

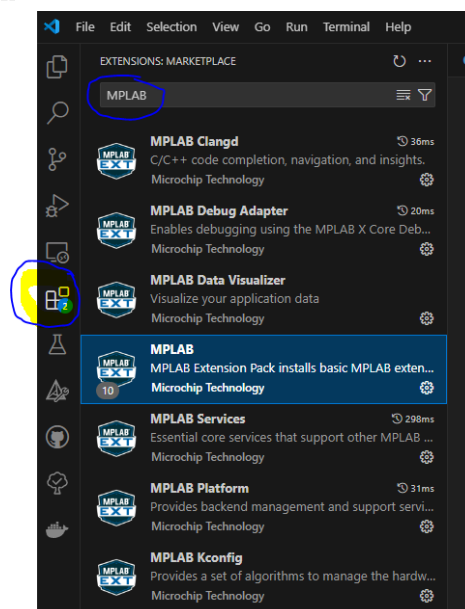


1, Giới thiệu

- Để lập trình C cho vi điều khiển, cần có:
 - Trình soạn thảo để viết Code
 - Compiler (hay Toolchains) để link, biên dịch code.
 - Programmer để nạp file đã dịch xuống vi điều khiển.
- Ngoài ra, còn có các công cụ khác giúp việc lập trình dễ dàng hơn (Debug, Nhắc lệnh,...). Tất cả tập hợp thành IDE.
- MPLAB X IDE là một IDE phát triển bởi Microchip. Tuy mạnh mẽ nhưng khá cồng kềnh.
- VSCode không hẳn là IDE nhưng cũng không hẳn là trình soạn thảo. Nó tích hợp với các extension để hỗ trợ việc lập trình, và kết nối với các compiler cho nhiều ngôn ngữ, nhiều mục đích lập trình khác nhau.
- VSCode đơn giản, nhẹ, dễ nhìn, mang phong cách hiện đại. Có nhiều extension, được sự hỗ trợ từ cộng đồng.
- Tuy nhiên, VSCode không phải là IDE cho một ngôn ngữ/ mục đích cụ thể nên cần có kiến thức nhất định để thiết lập và sử dụng, đôi khi khó khăn hơn các IDE truyền thống.

2, Cài đặt và sử dụng

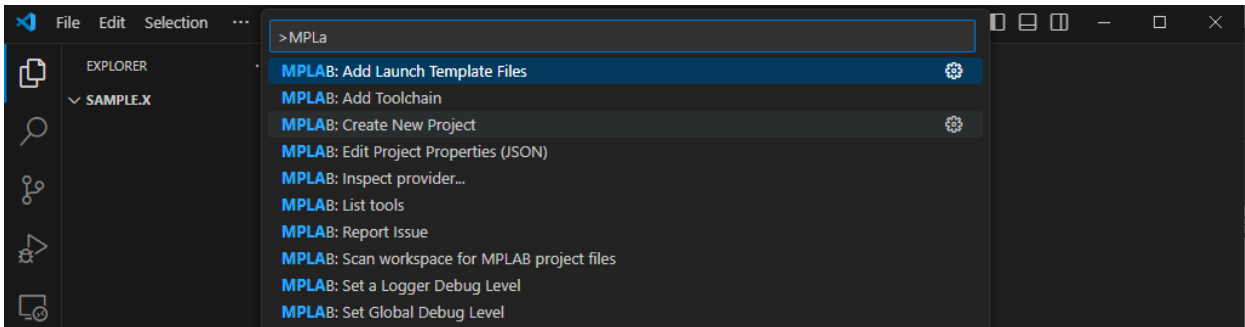
- Bước 1. Cài đặt Visual Studio Code.
- Bước 2. Vào extension, tìm các extension sau và cài đặt:
 - MPLAB
 - MPLAB Extension Pack



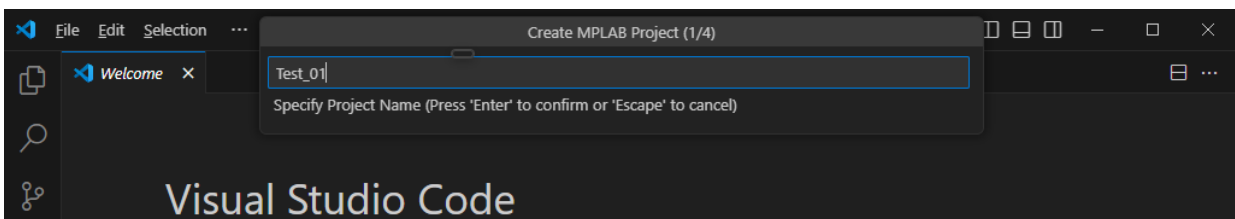
Các extension liên quan sẽ được tự động cài đặt.

Tutorial 1: Cài đặt môi trường làm việc với PIC

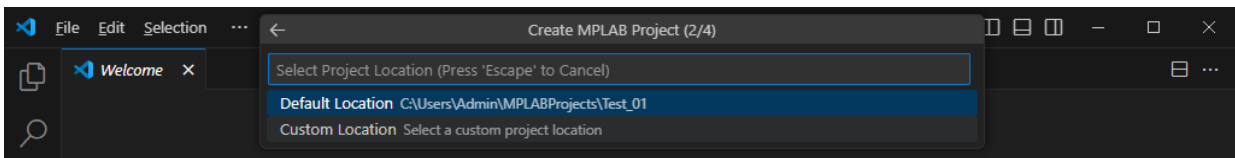
- Bước 3. Tạo project mới:
 - Dùng tổ hợp Ctrl+Shift+P
 - Gõ MPLab, chọn New Project. Làm theo hướng dẫn



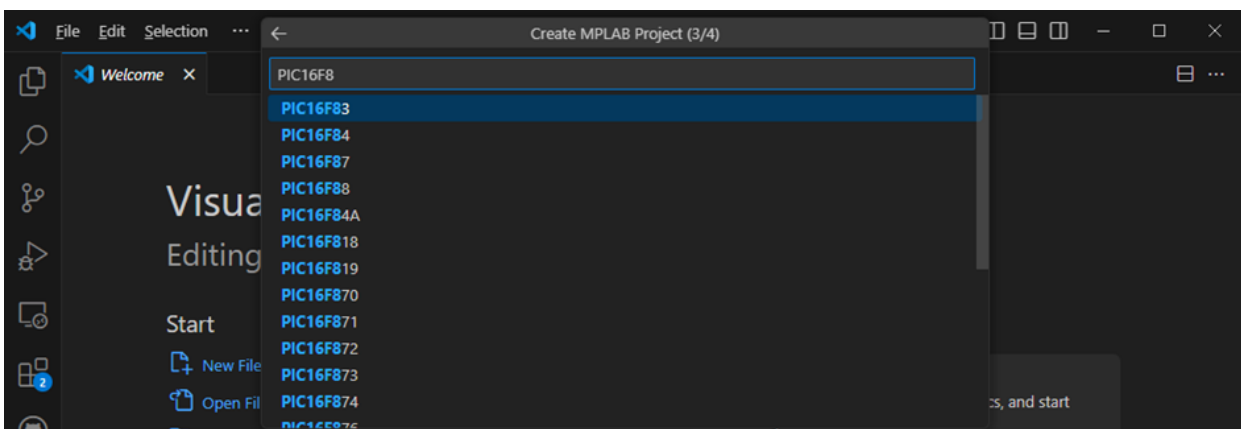
Đặt tên



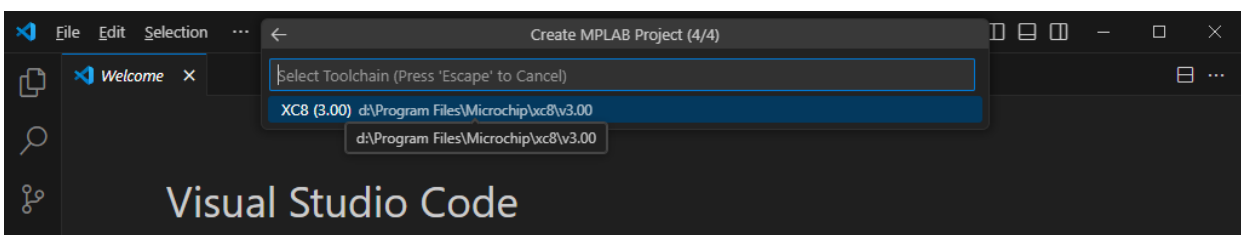
Chọn nơi lưu



Chọn dòng PIC

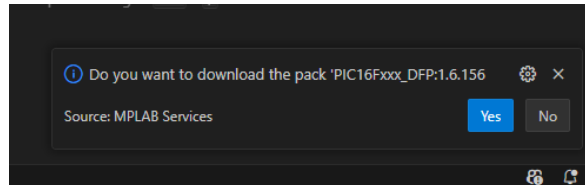


Chọn Toolchain



Project sẽ được tự động sinh ra.

- Bước 4. Cài đặt lại Toolchain.
 - Trường hợp VSCode không nhận được toolchain mặc dù đã cài đặt, có thể do toolchain chưa được đưa vào Path Environment của Window.
 - Khi đó, dùng Ctrl+Shift+P, gõ MPLAB, chọn Select Toolchain và trở vào thư mục chứa toolchain xc8.
 - Nếu đã cài toolchain mà VSCode vẫn yêu cầu cài lại: Không cần thiết phải cài.
 - Lần đầu tiên lập trình, có thể VSCode yêu cầu cài đặt Package, đồng ý.



- Bước 5. Lập trình
 - Tạo các file và lập trình như bình thường.
 - Ví dụ, đoạn code sau đây trong file main dùng để Toggle LED ở chân RA0.

```
// CONFIG1
#pragma config FOSC = HS      // Oscillator Selection bits (HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN)
#pragma config WDTE = OFF     // Watchdog Timer Enable bit (WDT disabled and can be enabled by SWDTEN bit of the WDTCON register)
#pragma config PWRT = OFF     // Power-up Timer Enable bit (PWRT disabled)
#pragma config MCLRE = OFF    // RE3/MCLR pin function select bit (RE3/MCLR pin function is digital input, MCLR internally tied to VDD)
#pragma config CP = OFF      // Code Protection bit (Program memory code protection is disabled)
#pragma config CPD = OFF     // Data Code Protection bit (Data memory code protection is disabled)
#pragma config BOREN = ON     // Brown Out Reset Selection bits (BOR enabled)
#pragma config IESO = OFF    // Internal External Switchover bit (Internal/External Switchover mode is disabled)
#pragma config FCEN = OFF    // Fail-Safe Clock Monitor Enabled bit (Fail-Safe Clock Monitor is disabled)
#pragma config LVP = OFF     // Low Voltage Programming Enable bit (RB3 pin has digital I/O, HV on MCLR must be used for programming)

#define _XTAL_FREQ 4000000 // Tần số thạch anh: 4 MHz

#include <xc.h>
#include <stdint.h>

void main(void) {

    ANS0 = 0;
    TRISA0 = 0;
    RA0 = 1;

    while(1) { // blinking
        PORTAbits.RA0 ^= 1; //LED Toggle
        __delay_ms(100);
    }
}
```

Để dịch:

Dùng Ctrl+Shift+P, gõ MPLAB, chọn Build.

Khi đó, trong thư mục out, sẽ tạo ra 1 file default.elf.

3. Một số lưu ý

3.1. Cần file .hex để nạp vào Proteus.

Trong trường hợp đó, ta sẽ can thiệp vào lệnh dịch và các chỉ lệnh của VSCode như sau.

- Trong cmake, tìm rule.cmake. Thay tất cả các cụm ‘-mno-download’ thành ‘-mdownload’

Tutorial 1: Cài đặt môi trường làm việc với PIC

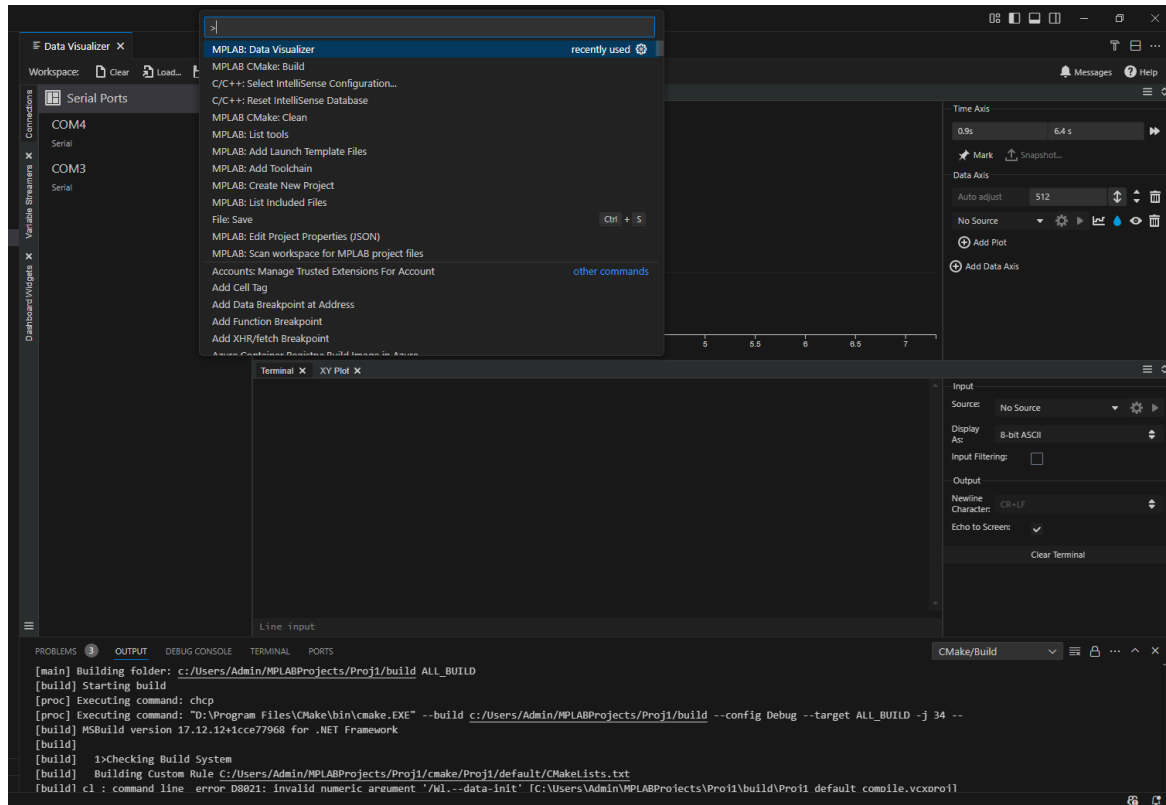
- Trong cmake, tìm file.cmake. Thay tất cả các đuôi .elf thành .hex.
- Trong các file .json, thay tất cả các đuôi .elf thành .hex.
- Khi đó, khi biên dịch chương trình sẽ tạo ra file default.hex trong folder out. File này có thể được sử dụng để nạp vào Proteus và nạp vào VDK dùng Pickit3/ BurnE.

3.2. Sử dụng Visualizer

Visualizer là công cụ của MPLab để kết nối VDK qua cổng COM và đọc dữ liệu.

Có thể sử dụng Visualizer trong VSCode bằng cách:

Dùng Ctrl+Shift+P. Gõ Visualizer. Chọn và sử dụng.



3.3. Để sử dụng trực tiếp PICKit3 với VSCode.

- SV tìm hiểu, sửa các file .json