Student Name:	蔡汉霖
Fuzhou University ID:	832002117

Student Name:	卿祺果
Fuzhou University ID:	832002127

EE302 Lab 3

Introduction：

EQUIPMENT：MPLAB Simulator, PIC16F877A, PICkit 3, LCD Display. Figure 1 below depicts a basic configuration of an embedded system based around the PIC16F877A.

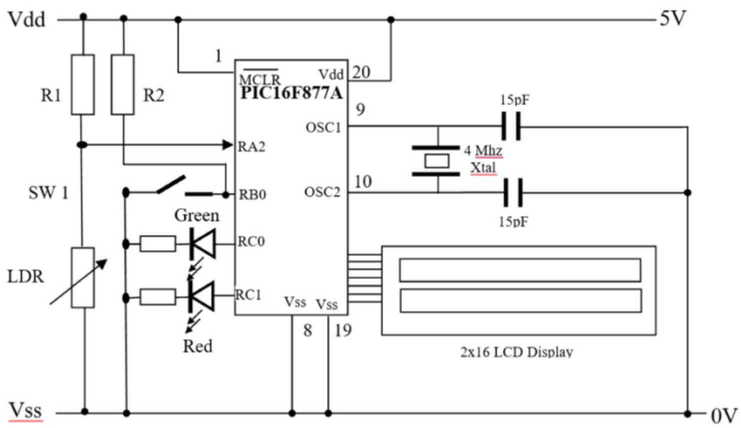

Figure 1 Basic Configuration

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
 	
setup():
    initialize LCD
    set PORTA bits 2 as inputs
    set PORTB bits 0 as inputs
    set PORTC all as outputs

```
main():
    setup()
    show Title
    if switch 1 pressed:
    for (all time)
        loop()


loop():
    GO_nDONE ← 1
    if GO_nDONE is 0:
        show detectedVoltage()
        if detectedVoltage() < 2:
            light LEDRED
        else:
            light LEDGREEN

detectedVoltage():
    return (ADRESH << 2) * (REF+) / (2 << 10))
```

Part 2

Q: How is the ADC sampling rate set in the superloop? What considerations should be given to the functions in the superloop in order to achieve the desired ADC sampling rate of ~10 samples per second. (10 marks)

A: According to datasheet, minimum acquisition time is given as $T_{ACQ} = 19.72\mu s$

EQUATION 11-1: ACQUISITION TIME

TACQ	=	Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient
	=	TAMP + TC + TCOFF
	=	2 µs + TC + [(Temperature – 25°C)(0.05 µs/°C)]
TC	=	CHOLD (RIC + RSS + RS) In(1/2047)
	=	- 120 pF (1 kΩ + 7 kΩ + 10 kΩ) In(0.0004885)
	=	16.47 µs
TACQ	=	2 µs + 16.47 µs + [(50°C – 25°C)(0.05 µs/°C)
	=	19.72 µs

And one acquisition period is about 12 T_{AD}

As a result, the minimum $T_{AD} = 1.6\mu s$

It is given that $F_{OSC} = 4MHz$

So $T_{OSC} = \frac{1}{F_{OSC}} = 0.25\mu s$

In order to satisfy such conditions, we have to choose operations below.

TABLE 11-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (F))

AD Clock Source (TAD)		Maximum Device Frequency
Operation	**ADCS2:ADCS1:ADCS0**	
2 TOSC	000	1.25 MHz
4 TOSC	100	2.5 MHz
8 TOSC	001	5 MHz
16 TOSC	101	10 MHz
32 TOSC	010	20 MHz
64 TOSC	110	20 MHz
RC(1, 2, 3)	x11	**(Note 1)**

e.g. $8T_{OSC}$ to $64T_{OSC}$

So the maximum acquisition period is $64 * T_{OSC} * 12 = 192\mu s$, which is far less than required period. So we consider than acquisition time can be ignored and delay $0.1s$ each run.

Part 3

Q: In special function register ADCON1 (page 128 of the PIC16F877A datasheet) bits 3-0 are used to set the A/D Port Configuration Control bits. Identify all lines of the table which meet our requirements, i.e.

- *RA2/AN2 : Analog input*

- *RE0/AN5, RE1/AN6 and RE2/AN7 must be Digital I/O for the LCD*

- V_{ref+} *and* V_{ref-} *should be set to* V_{dd} *and* V_{ss}*. (10 marks)*

Answer:

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

According to the datasheet above, when PCFG<3:0> is 0010 satisfy all the conditions.

Part 4

(a) List the Special Function Registers (SFR) associated with this program task and detail the bit configuration for each.

TRISA	0x04
TRISB	0x01
TRISC	0x00
ADCON0	0x51
ADCON1	0x02

(b) Explain what each SFR does.

SFR	Function
TRISA	Set AN2 as input
TRISB	Set PORTB bit 0 as input
TRISC	Set PORTC all bits as output
ADCON0	1. Set Clock Conversion frequencies. 2. Select AN2 as analog input channel
ADCON1	1. Set Clock Conversion frequencies. 2. Select left justified. 3. Configure A/D Port configuration bits

(c) What ADC channel is used in the circuit in figure 1? What changes are required to use ADC channel AN3 instead? (15 marks)

We use AN2 as ADC input channel

When we are using AN3, TRISA should be set to 0x08 and ADCON bit5-3 should be set to 011.

Part 5

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

Table 2
The C program based on MPLAB

```
/*
 * File:   lab3_main.c
 * Author: hanlincai & qiguoqing
 *
 * Created on October 5, 2022, 7:45 PM
 */

// 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.

#include <xc.h>
#include <stdio.h>     // Include Standard I/O header file
#include "ee302lcd.h"        // Include LCD header file. This file must be in same
directory as source main source file.

#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


#define SW1 RB0                        // Assign Label SW1 to PortB bit 0 (RB0)
```

```
#define G_LED RC0
#define R_LED RC1
#define CLOSED 0
#define OPEN 1
#define ON 1
#define OFF 0

const double threshold = 2.0;

void setup() {
   Lcd8_Init(); // Required initialisation of LCD to 8-bit mode

   TRISA = 0x04;
   TRISB = 0x01;
   TRISC = 0x00;

   ADCON0 = 0x51;
   ADCON1 = 0x02;
}
//
void lcdTitle() {
   Lcd8_Write_String("Laboratory 3");          // print "LCD Demo" on line 1 of LCD
         Lcd8_Set_Cursor(2,0);                                  // select line 2
         Lcd8_Write_String("EE302-ADC");                // print "LCD Demo" on
line 2 of LCD
}

int get_vol() {
   unsigned int ret = 0;
   GO_nDONE = 1;
   while (GO_nDONE) {
      continue;
   }
   ret += ((unsigned int) ADRESH) << 2;
   return ret;
}

void show() {
   __delay_ms(100);
   GO_nDONE = 1;
   while(GO_nDONE){
      continue;
   }
```

```c
    char temp[10];
    double voltage = (double) get_vol() * 5 / 1024;
    sprintf(temp, "%.1f", voltage);
    Lcd8_Set_Cursor(2, 6);
    Lcd8_Write_String(temp);
    Lcd8_Write_String("V");

    if (voltage < threshold) {
        R_LED = ON;
        G_LED = OFF;
    } else {
        R_LED = OFF;
        G_LED = ON;
    }
}

void loop() {
    show();
    __delay_ms(100);
}


void main() {
    setup();
    lcdTitle();

    while (SW1 == OPEN);
    Lcd8_Clear();
    Lcd8_Set_Cursor(1, 1);
    Lcd8_Write_String("ADC Voltage is");

    while (1) {
        loop();
    }
}
```

Summary for this Lab 3

In this Lab 3, we have learned the basic knowledge of ADC and known how to use PIC DIP-40 to verify the function of our C program. Thanks to Dr. Wu for her guidance.

HanlinCAI 832002117 & QiguoQING 832002127

In 2022/10/26