Microprocessor Principles and Applications

Final Exam

Fall 2022

The exam is 120 minutes long. The total score is 101pts. Please read questions carefully.

1. (12 pts) Assume PIC18F4321. Suppose that three switches are connected to bits 0–2 of port C and an LED to bit 6 of port D. If the number of HIGH switches is even, turn the LED ON; otherwise, turn the LED OFF. Write an assembly language program to accomplish this using "if-else" construct. Assume that a '1' will turn the LED ON while a '0' will turn it OFF.

```
#include"p18f4321.inc"
MOVLW   0x0F
MOVWF   ADCON1  ; AN0~AN12 are digital
SETF    TRISC   ; input
BSF     TRISD, TRISD6 ; output
BCF     PORTD, RD6    ; turn off
MOVLW   0x07    ; 0b00000111
ANDWF   PORTC, W
BZ      Turn_on ; 全都為0, 偶數.
MOVLW   0x01
ANDWF   PORTC, W
MOVWF   0x00 ; 拿第0個bit
MOVLW   0x02
ANDWF   PORTC, W
MOVWF   0x01
RRNCF   0x01 ; 拿第1個bit
MOVLW   0x04
ANDWF   PORTC, W
MOVWF   0x02
RRNCF   0x02
RRNCF   0x02 ; 拿第3個bit
MOVF    0x00, W
XORWF   0x01, W
XORWF   0x02, W ; RC0^RC1^RC2
BTFSS   WREG, 0 ; 汙 XOR結果
BRA Turn_on // 等於0
BRA Turn_off // else, 等於1
Turn_on: BSF PORTD, RD6
         goto finish
Turn_off:
         BCF PORTD, RD6
finish:
         end
```

2. (15 pts) In PIC18F, what is the internal clock and what is the external clock? What is the main disadvantage of the internal oscillator?

Internal clock：利用內建的 internal oscillator 的頻率來產生的 clock。

External clock：需要在 OSC pin 上接額外的 device，如 crystal 或 DC oscillator，再利用他們的頻率來生成的 clock。

Disadvantage of internal oscillator：Lack of precision and frequency stability，像是周圍的溫度會影響到電阻、電容的值，而 oscillator 就需要依靠一些被動元件來運作，因此就會影響到他的準度及產生頻率的穩定性。

(a) (8 pts) What is the difference between unconditional I/O and conditional I/O?
(b) (8 pts) What is the difference between polled I/O and interrupt I/O?

(a) unconditional I/O : Microcontroller 可以在"任意時間"和 external device 進行資料傳輸的動作。

conditional I/O : Microcontroller 需要經過"handshaking"的動作才能傳輸資料給 external device，也就是需要特定的事件觸發後，control signal 在 microcontroller 和 external device 之間交換後，才能去做 I/O。

(b) polled I/O : 通常會有一個 busy 的 signal，而當 busy 被設為 high 後，microcontroller 才能繼續往下執行其它事情，否則 microcontroller 只能在 loop 裡面不斷地 busy-waiting，直到特定事件觸發使得 busy signal 為 high，而這就會浪費 CPU 的資源，而且依賴 external device 的速度。

interrupt I/O : External device 需要接到 interrupt (INT) pin上，而當 flag 被設為 high 時，interrupt 就會觸發，microcontroller 會執行完當下指令，且將目前狀態及 Program counter 存到 stack，接著跳去執行 interrupt service routine (ISR)，當執行到 RETURN 或 RETFIE 後，代表 ISR 結束，此時再 pop stack 的資訊，也就是回到原本的位置，而這樣就不用 busy-waiting，因此較有 CPU。

4. (8 pts) Use Timer0 in 16-bit mode. Assume an 4-MHz crystal, and a prescale value of 1:4. If you need to write a PIC18F assembly language program to obtain a time delay of 1 ms, what values should be set to TMR0H and TMR0L?

$F_{osc} = 4MHz \Rightarrow T_{osc} = 0.25\,us \Rightarrow T_{timer} = 1\,us$

1ms delay $\Rightarrow 1000 = (count+1) \times 4 \times 1$

(overflow, prescaler, Timer)

$count = 249 = 0XF9$

$249 \rightarrow 0XFF06 \rightarrow 0XFFFF \rightarrow 1 \rightarrow 0X0000$

$TMR0H:TMR0L = 0XFFFF - 0X00F9 = 0XFF06$

$\Rightarrow TMR0H = 0XFF$

$TMR0L = 0X06$

#

(8 pts) Suppose that there is a 10-bit A/D converter with VREF- = 1 V and VREF+ = 5 V. Find the corresponding voltage values for the A/D conversion results of 40, 500, and 1000. (Note: calculation process is necessary.)

10-bit result $\Rightarrow 0 \sim 1023$

voltage $\Rightarrow 1 \sim 5$

$40: \frac{1023-0}{5-1} = \frac{40-0}{X-1}$, $1023(X-1)=160$, $X = 1 + \frac{160}{1023} \doteq 1.16\,(V)$

$500: \frac{1023-0}{5-1} = \frac{500-0}{Y-1}$, $1023(Y-1)=2000$, $Y = 1 + \frac{2000}{1023} \doteq 2.96\,(V)$

$1000: \frac{1023-0}{5-1} = \frac{1000-0}{Z-1}$, $1023(Z-1)=4000$, $Z = 1 + \frac{4000}{1023} \doteq 4.91\,(V)$

6. (18 pts) What is the main purpose the following modes. Explanation of how each of them works and the corresponding application should be provided.
   (a) (6 pts) Capture mode
   (b) (6 pts) Compare mode
   (c) (6 pts) PWM mode.

(a) Capture mode：當第n個 rising edge 或 falling edge（由 user 決定）被偵測到時，timer 1 或 timer 3 (TMRxH:TMRxL) 就會 copy 到 16-bit register (CCPRxH:CCPRx2)。因此可以用來測量兩個 event 中間的 period。

(b) Compare mode：當 timer1 或 timer3 的值(TMRxH:TMRxL)等於16-bit register(CCPRxH:CCPRxL)的值時，就能改變 CCP pin 輸出的值，如 low→high，high→low 或 toggle。因此可以用來在特定的 time delay 後產生不同的 CCP output，等於生成一個波形。

(c) PWM (pulse width modulation) mode：
透過設定 timer2 的 TMR2, PR2, CCPR1L 以及 CCPxCON 的 DCXB bits，可以產生特定 frequency 以及特定 duty cycle 的 square wave。因此可以用來控制燈泡的閃爍或是 DC motor 的速度，當 duty cycle 越大就代表 high 的時間越多，燈泡就會越亮等等。

(12 pts) Write a PIC18F assembly language program at 0x100 to generate a 16 kHz PWM with a 75% duty cycle on the CCP1 pin of the PIC18F4321. Assume 4 MHz crystal and CCP2 is used. Note that, you need to provide the calculation process of how you get the setup for PR2 and duty cycle.

$F_{osc} = 4MHz \Rightarrow T_{osc} = 0.25\mu s \Rightarrow T_{timer} = 1\mu s$ $F_{PWM} = 16kHz \Rightarrow T_{PWM} = 0.625ms$

$(PR2+1) \times 1 \times 1 \times 4 = 625$ (1 = pre, 4 = post)

$PR2+1 = 156.25 \doteq 156 \Rightarrow PR2 = 155 \Rightarrow$ duty cycle $= 155 \times 0.75 = 116.25$

```
#include "p18f4321.inc"
ORG 0x100
MOVLW 0x0F
MOVWF ADCON1 ; all digital
MOVLW 0x20
MOVWF OSCCON ; 4MHz
MOVLW 0x1C
MOVWF CCP2CON ; 0.25, PWM mode
MOVLW 0x01
MOVWF T2CON ; prescale 1:1, postscale 1:4
MOVLW 0x9B
MOVWF PR2 ; 155
MOVLW 0x74
MOVWF CCPR1L ; 116
BCF PIR1, TMR2IF ; flag clear
BSF T2CON, ON ; start timer
back: BTFSS PIR1, TMR2IF
goto back
BCF PIR1, TMR2IF
BCF T2CON, ON ; stop.
end
```

8. (12 pts) Assume PIC18F4321. Write a C-program that will measure the period of a periodic pulse train on the CCP1 pin using the capture mode. The 16-bit result will be performed in terms of the number of internal (Fosc/4) clock cycles, and will be available in the TMR1H:TMR1L register pair. Use 1:1 prescale value for Timer1.

```
#include "p18f4321.h"
unsigned char First_H, First_L, period_H, period_L
void main(void){
  ADCON1 = 0x0F;
  CCP1CON = 0x05; // capture mode, every rising edge
  T3CON = 0x00; // timer1 for ccp module
  T1CON = 0x00; // prescale 1:1
  PIR1bits.CCP1IF = 0; //flag clear
  while(PIR1bits.CCP1IF == 0); //waiting
  First_H = CCPR1H;
  First_L = CCPR1L; //記住第一次
  TMR1H = 0;
  TMR1L = 0;
  T1CONbits.ON = 1; //start timer
  PIR1bits.CCP1IF = 0; //flag clear
  while(PIR1bits.CCP1IF == 0); //waiting
  T1CONbits.ON = 0; //stop timer
  period_H = CCPR1H - First_H;
  period_L = CCPR1L - First_L;
  //result
}
```