

Bài thực hành số 1

LẬP TRÌNH VI ĐIỀU KHIỂN – Phần 2

1. Mục đích

- Lập trình cơ bản cho vi điều khiển họ 8051, điều khiển các ngoại vi cơ bản như cổng vào ra, timer, ngắt.
- Thực hiện được các ghép nối với LED đơn, LED 7 thanh, nút bấm, LCD1602,...

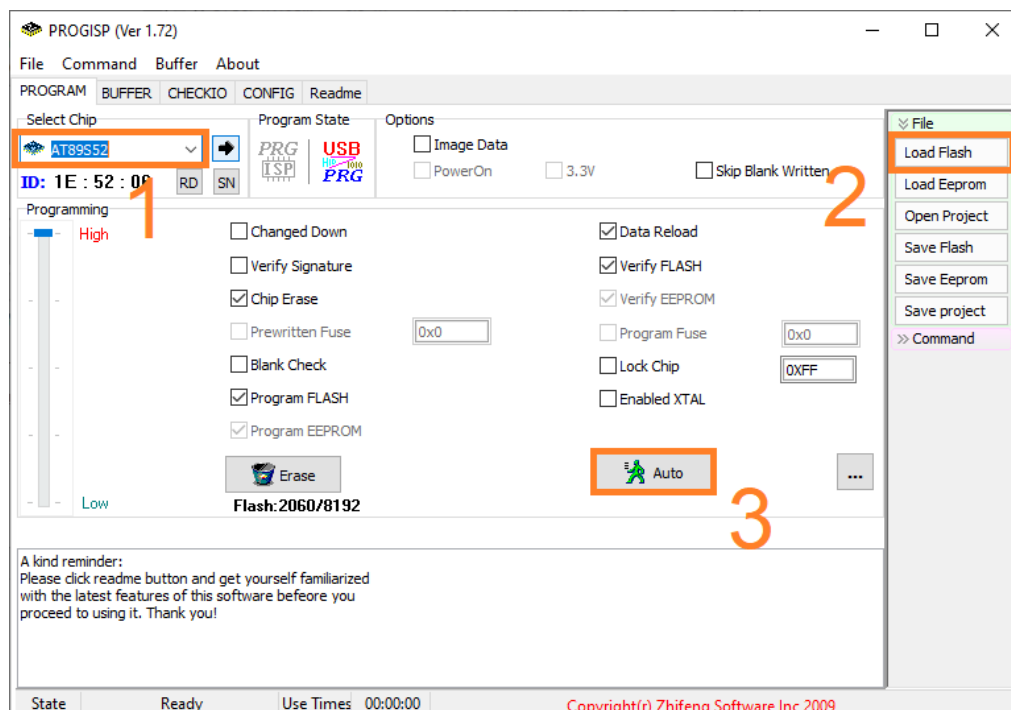
2. Chuẩn bị

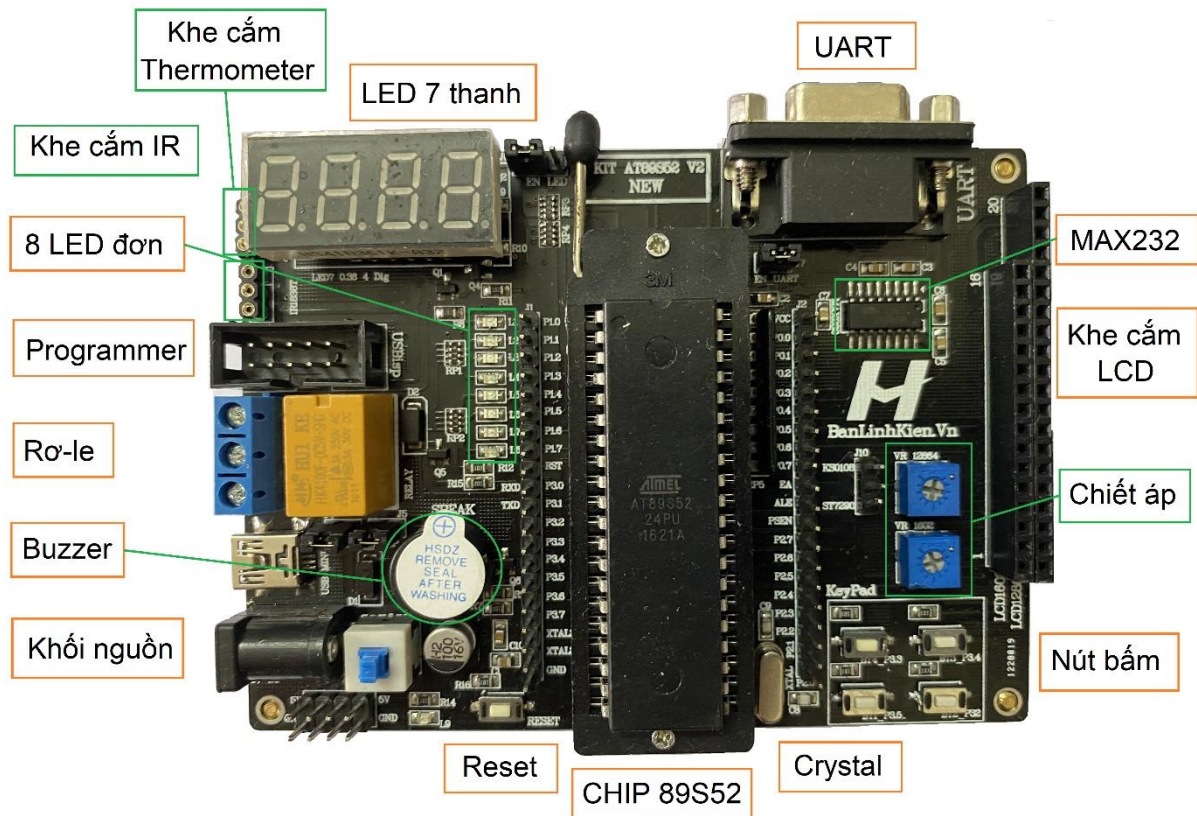
- KIT AT89S52 V2 NEW, sơ đồ mạch, các phụ kiện.
- Phần mềm: CodeBlocks và [SDCC](#), phần mềm nạp ProgISP phiên bản 1.72 trở lên.
- Mã nguồn mẫu và tài liệu hướng dẫn.

3. Nội dung

3.1. Làm quen với bo mạch KIT AT89S52 V2 NEW

- Quan sát bo mạch KIT AT89S52 V2 NEW, tìm hiểu các thành phần như mô tả trên hình vẽ.
- Tìm hiểu file schematics đi kèm.
- Nạp file firmware **testKIT.hex**:
 - Kết nối KIT AT89S52 V2 NEW với máy tính qua dây mạch nạp được cung cấp.
 - Chạy phần mềm *progisp.exe*.
 - Chọn đúng loại MCU (AT89S52). Bấm Load Flash rồi mở file hex đã tải. Bấm Auto để nạp chương trình vào MCU.
- Kiểm tra hoạt động của mạch: tiếng Bíp (rời ngưng), các LED đơn lần lượt sáng, dãy chữ số 1234 hiển thị trên 4 LED 7 thanh, nhấn mỗi nút bấm sẽ có tiếng Bíp.

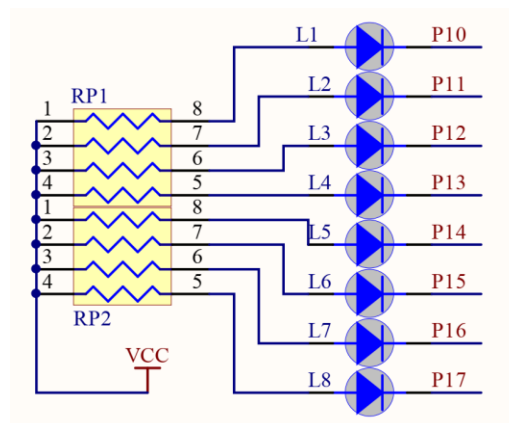




3.2. Lập trình trên KIT AT89S52 V2 NEW

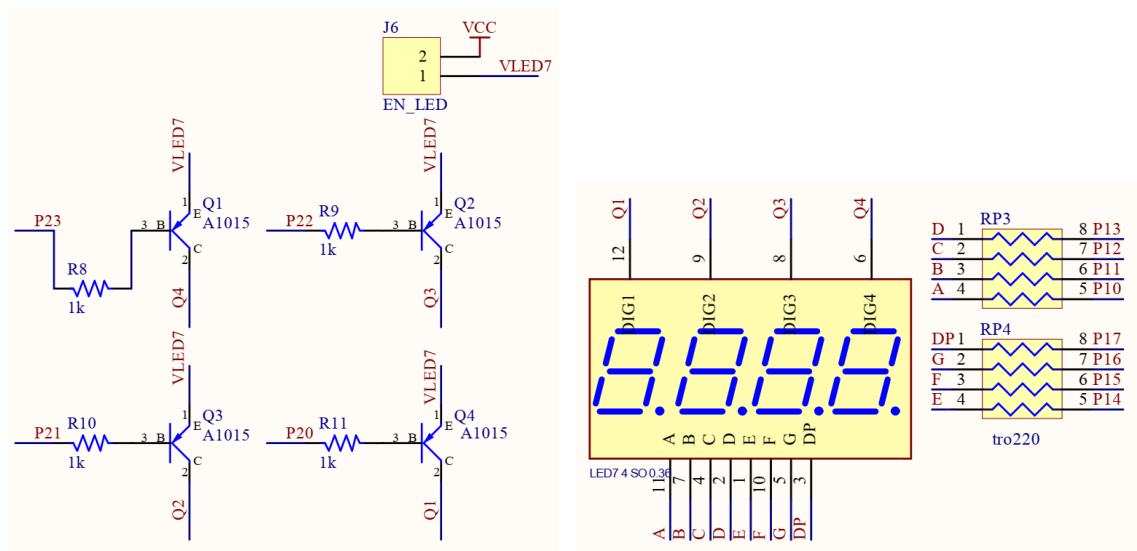
3.2.1 LED đơn

- Tham khảo sơ đồ mạch: 8 LED đơn được ghép nối vào cổng P1.
- Đọc hiểu mã nguồn file “**led_blinking.c**”. Tạo project trên SDCC với file mã nguồn và nạp code lên chip.
- Kiểm tra hoạt động của mạch.



3.2.2 LED 7 thanh

- Tham khảo sơ đồ mạch ghép nối LED 7 thanh.
 - Mạch có 4 module LED 7 thanh loại common cathode.
 - Các tín hiệu **a...dp** được nối trực tiếp với cổng P1.
 - Các tín hiệu chọn LED 7 thanh (LED1...LED4) được lấy từ 4 chân cổng P2.0, P2.1, P2.2, P2.3.

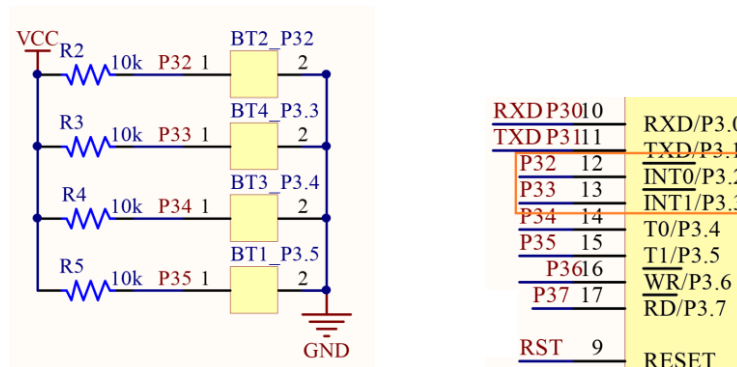


- Đọc hiểu mã nguồn file “**7seg.c**”. Tạo project trên SDCC với file mã nguồn và nạp code lên chip.

- Kiểm tra và giải thích hoạt động của mạch.

3.2.3 Ngắt ngoài

- Tham khảo sơ đồ mạch ghép nối 4 nút bấm, trong đó có 2 nút BT2 và BT3 được nối với ngắt ngoài INT0 và INT1.



- **Chú ý:** Tất cả các chân của chip đều đã có điện trở treo 10K lên VCC.

- Đọc hiểu mã nguồn file “**interrupt.c**”. Tạo project trên SDCC với file mã nguồn và nạp code lên chip. Kiểm tra và giải thích hoạt động của mạch.

3.2.4 Timer

- Đọc hiểu mã nguồn file “**timer_led.c**”. Tạo project trên SDCC với file mã nguồn và nạp code lên chip. Kiểm tra và giải thích hoạt động của mạch.

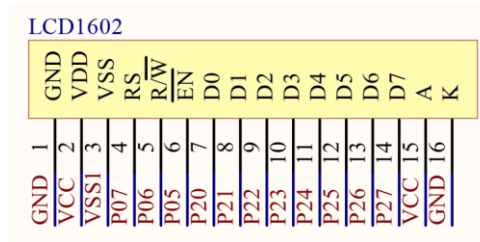
3.2.5 LCD 1602

- Trên KIT AT89S52 V2 NEW đã có sẵn khe cắm để kết nối với module màn hình LCD1602.

- Cắm LCD1602 vào slot 16 chân ở cạnh phải của kit. Chú ý số thứ tự chân trên LCD và trên mạch phải trùng nhau.

- Tham khảo sơ đồ mạch của KIT AT89S52 V2 NEW phần ghép nối LCD.

- Đọc hiểu mã nguồn file “**lcd.c**”. Tạo project trên SDCC với file mã nguồn và nạp code lên chip. Kiểm tra và giải thích hoạt động của mạch.



- Sử dụng code mẫu trong file “**lcd.c**”, lập trình hiển thị nội dung dưới đây trên màn hình LCD.

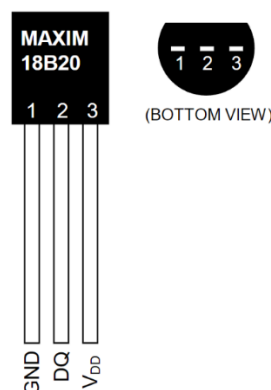
“He nhung 2022-2

Xin chao cac ban”

- Lập trình để tạo hiệu ứng chữ chạy sang trái/phải.

3.2.6 Cảm biến nhiệt độ DS18B20

- DS18B20 là cảm biến nhiệt độ số có 3 chân (GND, DQ, V_{DD}), trong đó DQ là chân dữ liệu. Bên trong DS18B20 được tích hợp cả cảm biến nhiệt và bộ chuyển đổi ADC. Đọc datasheet của cảm biến nhiệt độ số [DS18B20](#) và tìm hiểu cách sử dụng.



- Đọc hiểu mã nguồn file “**ds18b20.c**”. Tạo project trên SDCC với file mã nguồn và nạp code lên chip. Kiểm tra và giải thích hoạt động của mạch.

3.3. Bài tập tự làm

Bài 1. Viết chương trình C thực hiện các hiệu ứng LED như sau (chạy lần lượt các hiệu ứng, mỗi hiệu ứng 2 lần, tạo trễ thích hợp):

(◦: LED off, O: LED on)

- 1) Bật lần lượt từng cặp LED đơn từ trái qua phải và từ phải qua trái.

```

◦ ◦ ◦ ◦ ◦ ◦ → O O ◦ ◦ ◦ ◦ → ◦ O O ◦ ◦ ◦ ◦ → ◦ ◦ O O ◦ ◦
→ ◦ ◦ ◦ O O ◦ ◦ → ... → ◦ ◦ ◦ ◦ ◦ ◦ O O → ◦ ◦ ◦ ◦ ◦ ◦
→ ◦ ◦ ◦ ◦ ◦ O O → ◦ ◦ ◦ ◦ O O ◦ ◦ → ... → ◦ ◦ ◦ ◦ ◦ ◦

```

- 2) Bật lần lượt từng LED từ trái và dồn qua phải rồi tắt lần lượt 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 ◦ ◦ ◦ ◦ ◦ ◦ → ◦ ◦ ◦ ◦ ◦ ◦

```

3) Bật lần lượt 2 LED đối xứng từ ngoài vào trong rồi từ trong ra ngoài

○ ○ ○ ○ ○ ○ ○ ○ → ○ ○ ○ ○ ○ ○ ○ ○ → ○ ○ ○ ○ ○ ○ ○ ○ → ○ ○ ○ ○ ○ ○ ○ ○ →
 ○ ○ ○ ○ ○ ○ ○ ○ → ○ ○ ○ ○ ○ ○ ○ ○ → ○ ○ ○ ○ ○ ○ ○ ○ → ○ ○ ○ ○ ○ ○ ○ ○ →
 ○ ○ ○ ○ ○ ○ ○ ○ → ○ ○ ○ ○ ○ ○ ○ ○

Bài 2. Viết chương trình C sử dụng ngắt ngoài INT0, INT1 để thay đổi lần lượt 4 hiệu ứng LED trong bài 1. Bấm K4 sẽ chuyển sang hiệu ứng tiếp theo, bấm K3 để quay lại hiệu ứng trước.

Bài 3. Viết chương trình C để mô phỏng đồng hồ đếm ngược. Sử dụng 2 LED 7 thanh bên trái để hiển thị số phút và 2 LED 7 thanh bên phải để hiển thị số giây. Ban đầu đồng hồ hiển thị giá trị 00.00. Sử dụng 1 nút bấm để cài đặt thời gian đếm ngược theo phút (ví dụ: 05.00), và 1 nút bấm khác để bắt đầu đếm ngược. Khi đếm ngược về 00.00 thì còi (*buzzer*) sẽ phát 3 tiếng, mỗi tiếng 1s. **Lưu ý:** sử dụng timer để tạo trễ chính xác.

05.36

Minh họa đồng hồ đếm ngược

Bài 4. Viết chương trình C để viết ứng dụng theo dõi nhiệt độ theo thời gian thực. Cụ thể, nhiệt độ môi trường được nhận biết thông qua cảm biến nhiệt (DS18B20) và được cập nhật sau mỗi 6s. Thông tin nhiệt độ được hiển thị trên một LCD 1602. Cụ thể, dòng trên để hiển thị nhiệt độ tại thời điểm hiện tại và dòng dưới hiển thị nhiệt độ cao nhất và thấp nhất trong khoảng thời gian 1 phút theo dõi trước đó. Sử dụng 1 nút bấm (tùy chọn) để chuyển đổi giá trị nhiệt độ giữa °C và °F trên LCD.

Cur Temp: 30°C
L: 25 H: 31

Minh họa hiển thị nhiệt độ trên LCD1602

--- Hết ---