

## Bài thực hành 1

### GHÉP NỐI VI ĐIỀU KHIỂN VỚI NGOẠI VI

#### 1. Mục đích

- củng cố kiến thức về hai ngoại vi tiêu biểu của vi điều khiển: ngắt ngoài và timer.
- Thực hành ghép nối với LCD ký tự, lập trình điều khiển hiển thị nội dung lên LCD với các hiệu ứng theo bộ lệnh của LCD.
- Tìm hiểu NEC Protocol ứng dụng trong điều khiển từ xa hồng ngoại. Lập trình kết hợp ngắt ngoài và timer để nhận lệnh từ điều khiển từ xa.
- Xây dựng ứng dụng đơn giản với LCD, remote control, buzzer.

#### 2. Chuẩn bị

- Bộ KIT 8051 PRO với vi điều khiển STC89C52.
- Phụ kiện: điều khiển từ xa hồng ngoại.
- Phần mềm: Code::Blocks+SDCC, stc-isp-15xx-v6.86E.

#### 3. Bài tập mẫu

##### 3.1. Lập trình ghép nối ngắt ngoài

- Tạo project trên Code::Blocks với file mã nguồn mẫu ext-interrupt.c.
- Biên dịch và nạp firmware lên mạch.
- Quan sát hoạt động của mạch và giải thích mã nguồn.

##### 3.2. Lập trình ghép nối timer

- Tạo project trên Code::Blocks với file mã nguồn mẫu timer.c.
- Biên dịch và nạp firmware lên mạch.
- Quan sát hoạt động của mạch và giải thích mã nguồn.

##### 3.3. Lập trình ghép nối LCD1602

- Tạo project trên Code::Blocks với các file lcd1602.c, lcd1602.h, và lcd1602demo.c.
- Hoàn thiện code trong file lcd1602.c để mạch thực hiện chức năng:

+ Hiển thị dòng chữ                      Hello Elitech 65

\*\*\*\*\*

- + Màn hình chạy sang trái cho đến hết, rồi chạy lại sang phải, dừng chờ 3 giây.
- Giải thích mã nguồn.

### 3.4. Lập trình phát hiện sự kiện

- Tạo project trên Code::Blocks với các file double\_click.c.
- Hoàn thiện hàm main để mạch thực hiện chức năng.
  - + Nếu nút K4 được nhấn đúp thì đèn D8 sáng. Điều kiện nhấn đúp: thời điểm 2 lần bấm nút K4 cách nhau không quá 200 ms.
  - + Nếu quá 500 ms mà nút K4 không được bấm lại, thì đèn D8 tắt.
- Giải thích mã nguồn.

## 4. Bài tập tự làm

### 4.1. Giải mã lệnh điều khiển từ xa hồng ngoại

- Tìm hiểu phương pháp điều chế/giải điều chế tín hiệu khi truyền qua sóng hồng ngoại: [Infrared Communication Concepts | Online Documentation for Altium Products](#)
- Tham khảo tài liệu NEC Protocol mô tả khuôn dạng và cách mã hóa tín hiệu điều khiển từ xa hồng ngoại ở Phụ lục.
- Tìm hiểu datasheet IC thu và giải điều chế tín hiệu điều khiển từ xa HS0038.
- Đề xuất thuật toán giải mã tín hiệu nhận được từ HS0038 để khôi phục dữ liệu gửi từ điều khiển từ xa.
- Xây dựng chương trình thực hiện chức năng:
  - + Giải mã lệnh gửi từ điều khiển từ xa tới mạch.
  - + Hiển thị giá trị mã lệnh lên LCD.
  - + Nếu mã lệnh tương ứng với nút ON/OFF thì bật/tắt đèn D1 tương ứng.
  - + Mở rộng: khi đèn D1 sáng (trạng thái ON) thì bấm tiếp nút Mute trên điều khiển để bật/tắt buzzer.

-----

**Phu lục:** khuôn dạng và cách mã hóa lệnh của điều khiển từ xa theo NEC Protocol

# NEC Protocol

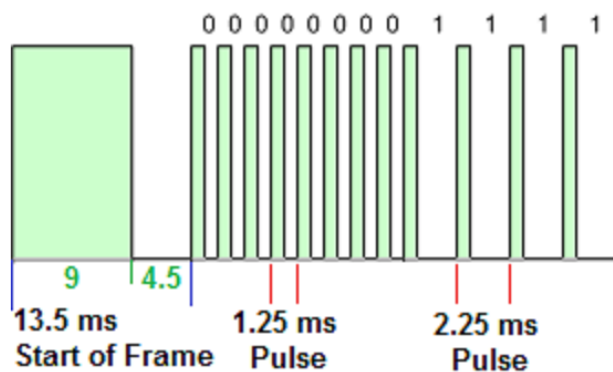
NEC IR protocol encodes the keys using a 32bit frame format as shown below.

NEC Frame Format			
Address	Complement of Address	Command	Complement of Command
LSB-MSB(0-7)	LSB-MSB(8-15)	LSB-MSB(16-23)	LSB-MSB(24-31)

Each bit is transmitted using the pulse distance as shown in the image.

**Logical '0':** A 562.5µs pulse burst followed by a 562.5µs space, with a total transmit time of 1.125ms

**Logical '1':** A 562.5µs pulse burst followed by a 1.6875ms space, with a total transmit time of 2.25ms



When a key is pressed on the remote controller, the message transmitted consists of the following, in order:

1. A 9ms leading pulse burst (16 times the pulse burst length used for a logical data bit)
2. A 4.5ms space
3. The 8-bit address for the receiving device
4. The 8-bit logical inverse of the address
5. The 8-bit command
6. The 8-bit logical inverse of the command
7. A final 562.5µs pulse burst to signify the end of message transmission.

The four bytes of data bits are each sent least significant bit first. Below image illustrates the format of an NEC IR transmission frame, for an address of 00h (00000000b) and a command of ADh (10101101b).

A total of 67.5ms is required to transmit a message frame. It needs 27ms to transmit the 16 bits of address (address + inverse) and the 16 bits of command (command + inverse).

