

## **THÍ NGHIỆM HỆ NHÚNG (IT4210)**

IT4210 (3-0-1-6)

Thời lượng: 15 tiết/ 1 nhóm

Các bài thí nghiệm:

- Bài 1. Lập trình vi điều khiển 8 bit

## Bài thí nghiệm 1

### LẬP TRÌNH VI ĐIỀU KHIỂN

#### 1. Mục đích

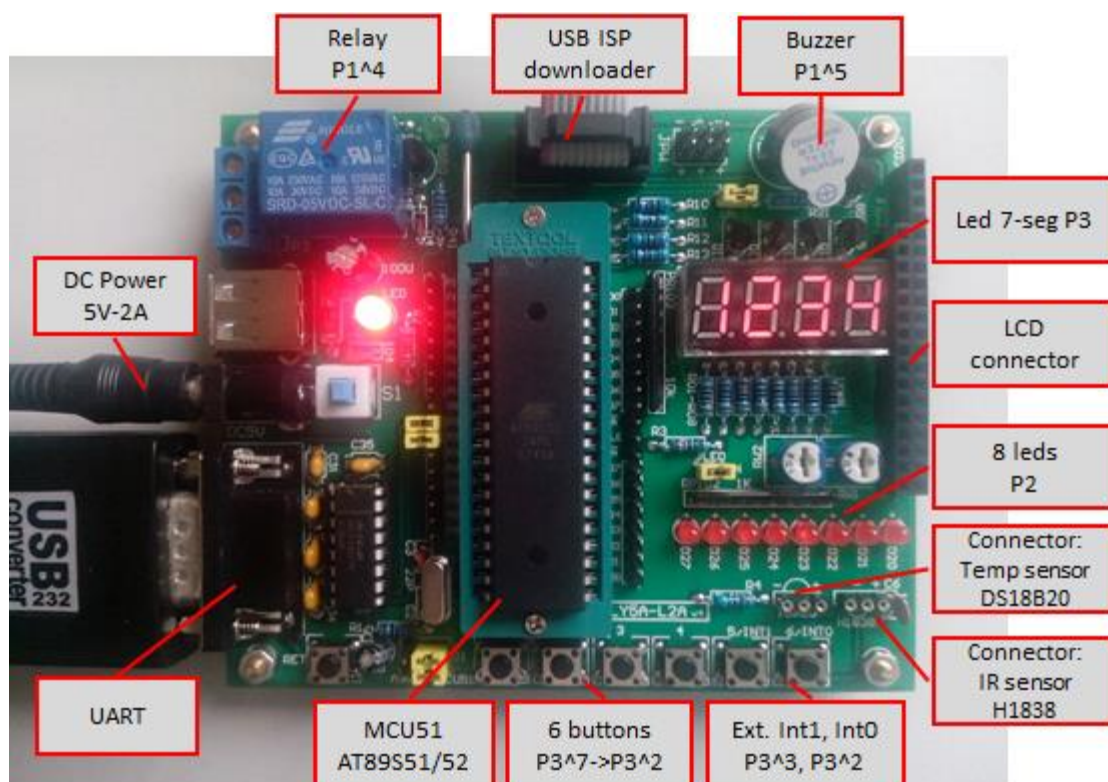
- Nắm được kỹ năng cơ bản về lập trình cho vi điều khiển, minh họa trên vi điều khiển họ 8051
- Thực hành các bài tập lập trình cho vi điều khiển sử dụng KIT 8051
- Thực hiện được các giao tiếp cơ bản của vi điều khiển: Led đơn, Led 7 thanh, nút bấm, Rơ le, UART, Buzzer, ...

#### 2. Chuẩn bị

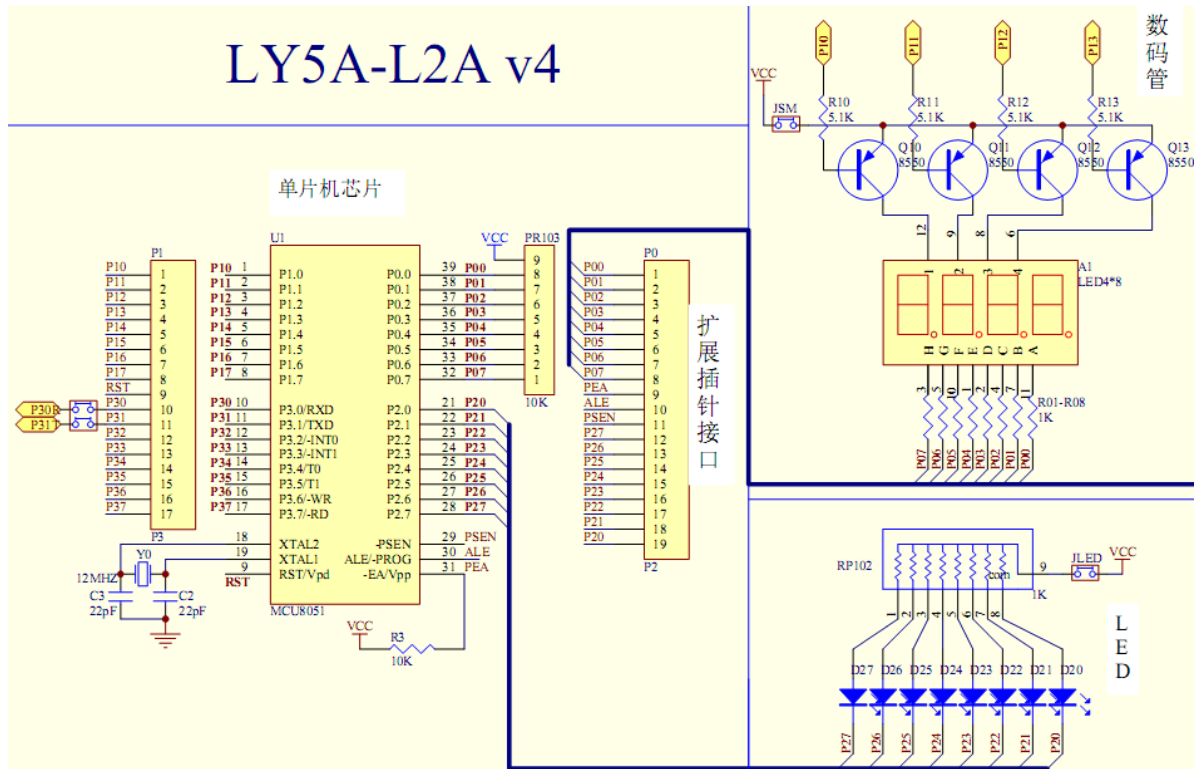
- Bộ KIT 8051 (vi điều khiển: STC89C51/STC89C52, hoặc AT89C51/AT89C52, hoặc AT89S51/AT89S52)
- Các phụ kiện: Adapter 5V-2A, RS232 cable/USB-COM convertor, USB cable, Mạch nạp (USB ISP)
- Phần mềm: Công cụ Keil C51 (lập trình họ 8051), phần mềm nạp IC Prog (nạp qua mạch nạp USB ISP cho AT89Sxx + AVR)

#### 3. Tiến hành

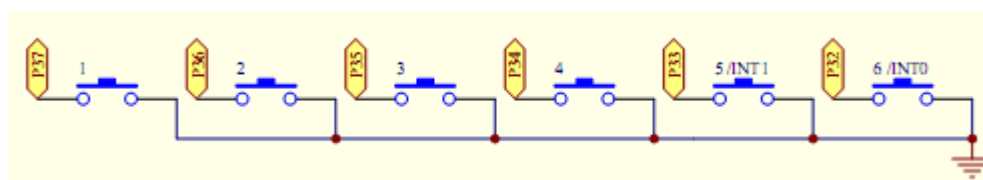
##### 3.1. Tìm hiểu về KIT 8051



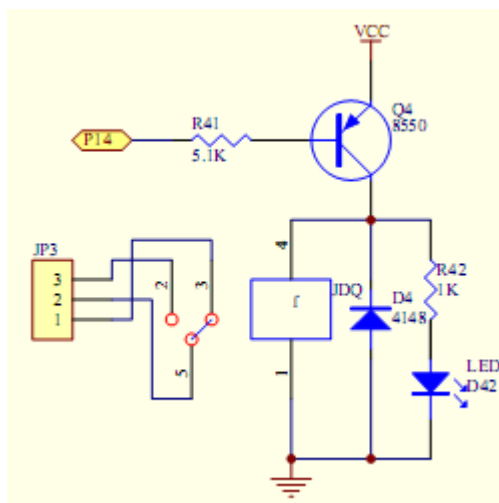
- Quan sát KIT, tìm hiểu các thành phần như mô tả trên hình vẽ.
- Tìm hiểu file schematics (đi kèm). Các thành phần chính:



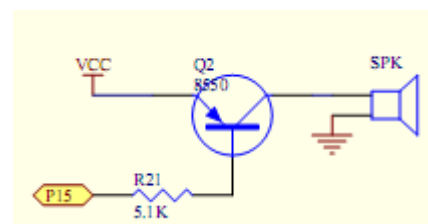
MCU8051 và 8 Led on Port P2, 4 Led 7 segments on Port P0



6 buttons on Port P3



Rơ-le on P1^4

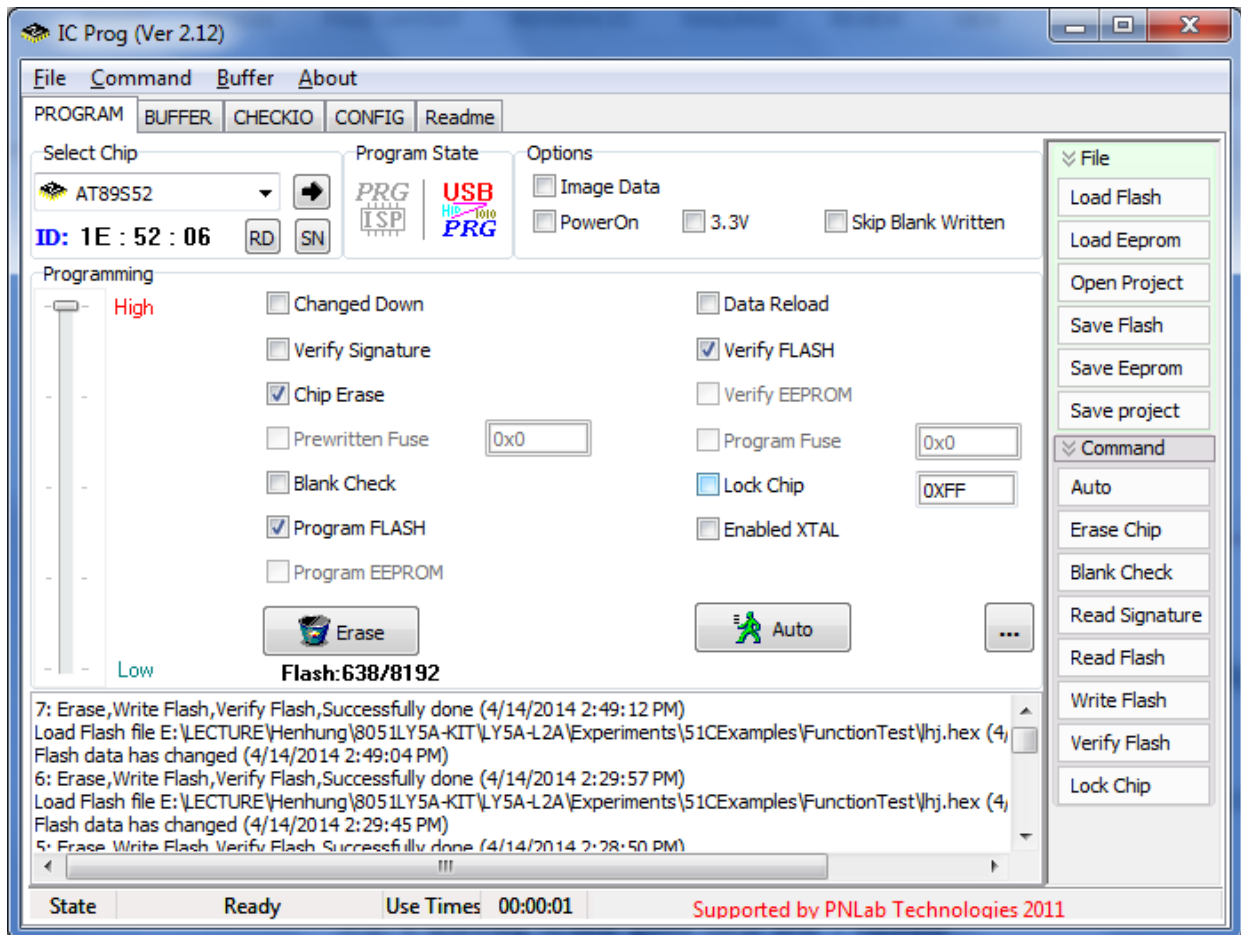


Buzzer on P1^5

### 3.2. Chạy chương trình Function Test

**Bước 1.** Sử dụng mã nguồn chương trình mẫu FunctionTest cung cấp. Dùng Keil C51 để biên dịch tạo file .hex tương ứng.

**Bước 2.** Sử dụng phần mềm nạp chip IC Prog để nạp chương trình vào bộ nhớ flash của vi điều khiển trên KIT. (Hình minh họa)



(Chú ý: Nếu không có thạch anh => Không nạp được firmware, Lỗi Chip Enable Error)

**Bước 3.** Sau khi nạp chương trình thành công, thực thi chương trình và quan sát kết quả:

- + Bật 8 led đơn, bật còi (buzzer), đóng/cắt rơ le
- + Hiển thị 4 số led 7 thanh
- + Bấm nút bấm để tăng giảm giá trị số hiển thị led 7 thanh.

**Bước 4.** Đọc hiểu mã nguồn chương trình mẫu đã cho.

### 3.3. Bài tập tự làm (Tham khảo các ví dụ mẫu)

**Bài 1.** Viết chương trình thực hiện các hiệu ứng LED như sau (chạy lần lượt các hiệu ứng, mỗi hiệu ứng 5 lần, tạo trễ thích hợp):

(o: led off, O: led on)

1) Flash: Bật tắt cả 8 led, trễ thích hợp sau đó, tắt tắt cả 8 led.

o o o o o o o o → O O O O O O O O → o o o o o o o o

2): Running spot R: Bật lần lượt từng led từ trái qua phải

o o o o o o o o → O o o o o o o o → o O o o o o o o → o o O o o o o o →  
o o o O o o o o → ..... → o o o o o o o O → o o o o o o o o

3): Running spot L: Bật lần lượt từng led từ phải qua trái

o o o o o o o o → o o o o o o o O → o o o o o o O o → o o o o o O o o →  
o o o o O o o o → ..... → O o o o o o o o → o o o o o o o o

4): Spot bumper: Bật lần lượt 2 led đối xứng từ ngoài vào trong rồi từ trong ra ngoài

o o o o o o o o → O o o o o o O → o O o o o o O o → o o O o o O o o →  
o o o O O o o o → o o o O O o o o → o o O o o O o o → o O o o o O o o →  
O o o o o o O → o o o o o o o

**Bài 2.** Kết hợp sử dụng ngắt ngoài Int0, Int1 để thay đổi lần lượt 4 hiệu ứng led trong bài

1. Bấm Int1 (key5) sẽ chuyển sang hiệu ứng tiếp theo, bấm Int0 (key6) để quay lại hiệu ứng trước). Hiện số thứ tự hiệu ứng tương ứng (1,2,3,4) trên led 7 thanh

Chú ý: Nút bấm ngắt ngoài Int0, Int1 không có mạch chống nảy phím => Có thể có nhiều lần ngắt trong 1 lần bấm phím. Xử lý bằng cách cấm ngắt EA=0 khi có 1 ngắt xảy ra, thực hiện thay đổi trạng thái, hiệu ứng LED (1 lượt) rồi cho phép ngắt trở lại EA=1. Hoặc thêm delay sau khi có ngắt xảy ra.

**Bài 3.** Viết chương trình có hàm tạo trễ thời gian thực 1s sử dụng ngắt timer. Hiển thị đếm giây trên 2 led 7 thanh (00->99).

**Bài 4.** Viết chương trình tạo đồng hồ stopwatch đếm thời gian 00.00 hiển thị trên 4 led 7 thanh tương ứng. Sử dụng các nút bấm key1/key2 để lựa chọn start/stop