Student Name:	蔡汉霖
Fuzhou University ID:	832002117

Student Name:	卿祺果
Fuzhou University ID:	832002127

EE302 Lab2

Introduction：

EQUIPMENT：MPLAB Simulator, PIC16F877A, PICkit-3, LCD Module.

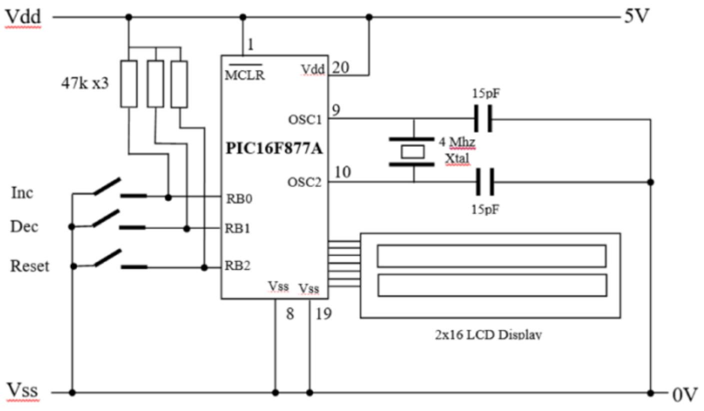


Figure 1 Operation of PIC16F877A

Figure 1 shows the operation of the PIC16F877A.

Part 1

In this part, we are required to outline the design of the program by using pseudo code. And the Table 1 shows the pseudo code.

Table 1 Pseudo Code for the Program

Pseudo code:

```
set configuration bits in code to set operation mode of PIC16F877A
include xc.h
include studio.h
include ee302lcd.h
define Xtal frequency 4Mhz for delay functions

cnt ← 0
main():
        setup()
        show text
        for (all time)
                loop()
```

```
setup():
    initialize LCD
    set PORTB bits 0, 1 and 2 as inputs
loop():
    if switch 1 pressed:
        if cnt less than 20:
            cnt ← cnt + 1
    if switch 2 pressed:
        if cnt greater than 0:
            cnt ← cnt - 1
    if switch 3 pressed:
        cnt ← 0
    show cnt
```

Part 3

The complete code of C program is shown in Table 2.

Table 2

The C program based on MPLAB

```
/**
 * File:     lab2-demo.c
 * Platform:  PICmicro PIC16F877 @ 4 Mhz
 * Author:    Hanlin Cai （蔡汉霖）
 *
 *
 * Function:   1 Read status of three switches Inc (RB0) Dec (RB1) and Reset (RB2).
 *            2 Start up display a title (Laboratory 2 EE302)
 *            3 Inc is pressed a button pressed counter is incremented once per
button press up to a maximum of 20
 *            4 Dec is pressed a button pressed counter is decremented once per
button press down to a minimum of 0
 *            5 Reset is pressed the counter should be reset to 0.
 *
 *
 * Created on 2022/10/05, 21:06
 **/
#include <pic.h>              // Include header file for PIC microcontroller
#include <stdio.h>            // Include Standard IO header file (this file includes
functions such as "sprintf.."
#include "lab-2.h" // Include required header file for LCD functions
```

```c
// CONFIG
#pragma config FOSC = XT       // Oscillator Selection bits (XT oscillator)
#pragma config WDTE = OFF      // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF     // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOREN = OFF     // Brown-out Reset Enable bit (BOR disabled)
#pragma config LVP = OFF       // Low-Voltage (Single-Supply) In-Circuit Serial
Programming Enable bit (RB3 is digital I/O, HV on MCLR must be used for programming)
#pragma config CPD = OFF       // Data EEPROM Memory Code Protection bit (Data
EEPROM code protection off)
#pragma config WRT = OFF       // Flash Program Memory Write Enable bits (Write
protection off; all program memory may be written to by EECON control)
#pragma config CP = OFF        // Flash Program Memory Code Protection bit (Code
protection off)


// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#ifndef _XTAL_FREQ
 // Unless already defined assume 4MHz system frequency
 // This definition is required to calibrate the delay functions, __delay_us() and
__delay_ms()
 #define _XTAL_FREQ 4000000
#endif


// Definitions____________________________________________________________
// Read status of three switches Inc (RB0) Dec (RB1) and Reset (RB2).
#define Closed 0             // Define "Closed" as 0 to correspond to Switch action
#define Inc RB0              // Label RB0 as Inc
#define Dec RB1              // Label RB1 as Dec
#define Reset RB2            // Label RB2 as Reset


// globals ______________________________________________________
unsigned char button_press;           //button press counter


// Prototypes_____________________________________________________________

void init(void);                      //Declare init() function
void LCD_demo(void);                 //Declare LCD_demo() function
void LCD_title(void);                 //Declare LCD_title() function
```

```
// main entry point to application

void main(void)
{

          init(); //do intialisation
          LCD_title(); //display a title on LCD

         // superloop...
          for(;;)
          {
                    LCD_demo();
//main function to detect button presses and increment/decrement
//counter appropriately, and display result to LCD
          }
}



//Initialisation
void init(void)
          {
          Lcd8_Init();       // Required initialisation of LCD to 8-bit mode
          TRISB=0x07;        // Set PORTB bit 0 as input
          TRISA=0x01;        // Set PORTA bit 0 as input
          button_press = 0;  // Initialise button_press variable to 0
          }

//Start up display a title (Laboratory 2 EE302)
void LCD_title(void)
          {
          Lcd8_Write_String("Laboratory2");
// display "Laboratory 2" on first line of LCD
         Lcd8_Set_Cursor(2,0);        // select line 2 of LCD
          Lcd8_Write_String("EE302"); // display "EE302" on second line of LCD
          }



void LCD_demo(void)
          {
          unsigned char string_output[16];
//character array for LCD string. 16 char corresponds to one line of LCD

                    if ((Inc == Closed) && (button_press < 20))
                   // if SW1 closed AND button_press less than 99
                   {
                             button_press++;  //increment button press count
                             Lcd8_Clear();                            //clear LCD
                             sprintf(string_output,"  Button Press");
//load character array "string_output" with characters to be displayed
                             Lcd8_Write_String(string_output);
                             //diplay contents of "string_output
                             Lcd8_Set_Cursor(2,0);
                             //select line 2 of LCD
                             sprintf(string_output,"     = %d",button_press);
         //connvert "button_press" value to a character
                             Lcd8_Write_String(string_output);
         //diplay contents of "string_output"

                             while (Inc == Closed);
//wait for button to be released before continuing

                             __delay_ms(100);
         //switch debounce delay
                   }

                   if ((Dec == Closed) && (button_press > 0))
// if SW2 closed AND button_press greater than 0
                   {
                             button_press--;    //deccrement button press count
                             Lcd8_Clear();                            //clear LCD
                             sprintf(string_output,"  Button Press");
//load character array "string_output" with characters to be displayed
                             Lcd8_Write_String(string_output);
                             //diplay contents of "string_output
                             Lcd8_Set_Cursor(2,0);
                             //select line 2 of LCD
                             sprintf(string_output,"     = %d",button_press);
         //connvert "button_press" value to a character
                             Lcd8_Write_String(string_output);
                             //diplay contents of "string_output"

                             while (Dec == Closed);
//wait for button to be released before continuing

                             __delay_ms(100);            //switch debounce delay
                   }
                   if (Reset == Closed)         // if Reset pressed
                   {
                            button_press = 0;    //resett button press count to 0
                            Lcd8_Clear();        //clear LCD
                            sprintf(string_output,"  Button Press");
//load character array "string_output" with characters to be displayed
                            Lcd8_Write_String(string_output);
                            //diplay contents of "string_output
                            Lcd8_Set_Cursor(2,0); //select line 2 of LCD
                            sprintf(string_output,"     = %d",button_press);
          //connvert "button_press" value to a character
                            Lcd8_Write_String(string_output);
                            //diplay contents of "string_output"

                            while (Reset == Closed);
//wait for button to be released before continuing

                            __delay_ms(100);     //switch debounce delay
                   }
          }

// This .c is modified by Hanlin Cai
// In 2022/10/06
```

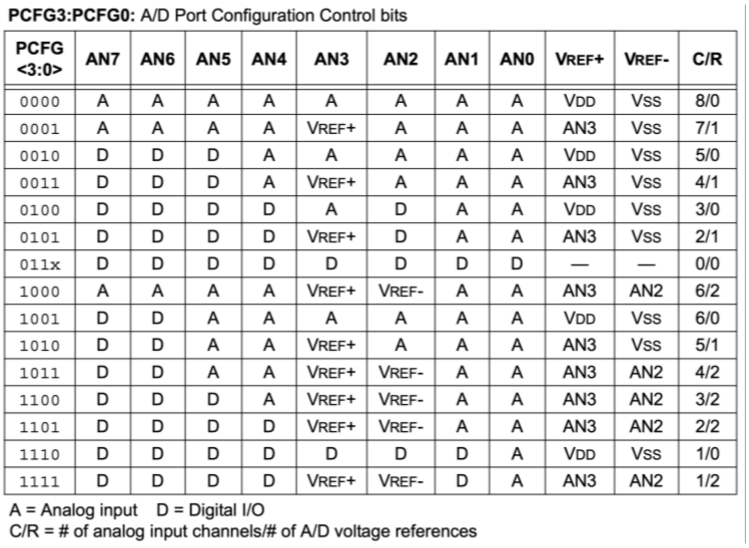
Part 3

In the ee302lcd.h header file there is a LCD initialisation function called, Lcd8_Init(). In this function the ADC special function register ADCON1 is set to 0x07. (a) Why is this necessary for the digital LCD device to work? (b) Which analog input pins AN0 to AN7 must be configured to digital I/O in order for the LCD to work? (c) Give another value for bits 3-0 (PCFG3:PCFG0) which would also enable the LCD to work?

a) Due to LCD required digital signal to communicate, the corresponding pin in the PIC16F877A should also be set to digital I/O instead of analog input.

b) The RS, R/W and E pin in the LCD are connected to RE0, RE1 and RE2 pin on the board. By checking the datasheet (Figure 2), we can figure out that RE0, RE1 and RE2 refer to AN5, AN6 and AN7 pin. So AN5 to AN7 should be configured to digital I/O to make LCD work.

c) According to the datasheet below, in order to set AN5, AN6 and AN7 as digital I/O, we should set ANCON1 bit 3 - 0 to (0x1x, 010x or 11xx).



PCFG3:PCFG0: A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O
C/R = # of analog input channels/# of A/D voltage references

Figure 2 Datasheet of LCD

Summary for this Lab 2

In this Lab, we have learned the basic knowledge of PIC16F877A and how to use the MPLAB to display the LCD module in PIC. Lastly, thanks to DR. Chen, for her patient explanation and guidance.

Hanlin Cai and Qiguo Qing.

In 2022/10/06.