**Trương Đức Thắng**

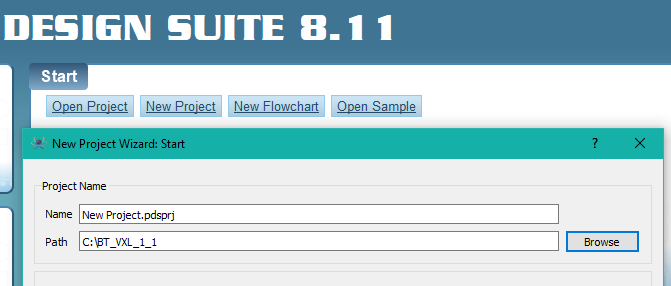
**20145196**

**Assignment 1.1: Blinking LED with MPLABX and XC8**

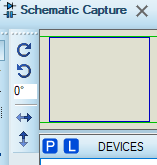
**Đầu tiên, Tạo 1 thư mục có tên: BT\_VXL\_1\_1**

****

**Mở Proteus lên >> Click vào New project >> Chọn đường dẫn lưu vào thư mục “BT\_VXL\_1\_1”**



**Click vào Thư viện có hình chữ “P” để chọn các linh kiện**

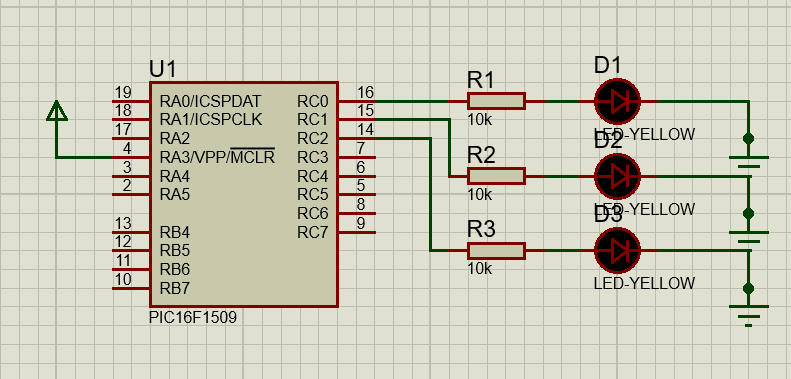
****

**Sau đó, chúng ta sẽ lấy các linh kiện có tên như hình**

Ảnh có chứa văn bản

Mô tả được tạo tự động

**Sau đó, chúng ta sẽ lấy các linh kiện lắp mạch như hình dưới:**

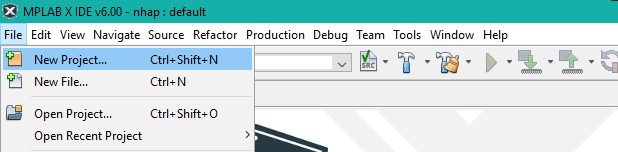
****

**Tiếp theo là chúng ta sẽ phần mềm để code cho vi điều khiển là MPLAB lên**

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

**Chọn vào file >> new project. ở phía trên cùng góc bên trái**

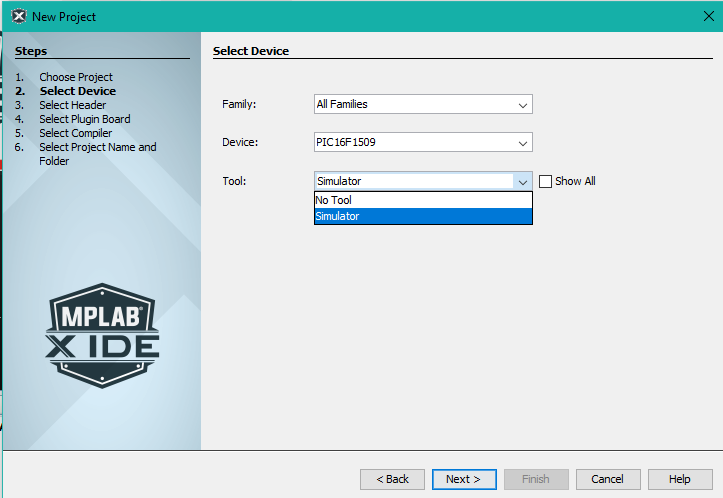
****

**Chọn Microchip Embedded >> Standalone project >>Next**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Tiếp tục: Family: chọn All families => Device : pic16f1509 => Tool: Simulator. Như hình dưới:**

****

**Tiếp tục chọn NONE**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Tiếp tục, chọn trình biên dịch XC8 đã tải lúc trước**

**Ảnh có chứa văn bản

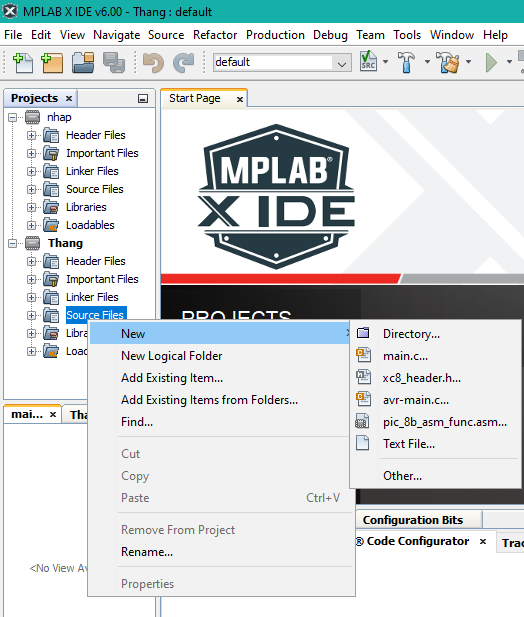
Mô tả được tạo tự động**

**Tiếp theo là đặt tên >> Finish**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Chọn vào thư mục Thang >> Nhấp chuột trái vào source file >> new >> main.c**

****

**Tiếp tục bấm finish để tạo và hiện ra trang main.c**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Tiếp theo, chúng ta cấu hình cho pic16f1509. Click vào WINDOW >> Target Memory View >> Configuration Bits**

**Ảnh có chứa văn bản

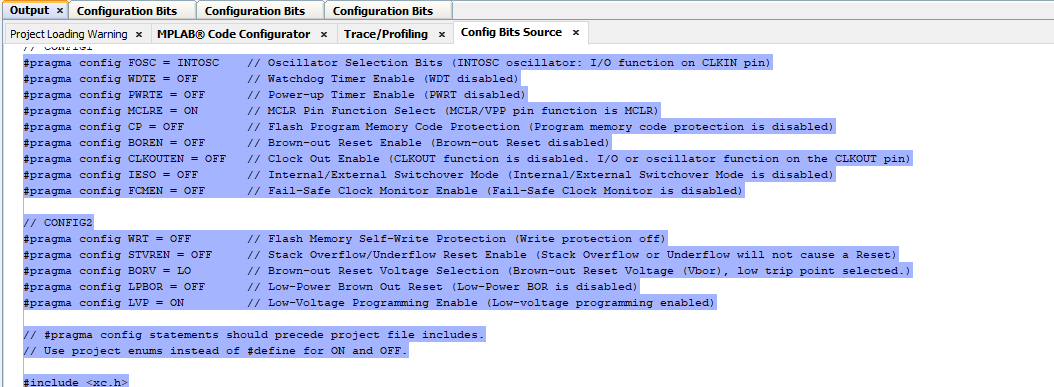
Mô tả được tạo tự động**

**Sau đó chúng ta cấu hình cho pic như hình dưới, và click vào generate**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Sau khi generate, sẽ hiện ra các lệnh khai báo, chúng ta copy đem phía trên cùng của code để khai báo**

****

**Ảnh có chứa văn bản

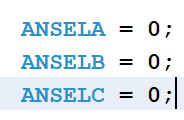
Mô tả được tạo tự động**

**Tiếp theo, chúng ta dùng hàm TRISbit để cấu hình cho các chân C0, C1, C2 là cổng ra (output)**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Tiếp theo là chúng ta dùng ANSEL để chuyển chân Analog sang output**

****

**Ta dùng hàm LAT để hiển thị đèn sáng (mức 1) hay tắt (mức 0)**

**Ban đầu tất cả các đèn đều tắt nên:**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Để dùng được hàm delay thì trước tiên ta phải khai báo hàm #define \_XTAL\_FREQ 1000000 mới có thể sử dụng được**

**Ở đây, chúng ta truyền vào 1 số, 100 tương ứng với ngưng trong 0.1s để chúng ta có thể thấy được đèn tắt**

**Tiếp theo chúng ta sẽ cho đèn 1 sáng, đèn 2,3 tắt. Thì ta làm như sau:**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Vì đèn 1 sáng nên dùng hàm LAT để hiển thị chân C0 ở mức logic 1, 2 đèn còn lại tắt nên ở mức logic 0. Và dùng hàm delay để ngưng 1 khoảng thời gian để thấy được đèn sàng**

**Tương tự với các trường hợp còn lạ:**

**Đèn 1, 3 sáng => logic 1, đèn 2: tắt => logic 0;**

**Đèn 1,2 tắt => logic 0, đèn 3: sáng => logic 1**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Sau đó chúng ta, build and run sẽ hiện chữ sucessful**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Tiếp tục chúng ta sẽ nhấp chuột vào pic16f1509 trong protues nạp theo đường dẫn: ..\BT\_1\_1\Thang.X\dist\default\production\Thang.X.production.hex**

**>> OK**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

**Kết quả** Ảnh có chứa văn bản, thiết bị điện tử, màn hình, ảnh chụp màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, thiết bị điện tử, màn hình, ảnh chụp màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, thiết bị điện tử, màn hình, ảnh chụp màn hình

Mô tả được tạo tự động

XIN CHÂN THÀNH CẢM ƠN THẦY ĐÃ ĐỌC BÀI CỦA EM