LAB1: INTRODUCTION TO SOFTWARE TOOLS - MPLABX, PROTEUS AND CONTROLLING LEDS

1. Objectives:

- Provide an overview of the PIC18f4550 microcontroller
- To be familiar with some software tools like MPLABX and PROTEUS.
- To know how to make a project using MPLABX and then get the hex file of the software program of the project.
- How to simulate your program using PROTEUS..

2. Introduction

a. PIC18f4550

A micro controller unit (MCU) is an implementation of a computer on a single very large scale integrated circuit. It uses a microprocessor as its central processing unit (CPU) with additional peripheral components such as memory, timers, analog-to-digital converters, parallel I/O interface, asynchronous and synchronous serial communication interfaces. MCUs are used in almost every application that requires certain level of intelligence, such as controllers for printers, keyboards and home appliances.

Microchip's 8-bit PIC microcontroller is available in three product architecture categories, providing a variety of options for application requirement as needed by the user. The specific families include:

· Baseline – PIC10F and some PIC12F and PIC16F;
· Mid-Range– PIC12F and PIC16F; and
· High Performance – PIC18F with J and K-Series.

The PIC18 family utilizes a 16-bit program word architecture and incorporates an advanced RISC architecture with 32 level-deep stack, 8x8 hardware multiplier, and multiple internal and external interrupts.

Table 1.1 below provides a summary of the features available for the PIC18F4550 chip. The pin-outs configuration of the PIC18F4550 is as shown in Figure 1.2.

table

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EAUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I^2C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

Table 1.1: PIC18F4550

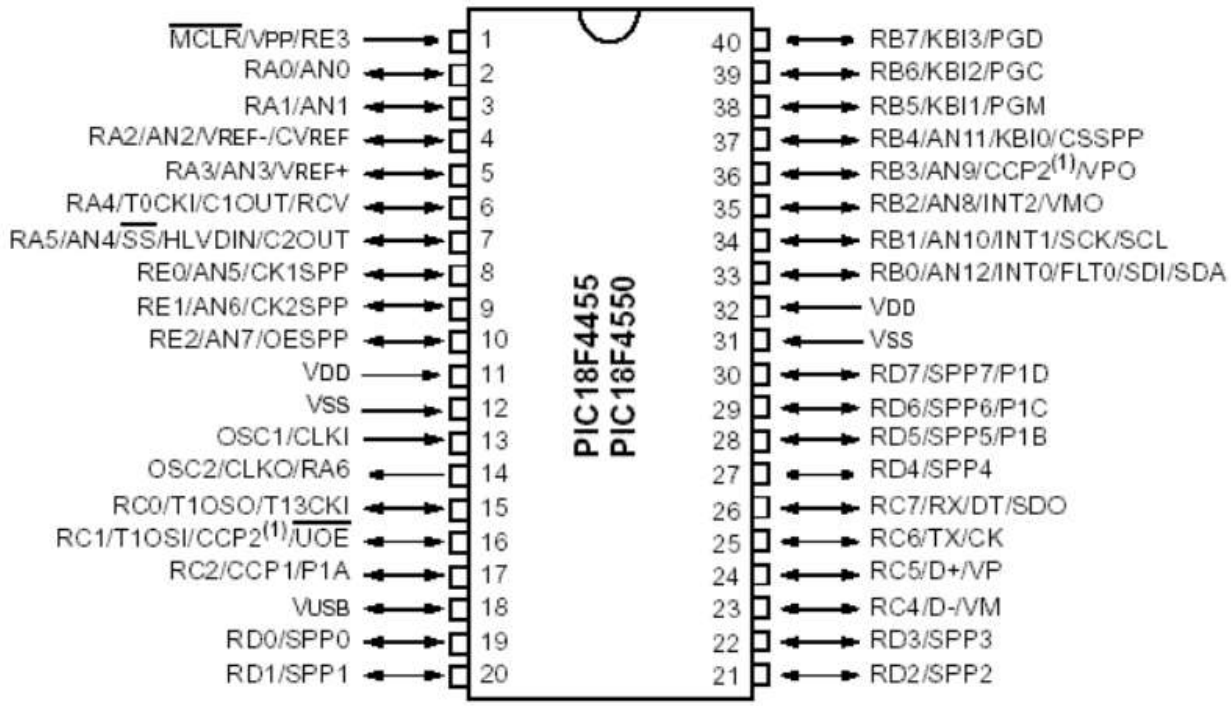


Figure 1.2: Pin Diagram

The PIC18 MCUs provide the following peripheral functions:

- High Performance RISC CPU.
- Operating Frequency: DC – 48 MHz.
- Linear program memory addressing to 32 Kbytes.
- Can store up to 16,384 single-word instructions.
- Linear data memory addressing to 2 Kbytes.
- 256 bytes EEPROM memory.
- Five bidirectional I/O Ports: A, B, C, D, E.
- Three independent Timers/Counters & Watch-dog timer.
- Serial Communications:
- Enhanced USART Module, Supports RS-485, RS-232.
- Master Synchronous Serial Port (MSSP): supports SPI and I2C mode.
- Streaming Parallel Port (SPP).
- Incorporate a fully featured Universal Serial Bus communications module that is compliant with the USB Specification Revision 2.0.
- Compatible 10-bit Analog-to-Digital Converter: 13 input channels.

b. MPLABX IDE

MPLABX IDE is a software program that runs on a PC to develop applications for Microchip microcontrollers. It is called an Integrated Development Environment (IDE), because it provides a single integrated "environment" to develop code for embedded microcontrollers.

The Integrated Development Environment allows the embedded systems design engineer to progress through the design cycle without the distraction of switching among

an array of tools. By using MPLABX IDE, all the functions are integrated, allowing the engineer to concentrate on completing the application without the interruption of separate tools and different modes of operation.

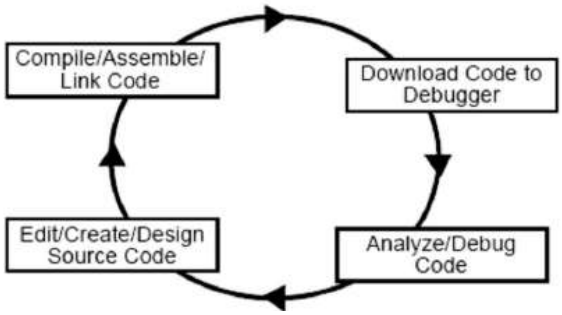


Figure 1.3: Design Cycle

Let's start with MPLABX IDE.

1. Download compiler XC8 and Install it.

Here: http://www.microchip.com/mplab/compilers

2. Download MPLABX IDE and install it also.

Here: http://www.microchip.com/mplab/mplab-x-ide

Procedure in using MPLABXC8 IDE: For this lab preview, your lab instructor will guide you step by step to create the source file to be put into a project.

```
int main(void) {
    TRISD0 =0;
    while(1){
        RD0 =1;
        __delay_ms(100);
        RD0 =0;
        __delay_ms(100)
    }
    return 0;
```

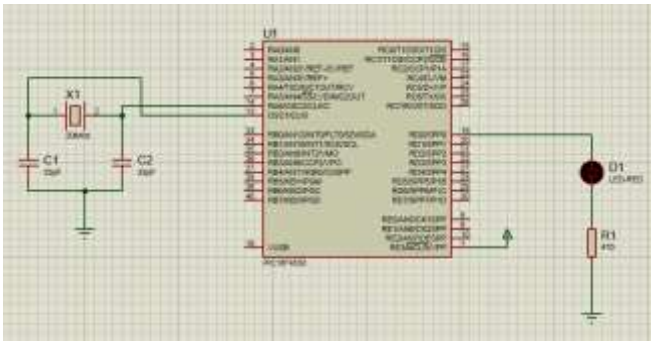
c. Proteus

Proteus contains everything you need to develop; test and virtually prototype your embedded system designs based around the Microchip Technologies™ PIC18 series of microcontrollers. The unique nature of schematic based microcontroller simulation with Proteus facilitates rapid, flexible and parallel development of both the system hardware and the system firmware. This design synergy allows engineers to evolve their projects more quickly, empowering them with the flexibility to make hardware or firmware changes at will and reducing the time to market.

Proteus VSM models will fundamentally work with the exact same HEX file as you would program the physical device with, produced by any assembler or compiler.

Procedure in using PROTEUS:

- Go to Start>All Programs>Proteus 7 Professional>ISIS 7 Professional.
- For this lab preview, your lab instructor will guide you step by step to design the circuit below.



- To run the simulation, you need to load the hex file first by right-clicking on the microcontroller and then click edit properties.
- At the Edit Component menu:
    - Under the Program File tab, select your hex file by clicking on the browse icon.
    - Under the Processor Clock Frequency, select your desired clock frequency.
    - Under Advanced Properties, select Enable Watchdog Timer and then select No.
    - Finally, click OK.
- Finally, click on the Play button at the bottom of the workspace to run the simulation.

3. **Lab exercise**

a. **Lab 1a.**

- Connect the circuit using Proteus with PIC18F4550 and 8 LEDs on PORTD.
- Write an C language program that sent 0x8a to port D.
- Load the (.hex) file to PIC18F4550 microcontroller.
- Configure PIC18f4550 parameters needed for the simulation by putting Processor clock frequency 20MHz.
- Simulate the circuit using Proteus ISIS program.

b. **Lab 1b.** Write an C program to control 8 LEDs on PORTD with a specified pattern. They blink alternate LEDs with a sequence of one, two and three respectively. Use MPLABX to write our code and Proteus to show the results.

Name: Student Code: Class: Lab:

1. Circuit
2. Algorithm flowchart
3. Code and explanation
4. Summary