Student Name	蔡汉霖
Fuzhou University ID	*832002117*

Student Name	卿祺果
Fuzhou University ID	*832002127*

Report for EE302FZ Lab 4

Introduction

The following Figure 1 depicts a basic configuration of an embedded system based around the PIC16F877A. And Figure 2 below depicts a system overview of the embedded system and a bidirectional connection to a PC.

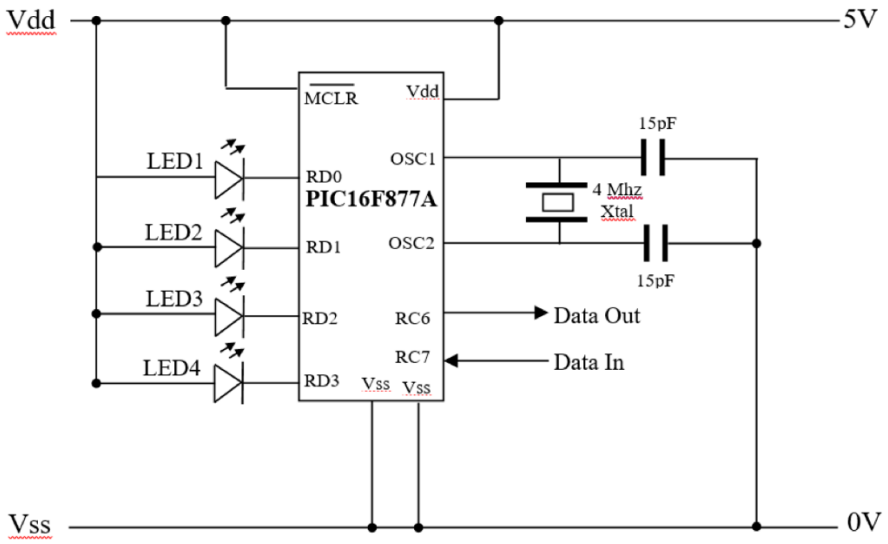

Figure 1 Basic Configuration

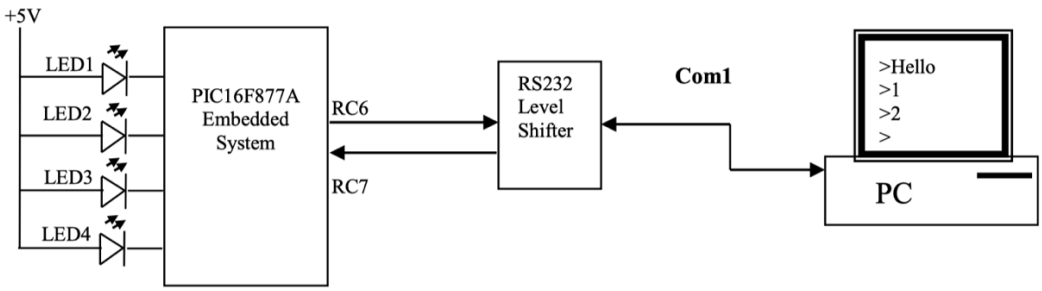

Figure 2 Basic Configuration

Software: MPLAB, WCHSerialPort (in macOS)
Equipment: PIC16F877A, PICkit-3, LED*4, PC.

Part 1

(a) Sketch the bit pattern for the asynchronous data packet for the character 'S' in Part 1 above. What is the duration of each bit?

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64 (X + 1))	Baud Rate = Fosc/(16 (X + 1))
1	(Synchronous) Baud Rate = Fosc/(4 (X + 1))	N/A

Legend: X = value in SPBRG (0 to 255)

Figure 3 Baud rate formula

As shown in the Figure 3, the duration of each bit can be calculated as follows:

$$X = \frac{F_{OSC}}{16 * \text{ Baud Rate}} - 1 = 25.041$$

(b) Using bit-banging send the character 'S' out on pin RC6. This should be done in pseudo code only. The following 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
define Xtal frequency 4Mhz for delay functions

main():
    setup()
    while (!TXIF) wait()
    TXREG = 0x53

setup():
    TRISC ← 0xC0
    TXSTA ← 0x24
    RCSTA ← 0x90
    SPBRG ← 0x19
```

Part 2

For Part 2 write a new program. Configure the USART for Asynchronous Transmit. Transmit the string “Hello” using the TXIF flag and the TXREG register.

The complete code of C program is shown in the following Table 1.

Table 1

The C program based on MPLAB

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


#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


void setup() {
    TRISC = 0xC0;
    TXSTA = 0x24;
```

```
    RCSTA = 0x90;
    SPBRG = 0x19;
}

void send(char* str) {
    for (int i = 0; str[i]; i++) {
        if (str[i] < 0 || str[i] >= 256) {
            continue;
        }
        while (!TXIF);
        TXREG = str[i];
    }
    while (!TXIF);
    TXREG = 0x0D;
}

void main() {
    setup();
    for (;;) {
        send("Hello");
        __delay_ms(1000);
    }
}
```

Part 3

For Part 3 write a new program. Configure the USART for Asynchronous Receive as well as transmit (i.e. set the CREN bit). Your program should receive characters from the PC (typed in Putty). Use the RCIF flag to determine when a character is received. A received character will appear in the RCREG register.

Keyboard Character	Description
1	Light LED1
2	Light LED2
0	Switch off all LEDs

Figure 3 Description of the keyboard char

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

Table 2

The C program based on MPLAB

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

```
#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 LED1 RD0
#define LED2 RD1
#define LED3 RD2
#define LED4 RD3

#define ON 0
#define OFF 1

void setup() {
    PORTD = 0xff;
    TRISD = 0x00;

    TRISC = 0xC0;
    TXSTA = 0x24;
    RCSTA = 0x90;
    SPBRG = 0x19;
}

void send(char* str) {
    for (int i = 0; str[i]; i++) {
        if (str[i] < 0 || str[i] >= 256) {
            continue;
        }
        while (!TXIF);
        TXREG = str[i];
    }
    while (!TXIF);
    TXREG = 0x0D;
    __delay_ms(1);
}

char recv() {
    while (!RCIF);
    char ret = RCREG;
    RCREG = 0;
    return ret;
}
```

```
void main() {
    setup();
    send("Ready");
    for (;;) {
        char op = recv();
        if (op == '1') {
            LED1 = ON;
        } else if (op == '2') {
            LED2 = ON;
        } else if (op == '3') {
            LED3 = ON;
        } else if (op == '4') {
            LED4 = ON;
        } else if (op == '0') {
            LED1 = LED2 = LED3 = LED4 = OFF;
        }
        __delay_ms(100);
    }
}
```

Summary for Lab 4

In Lab 4, we have learned the basic knowledge of USART and known how to utilize PIC DIP-40 to verify the function of our C programs. Thanks to Dr. Wu for her guidance.

Hanlin CAI 832002117

Qiguo QING 832002127

In 2022/11/30