

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



THÍ NGHIỆM
TIN HỌC VİP

BM Kỹ thuật Máy tính

2017

Bài 1 : Giới thiệu MPLAB IDE và KIT PIC

Chuẩn bị :

- Đọc tài liệu **MPLAB X IDE User's guide**.
- Đọc tài liệu **XC8 User's guide**.
- Xem **PICKit3 poster**.
- Xem phần I/O ports của MCU PIC18F8722.

Nội dung :

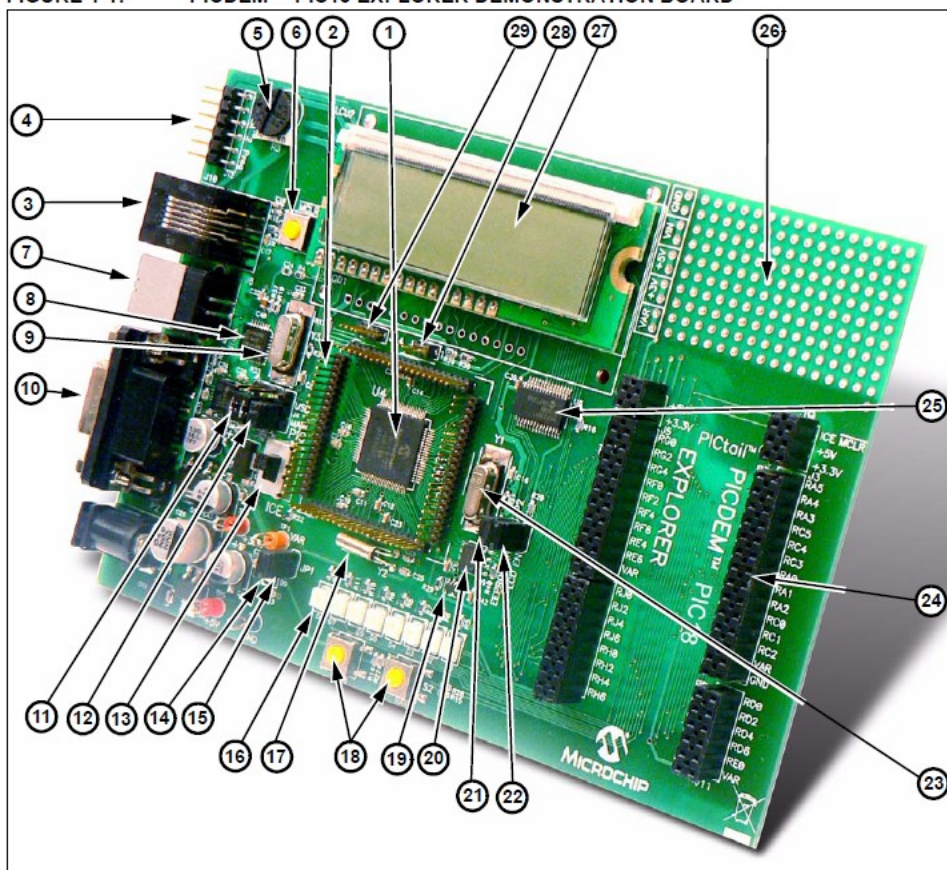
- Tạo project trên MPLAB X IDE.
- Viết chương trình trên XC8.
- Dịch và nạp chương trình vào vi điều khiển PIC.
- Chạy và gỡ rối chương trình.

1.1 Phần cứng thí nghiệm Pickit3 và PICDEM PIC18.

Bộ hỗ trợ lập trình dùng với máy tính PICKit™3.

Kit thí nghiệm PICDEM PIC18 có các đặc điểm như hình sau :

FIGURE 1-1: PICDEM™ PIC18 EXPLORER DEMONSTRATION BOARD



1. Vi điều khiển PIC18F8722.
2. Chân cắm vi điều khiển thứ hai (Plug-In Modules PIMs).
3. Đầu kết nối với bộ lập trình ICD.
4. Đầu kết nối với bộ lập trình PICKit™.
5. Biến trở 10KΩ dùng cho tín hiệu nhập tương tự.
6. Nút nhấn reset.
7. Đầu cắm USB dùng cho chuẩn giao tiếp RS-232.



8. PIC18LF2450 - dùng để đổi giao tiếp RS-232 ra USB nối với PC.
9. Thạch anh 12MHz dùng cho PIC18F2450.
10. Đầu giao tiếp RS-232 DB9.
11. Jumper J13 chọn đường giao tiếp RS-232 thông qua USB hay DB9.
12. Jumper J4 chọn lập trình PIC18F2450 hay PIC chính trên mạch.
13. Chuyển mạch S4 chọn PIC18F8722 hay PIM.
14. LED nguồn.
15. Jumper J1 dùng để ngắt nguồn 8 LED chỉ thị PORTD.
16. Tám LED chỉ thị cho PORTD.
17. Thạch anh 32.768 kHz dùng cho bộ tạo xung clock cho Timer1.
18. Hai nút nhấn tạo tín hiệu nhập.
19. Cảm biến nhiệt MPC9701A.
20. 25LC256 SPI EEPROM.
21. Jumper J2 cho phép EEPROM.
22. Jumper J3 cho phép LCD.
23. Thạch anh 10 MHz dùng cho PIC chính.
24. PICtail™ daughter board connector socket.
25. Bộ mở rộng giao tiếp SPI dùng cho màn hình LCD ngoài MCP23S17.
26. Vùng lắp thêm linh kiện.
27. Màn hình LCD.
28. J2, 3 chân, dùng để chọn nguồn 3.3V hay 5V.
29. J14, 4 chân, dùng với PIM nếu cần để lấy 3.3V, 5V, V_{IN} và MCLR.

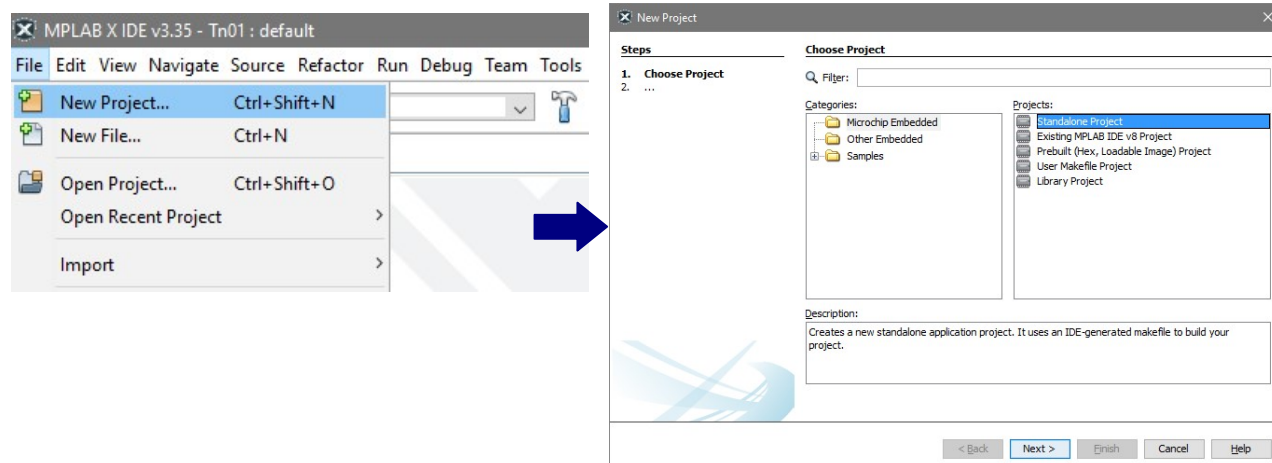
1.2 Môi trường phát triển MPLAB X IDE

Bước 1. Kết nối bộ thí nghiệm như sau:

Bộ nguồn → Bo PICDEM PIC18 (xanh lá) → PICKit3 (đỏ) → dây cáp USB → PC.

Bước 2. Chạy phần mềm MPLAB X Vx.xx.

Bước 3. Chọn Menu : **File → New Project ...**

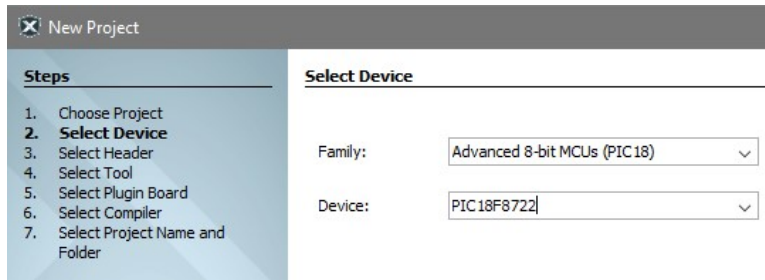


Chọn **Microchip Embedded / Standalone Project** → **Next** ở cửa sổ New Project.

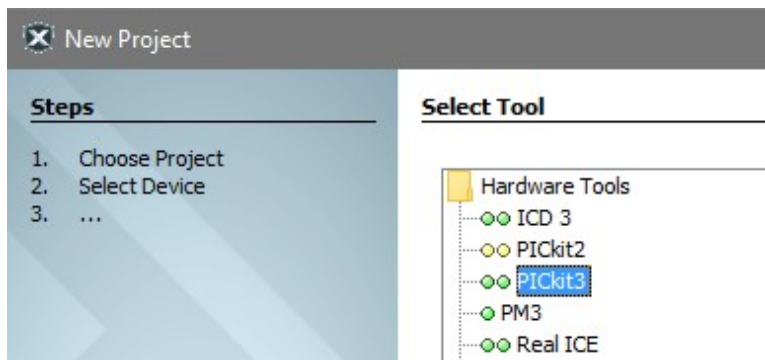
Bước 4. Chọn tiếp

Family: **Advanced 8-bit MCUs (PIC18)**

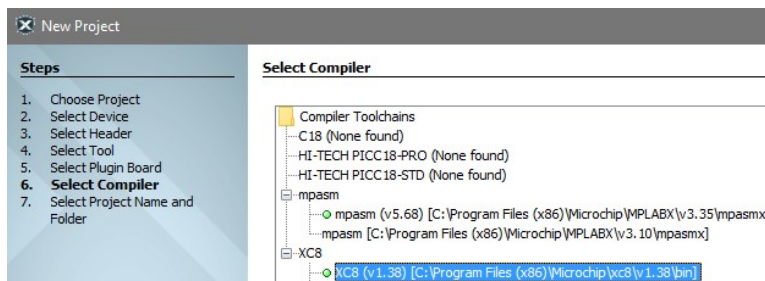
Device: **PIC18F8722**



Bước 5. Chọn công cụ lập trình **PICKit3**.



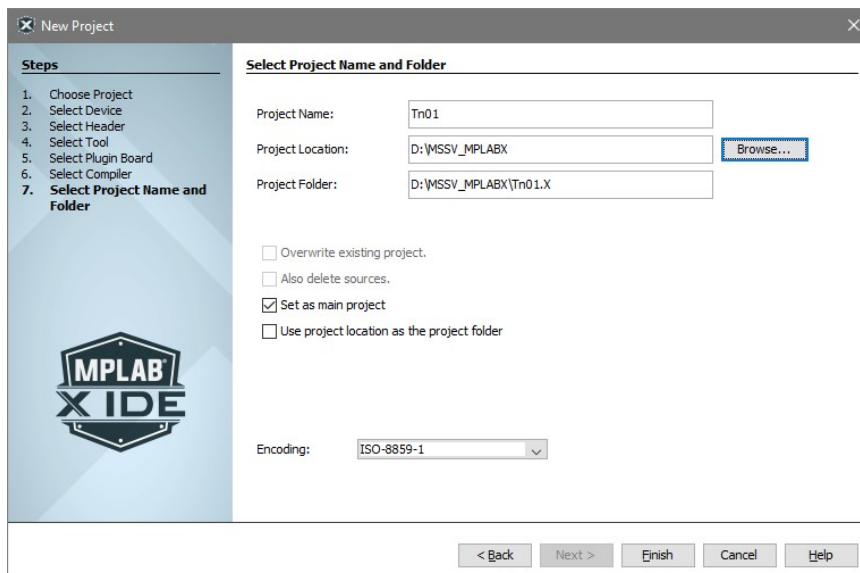
Bước 6. Chọn trình biên dịch **XC8**.



Bước 7. Chọn thư mục lưu dự án.

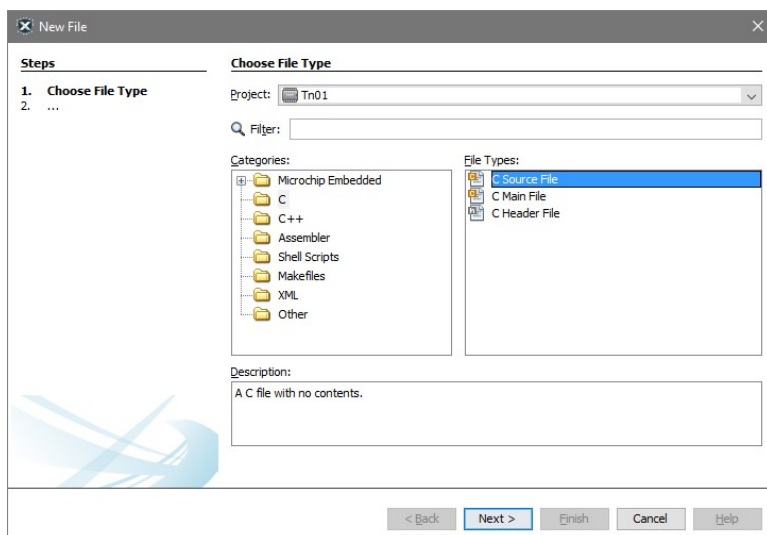
Sinh viên nên chọn ổ đĩa dữ liệu (thường là ổ D:), tạo thư mục có tên là mã số sinh viên và vào trong tạo thư mục con tên Tn01 cho bài thí nghiệm 1 (các bài 2, 3, ... sau này cũng sẽ lưu vào thư mục Tn02, Tn03, ...).

Việc tạo thư mục con như trên giúp sinh viên dễ dàng sao chép dự án ra USB drive lúc kết thúc buổi thí nghiệm hoặc buổi thi.



Bước 8. Thêm Source file vào dự án:

File → New File → C → C Source File



Bước 9. Trong cửa sổ Tn01.c, đánh đoạn code sau đây vào để hoàn tất dự án:

```
/* File:   Tn01.c
 * Author:
 * Subject: Cai dat, lam quen IDE MPLABX, XC8 va IO port
 * Created on
 */
#include <xc.h>
#pragma config OSC=HS, WDT=OFF, LVP=OFF
// Che do dao dong High Speed, tan so = 10 MHz


volatile unsigned char led @ 0xF8C; //LATD

void init()
{
    ADCON1=0x0F;    //Khong dung Analog
    TRISD=0;        //Cong D xuat 8 bit (LED x 8)
    led=0x00;
    TRISAbits.RA5=1;    //bit 5 cong A nhap (nut RA5)
}

void xuly()
{
    unsigned char tam;
    if (PORTAbits.RA5==0)
    {
        led++;
        while(PORTAbits.RA5==0);
    }
}

void main()
{
    init();
    while(1)
    {
        xuly();
    }
}
```

1.3 Dịch chương trình và nạp code vào vi điều khiển PIC

Bước 10. Để dịch chương trình, ta dùng chức năng: **Run → Build Project** hoặc bấm chuột vào icon  trên thanh công cụ.


Nếu việc dịch thất bại (BUILD FAILED), ta cần tìm dòng báo lỗi để sửa.

```
"C:\Program Files (x86)\Microchip\xc8\v1.38\bin\xc8.exe" --pass1 --chip=18F8722 --
Tn01.c:15: error: (221) hexadecimal digit expected
Tn01.c:15: error: (195) expression syntax
```

Việc tìm lỗi (nhấn chuột tại dòng có chữ **error** để chuyển đến vị trí lỗi), sửa lỗi và dịch lại được làm cho đến khi dịch thành công (BUILD SUCCESSFUL).


```
make[2]: Leaving directory 'D:/MSSV/MPLABX/Tn01.X'
make[1]: Leaving directory 'D:/MSSV/MPLABX/Tn01.X'

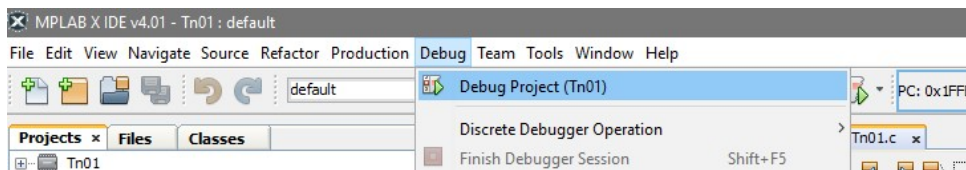
BUILD SUCCESSFUL (total time: 2s)
Loading code from D:/MSSV/MPLABX/Tn01.X/dist/default/production/Tn01.X.production.hex...
Loading completed
```


Bước 11. Nạp code ra vi điều khiển dùng icon .

1.4 Debug dùng MpLab SIM

Bước 1. Chọn Debugger PICKit3 hoặc Simulator trong **Project Properties**.

Bước 2. Bắt đầu debug bằng menu hoặc icon  trên thanh công cụ.



Các chức năng thường dùng trong debug:

- Pause : tạm ngưng tiến trình debug.
- Reset: trở về đầu chương trình.
- Step Into (F7): chạy từng bước, vào trong chương trình con (nếu gặp).
- Step Over (F8): chạy từng bước, gọi chương trình con cũng xem như 1 bước.
- Continue (F5): tiếp tục chạy sau khi đặt các **breakpoint** để bẫy.
- Run (F9): chạy chương trình, chương trình sẽ chạy liên tục đến khi nào có breakpoint thì dừng.
- Breakpoints (F2): tạo ra breakpoint tại vị trí hiện tại của con trỏ (cũng có thể "double click" vào hàng code mình mong muốn đặt Breakpoint).
- Run to cursor (F4): chạy đến vị trí điểm nhảy.

Quá trình debug đòi hỏi sự nhẫn nại, vừa chạy từng bước, từng lệnh, hoặc từng đoạn lệnh vừa kết hợp với cửa sổ **Watch** xem chuyển biến nội dung các biến hay biểu thức để tìm ra nguyên nhân làm chương trình chạy sai.

1.5 Bài tập

- a. Cài tiến chương trình Tn01.c để khi nhấn nút RA5 sẽ thấy 8 LED PORTD sáng tắt luân phiên (toggle).
- b. Cài tiến chương trình Tn01.c để khi nhấn nút RA5 sẽ thấy 2 LED PORTD sáng và chạy từ trái sang phải.