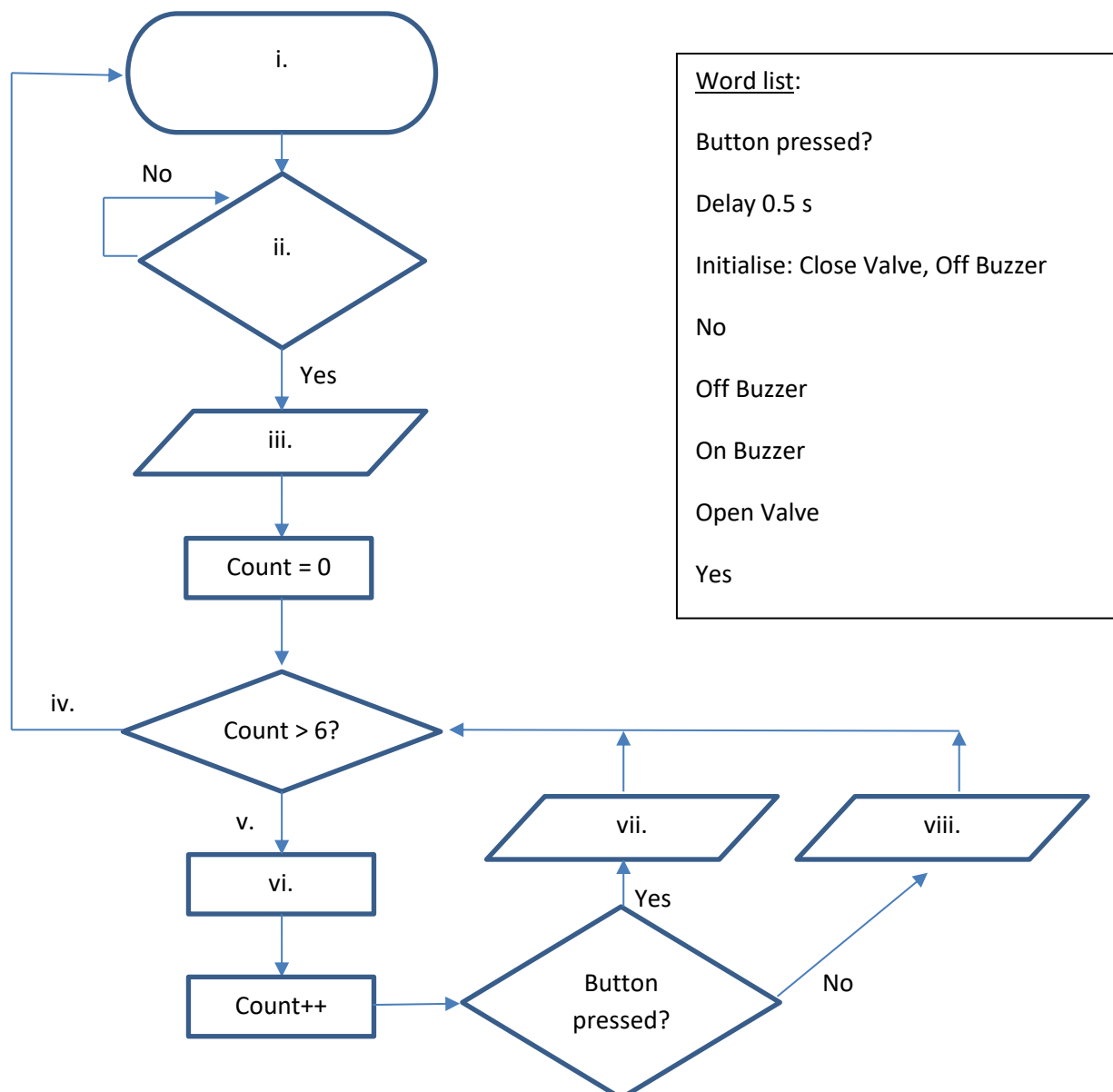
unsigned char Count_vacancies (void)
{
    unsigned char vacancy = 0;
    if (... ) vacancy++;
    ...
    ...
    return(vacancy);
}

```

- B4.** A drinking water dispenser fills a cup with water in 3 seconds, each time a (push button) switch W is pressed (i.e. $W = 1$). But if W is pressed again during the 3 seconds, a buzzer B will sound (i.e. $B = 1$). The water is dispensed by opening a valve V (i.e. $V = 1$).

A PIC18 is used as the microcontroller to control the operation of the dispenser.

- (a) Draw the block diagram for the system, indicating clearly how the switch, the buzzer and the valve should be connected to the PIC18 microcontroller. (4 marks)
- (b) Complete the flowchart below, using the word list below. Indicate the answers in your answer booklet as i. your answer etc. (8 marks)

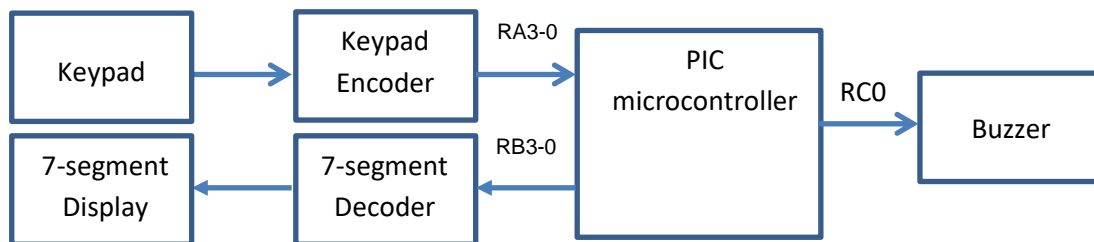


- (c) If a user accidentally presses the button while the water dispensing has just started, and quickly releases the button once he hears the buzzer sound, what is the duration of the sound from the buzzer? (2 marks)

- B5.** At a popular “Nasi Lemak” stall, a simple Q-number system is used. After placing an order, the customer is issued a ticket numbered 0 to 9.

When the food for that customer (say, Q-no 3) is ready, the stall operator presses a number key (‘3’) on the keypad and the 7-segment display will show the number (‘3’). A buzzer will also beep to alert those customers still waiting for their food. The customer returns the ticket when collecting the food.

The block diagram of the system is shown below.



The keypad encoder produces a 4-bit binary number corresponding to the number key pressed (e.g. binary 0011 if ‘3’ pressed). If no key is pressed, it gives 1111.

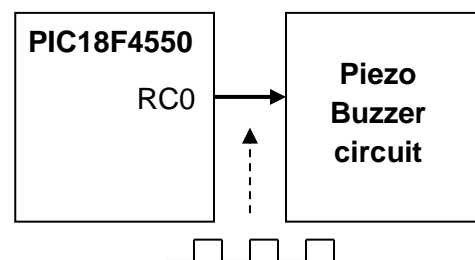
- (a) What does the keypad encoder produce when key “6” is pressed? (3 marks)
- (b) The 7-segment decoder accepts a 4-bit binary number and produces 7-bits (a to g) for the 7-segment display (e.g. binary 0101 for ‘5’ to be displayed).

What does the 7-segment display show when RB3-0 equal binary 0100? (3 marks)

- (c) The beep sound from the buzzer is produced by a function called BUZZ outlined below:

```

void BUZZ (void) {
  for ( k = 0; k < 100; k++ ) {
    delay_ms (5);
    PORTCbits.RC0 = !PORTCbits.RC0;
  }
}
  
```



The customers complain that the “beep” sound from the buzzer is too short. What change to the code is necessary so that “beep” sound lasts longer? (4 marks)

- (d) The 4 lines of code below are required in the while (1) loop of the main program for the above application. Arrange them in correct order. (Key has been declared as an unsigned char.)

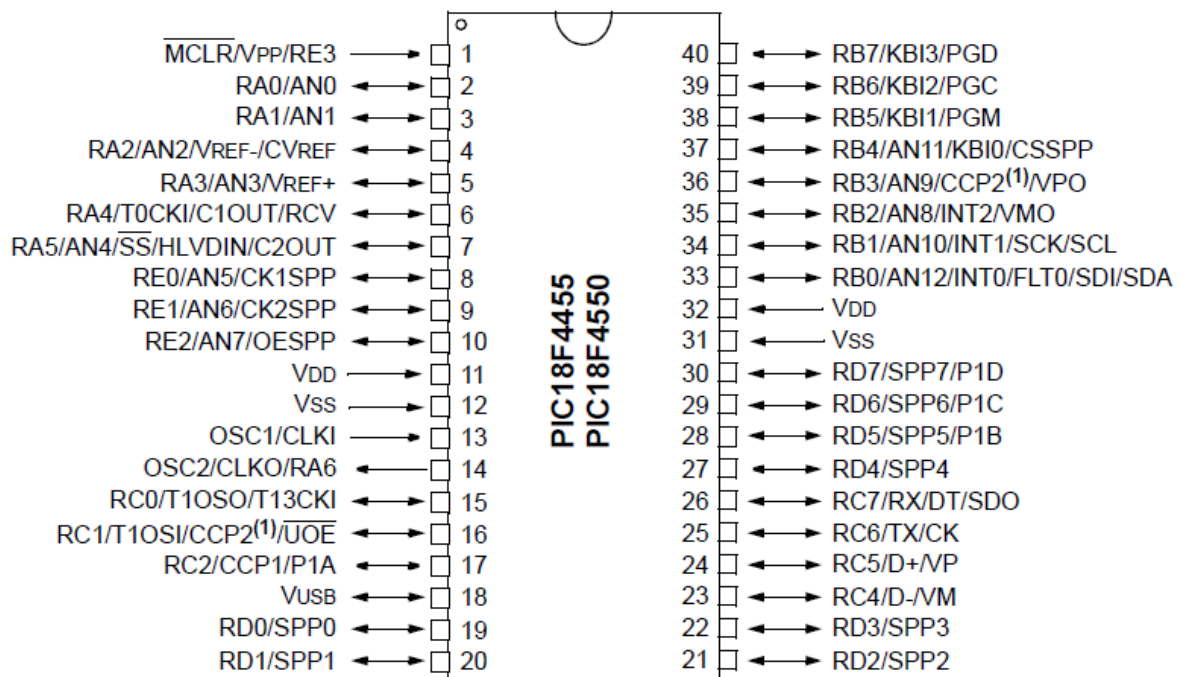
while ((PORTA & 0x0F) == 0x0F); // wait if bottom 4 bits all 1’s

PORTB = Key; // write to RB3-0

BUZZ (); // produce beep sound

Key = PORTA & 0x0F; // read RA3-0

(4 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/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/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 -