

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

BỘ MÔN TIN HỌC CÔNG NGHIỆP



TÀI LIỆU THỰC HÀNH

HỌC PHẦN: VI XỬ LÝ – VI ĐIỀU KHIỂN

SỐ TÍN CHỈ: 01

Thái Nguyên, năm 2019

MỤC LỤC

PHẦN 1. VI XỬ LÝ 8086.....	4
BÀI 1. THỰC HÀNH LẬP TRÌNH VI XỬ LÝ 8086.....	4
1.1. Mục đích của phần thực hành.....	4
1.2. Kiến thức cần chuẩn bị:.....	4
1.3. Bài thực hành mẫu.....	4
1.4. Nội dung thực hành.....	6
PHẦN 2. VI ĐIỀU KHIỂN 8051.....	9
BÀI 2. VÀO/RA SỐ.....	9
2.1. Mục đích của phần thực hành.....	9
2.2. Các kiến thức cần chuẩn bị.....	9
2.3. Bài tập thực hành mẫu.....	9
2.3.1. Thiết kế và lập trình điều khiển led đơn và nút bấm.....	9
2.3.2. Thiết kế và lập trình điều khiển led 7 thanh và nút bấm.....	16
2.4. Nội dung thực hành.....	19
BÀI 3: GIAO TIẾP BÀN PHÍM VÀ ĐIỀU KHIỂN ĐỘNG CƠ.....	20
3.1. Mục đích của phần thực hành.....	20
3.2. Các kiến thức cần chuẩn bị.....	20
3.3. Bài tập thực hành mẫu.....	20
3.3.1. Thiết kế và lập trình điều khiển Động cơ và nút bấm.....	20
3.3.2. Thiết kế và lập trình điều khiển Keypad và Led 7 thanh.....	22
3.4. Nội dung thực hành.....	26
BÀI 4: HIỂN THỊ LED MATRIX VÀ LCD.....	27
4.1. Mục đích của phần thực hành:.....	27
4.2. Các kiến thức cần chuẩn bị:.....	27
4.3. Bài thực hành mẫu.....	27
4.3.1. Thiết kế và lập trình điều khiển Led Matrix.....	27
4.3.2. Thiết kế và lập trình điều khiển LCD.....	29
4.4. Nội dung thực hành:.....	32
PHỤ LỤC.....	33
PHỤ LỤC 1: SƠ ĐỒ MẠCH NGUYÊN LÝ TỔNG THỂ.....	33
PHỤ LỤC 2: GIỚI THIỆU MODULE ĐÀO TẠO VI ĐIỀU KHIỂN 8051.....	34
1.1. Giới thiệu.....	34
1.2. Nhung tính năng của Kit vi điều khiển 8051.....	35
1.3. Cấu hình chi tiết các khối module.....	36
1.3.1. Khối nguồn:.....	36
1.3.2. Khối Vi điều khiển.....	36
1.3.3. Khối hiển thị LCD 16x2:.....	37
1.3.4. Khối phím đơn.....	37
1.3.5. Khối phím ma trận.....	37
1.3.6. Khối giao tiếp RS232.....	38
1.3.7. Khối tạo xung (Bộ đếm).....	38
1.3.8. Khối thời gian thực (Real Time Clock).....	39
1.3.9. Khối đo nhiệt độ - DS18S20.....	39
1.3.10. Khối thu tín hiệu hồng ngoại – IR.....	40
1.3.11. Khối chuyển đổi analog – digital (ADC).....	40

1.3.12.	Khối hiển thị LED 7 thanh.....	40
1.3.13.	Khối led Matrix.....	41
1.3.14.	Khối led đơn.....	41
1.3.15.	Khối điều khiển STEP Motor	42
1.3.16.	Khối điều khiển động cơ một chiều (DC motor)	42
1.3.17.	Khối hiển thị GLCD 128x64.....	42
PHỤ LỤC 3: HƯỚNG DẪN SỬ DỤNG MODULE ĐÀO TẠO VI ĐIỀU KHIỂN 8051		
.....		43
2.1.	Hướng dẫn sử dụng Module đào tạo vi điều khiển 8051	43
2.1.1.	Khi kết nối mạch.....	43
2.1.2.	Khi viết chương trình:.....	43
2.2.	Nạp chương trình cho Vi điều khiển dùng ISP Prog	43

PHẦN 1. VI XỬ LÝ 8086

BÀI 1. THỰC HÀNH LẬP TRÌNH VI XỬ LÝ 8086

1.1. Mục đích của phần thực hành

1. Giúp sinh viên củng cố các kiến thức về thiết kế và lập trình cho một hệ Vi xử lý
2. Giúp sinh viên chuyển đổi tư duy từ kiến thức lý thuyết đến tiếp cận với kiến thức thực tế
3. Làm quen vi xử lý 8086.
4. Làm quen với phần mềm lập trình EMU8086 và mô phỏng lập trình vi xử lý 8086.
5. Củng cố các kiến thức về lập trình Assembly cho vi xử lý 8086.

1.2. Kiến thức cần chuẩn bị:

1. Các thao tác cơ bản trên hệ điều hành Windows.
2. Cấu trúc chương trình hợp ngữ dạng COM.
3. Trình tự Soạn thảo – Dịch chương trình dùng EMU8086.
4. Tập lệnh của vi xử lý 8086.

1.3. Bài thực hành mẫu

Nhập và sửa lỗi đoạn chương trình sau đây để nhận 1 ký tự. Nếu là ký tự HOA thì in ra màn hình "Ky tu HOA". Ngược lại in ra câu "Ky tu thuong".

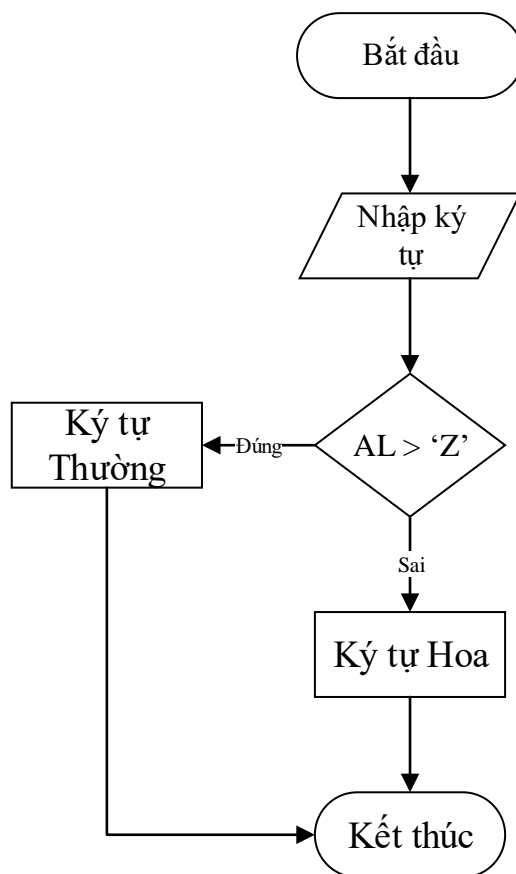
➤ **Các kiến thức cần có**

- Cách viết một chương trình dạng COM
- Khai báo biến
- Nhập một ký tự từ bàn phím
- Cách sử dụng bảng mã ASCII
- Cách xuất một thông báo

➤ **Phân tích cách giải**

Nhập một ký tự từ bàn phím, so sánh ký tự vừa nhập (được lưu trong AL) với ký tự 'Z'. Nếu ký tự vừa nhập lớn hơn ký tự 'Z' thì đó là ký tự hoa, ngược lại thì đó là ký tự thường.

➤ **Sơ đồ thuật toán**



#####

➤ **Code mẫu**

```

ORG 100H
tbao1 DB "Ky tu HOA.$"
tbao2 DB "Ky tu thuong.$"
mov ax, data
mov ds, ax
mov ah, 01h
int 21h
cmp al, 'Z'
ja nhan
mov ah, 09
lea dx, tbao1
int 21h
jmp exit
nhan:
mov ah, 09
lea dx, tbao2
int 21h
exit:
mov ah, 7
int 21h
mov ah, 4Ch
RET
  
```

1.4. Nội dung thực hành

- Thực hiện bài làm trong vòng 90 phút và lưu kết quả vào thư mục có tên đặt theo quy tắc sau: **STT_Đề số** (Ví dụ: 1_Đề12).
- Điểm đánh giá mỗi bài được tính như một bài kiểm tra thành phần.
- Sinh viên được bốc chọn ngẫu nhiên 1 đề trong các đề sau để làm bài thực hành.

- Đề số 1. Viết chương trình cho phép nhập vào 2 số A, B từ bàn phím. Tính hiệu 2 số A-B (với $A \leq B < 10$), in ra màn hình kết quả.
- Đề số 2. Viết chương trình cho phép nhập vào 2 số A, B từ bàn phím. Tính tổng 2 số A+B (với $A, B < 10$), in ra màn hình kết quả.
- Đề số 3. Viết chương trình nhập vào 2 chữ in thường bất kỳ giả sử ký hiệu 2 chữ đó là ch1 và ch2 (với $ch1 < ch2$) và in các ký tự nằm trong khoảng từ ch1 đến ch2.
Ví dụ: nhập ký tự thứ nhất là a, ký tự thứ hai là h kết quả sẽ in các ký tự nằm trong khoảng từ a đến h: abcdefgh.
- Đề số 4. Viết chương trình để nhập một mảng và in ra màn hình phần tử nhỏ hơn 5 của mảng (chú ý: các phần tử mảng có giá trị nhỏ hơn 10)
- Đề số 5. Viết chương trình để nhập một mảng và in ra màn hình phần tử chia hết cho 3 trong mảng (chú ý: các phần tử mảng có giá trị nhỏ hơn 10)
- Đề số 6. Viết chương trình để nhập một mảng và in ra màn hình phần tử chẵn trong mảng (chú ý: các phần tử mảng có giá trị nhỏ hơn 10)
- Đề số 7. Viết chương trình để nhập một mảng và in ra màn hình phần tử lẻ trong mảng (chú ý: các phần tử mảng có giá trị nhỏ hơn 10)
- Đề số 8. Viết chương trình để nhập một mảng và in ra màn hình phần tử lớn nhất trong mảng (chú ý: các phần tử mảng có giá trị nhỏ hơn 10)
- Đề số 9. Viết chương trình để nhập một mảng và in ra màn hình phần tử bé nhất trong mảng (chú ý: các phần tử mảng có giá trị nhỏ hơn 10)
- Đề số 10. Thực hiện nhập một ký tự bất kỳ từ bàn phím và in ra màn hình 10 dòng, mỗi dòng chứa 10 ký tự vừa nhập
- Đề số 11. Viết chương trình để nhập một số nguyên nhỏ hơn 10 từ bàn phím. Kiểm tra số nguyên đó là số chẵn hay lẻ. Nếu lẻ in ra màn hình là: LE; Nếu chẵn in ra màn hình là: CHAN
- Đề số 12. Nhập vào một xâu có độ dài nhỏ hơn 10 ký tự, in ra màn hình độ dài của xâu đó.
- Đề số 13. Thực hiện nhập vào từ bàn phím 2 ký tự trong đó phải có 1 ký tự là số, 1 ký tự là chữ in thường.

- Đề số 14. Thực hiện nhập vào 2 số tự nhiên nhỏ hơn 10. Tìm số nhỏ hơn trong 2 số và in kết quả ra màn hình trên dòng tiếp theo.
- Đề số 15. Thực hiện nhập vào 2 số tự nhiên nhỏ hơn 10. Tìm số lớn hơn trong 2 số và in kết quả ra màn hình trên dòng tiếp theo.
- Đề số 16. Chương trình nhập vào hai số nguyên A, B nhỏ hơn 10. Tính $X=B-2*A$ và in kết quả ra màn hình
- Đề số 17. Viết chương trình để nhập vào một xâu bất kỳ, in ra màn hình các ký tự không phải là chữ.
- Đề số 18. Viết chương trình để nhập vào một xâu bất kỳ, in ra màn hình các ký tự là số.
- Đề số 19. Viết chương trình nhập vào một ký tự bất kỳ và in ra màn hình ký tự đứng trước và sau ký tự đó.
- Đề số 20. Viết chương trình nếu nhập vào ký tự hoa thì in ra màn hình ký tự thường; nếu nhập vào ký tự thường thì in ra màn hình ký tự hoa.
- Đề số 21. Viết chương trình nhập vào một xâu bất kỳ, in xâu đó ra màn hình dưới dạng chữ thường.
- Đề số 22. Viết chương trình nhập vào một xâu bất kỳ, in xâu đó ra màn hình dưới dạng chữ hoa.
- Đề số 23. Viết chương nhập Họ tên của anh(chị) bằng chữ thường và in ra màn hình dưới dạng chữ hoa.
- Đề số 24. Cho 2 xâu ký tự S1 và S2, kiểm tra xem độ dài S1 và S2 có bằng nhau hay không? (Chú ý: Nhập 2 xâu bất kỳ có độ dài nhỏ hơn 10 ký tự).
- Đề số 25. Nhập một xâu bất kỳ từ bàn phím, viết chương trình để in xâu đảo ngược của xâu vừa nhập ra màn hình.
- Đề số 26. Thực hiện nhập vào từ bàn phím một chữ in thường. Yêu cầu in ra các chữ in thường từ 'a' đến ký tự vừa nhập.
- Đề số 27. Thực hiện nhập vào từ bàn phím một chữ cái in thường. Yêu cầu in ra các chữ in thường từ ký tự vừa nhập đến 'z'.
- Đề số 28. Nhập vào 2 chữ in thường bất kỳ là thực hiện đếm số chữ cái trong đoạn từ chữ cái thứ nhất đến chữ cái thứ hai và in ra màn hình. (Chỉ nhập các chữ có khoảng <10).
- Đề số 29. Cho trước 3 số tự nhiên trong thanh ghi AX, BX và CX. Viết chương trình để in ra màn hình thanh ghi có giá trị nhỏ nhất.
- Đề số 30. Cho trước 3 số tự nhiên trong thanh ghi thanh ghi AX, BX và CX. Viết chương trình để in ra màn hình thanh ghi có giá trị lớn nhất.

PHẦN 2. VI ĐIỀU KHIỂN 8051

BÀI 2. VÀO/RA SỐ

2.1. Mục đích của phần thực hành

Qua bài thí nghiệm này giúp sinh viên làm chủ được:

- Phương pháp vẽ mạch mô phỏng dùng phần mềm Proteus
- Phương pháp lập trình Assembly cho vi điều khiển 8051
- Một số lệnh cơ bản của vi điều khiển 8051
- Các lệnh nhập cơ bản qua cổng I/O nối với nút bấm

2.2. Các kiến thức cần chuẩn bị

- Nắm vững tập lệnh của vi điều khiển MCS-51
- Nắm vững các kỹ thuật lập trình I/O của MCS-51
- Nắm vững các cách thiết kế và lập trình điều khiển nút bấm kết nối để hiển thị Led đơn, Led 7 đoạn.

2.3. Bài tập thực hành mẫu

2.3.1. Thiết kế và lập trình điều khiển led đơn và nút bấm

- Viết chương trình cho phép các Led đơn sáng với 8 chu trình khác nhau. Các chu trình được chọn bằng cách bấm 1 trong 8 nút bấm đơn.
- Trình tự thực hành
 - **Đọc sơ đồ nguyên lý để biết các tín hiệu điều khiển**

➤ **Các kiến thức cần có**

Trong các kiến thức cần chuẩn bị, cần chú ý nắm rõ các kiến thức:

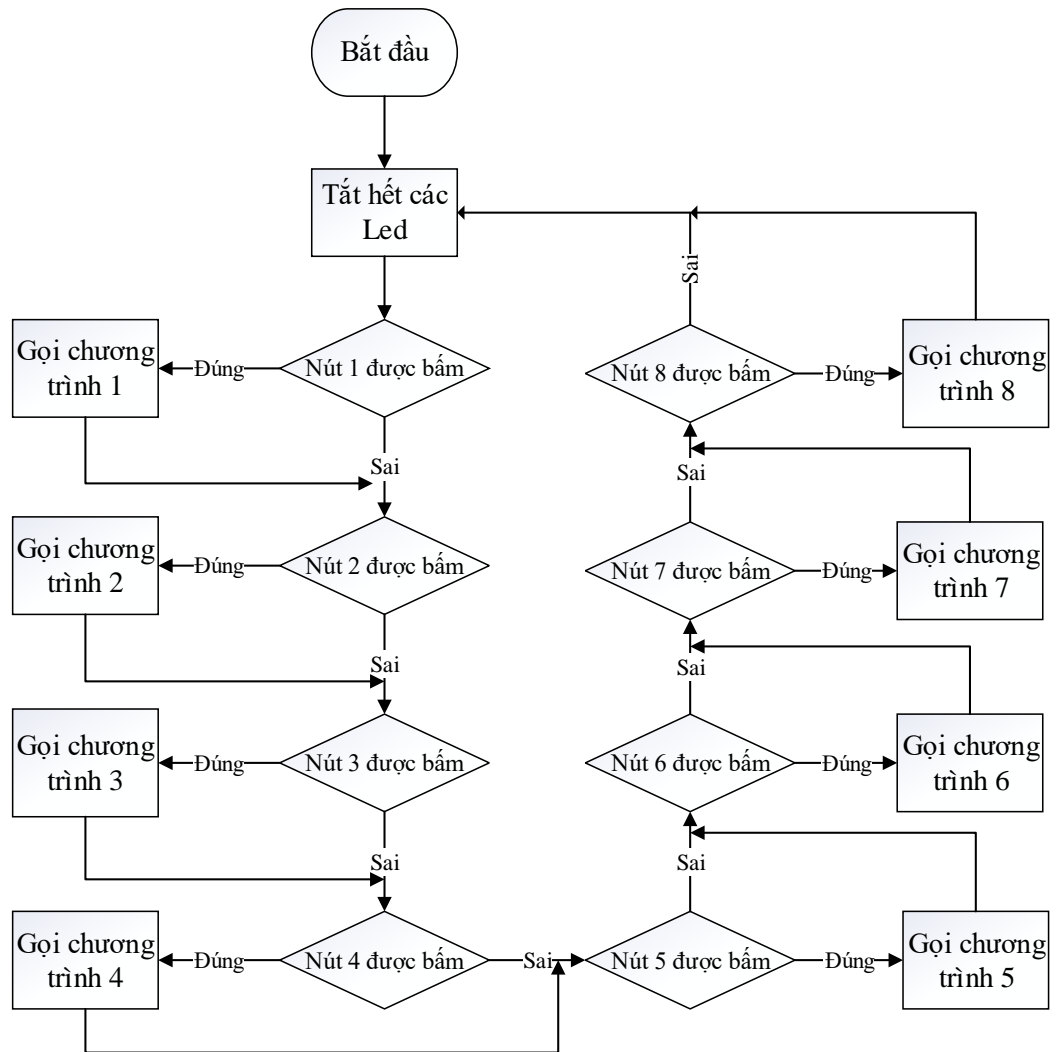
- Cấu trúc một chương trình viết bằng ngôn ngữ Assembly.
- Nguyên lý hoạt động của Led đơn.
- Cách viết chương trình Delay (Dùng timer hoặc không dùng Timer).
- Cách kiểm tra trạng thái của nút bấm.
- Cách viết, gọi một chương trình con.
- Các lệnh JB, JNB, DJNZ, CJNE và các lệnh quay (RR, RL, RLC và RRC).

➤ **Phân tích cách giải**

- Chương trình cần điều khiển 8 Led đơn sáng với 8 chu trình khác nhau, do đó, ta sẽ viết 8 chương trình con. Mỗi chương trình con sẽ điều khiển 8 Led đơn sáng một kiểu nào đó.
VD: Chương trình con số một sẽ điều khiển 8 Led sáng nhấp nháy. Chương trình con số 2 sẽ điều khiển 8 Led sáng đan xen...
- Kiểm tra trạng thái của 8 nút bấm để hiển thị chu trình tương ứng với nút được bấm.
VD: Nút số 1 được bấm sẽ nhảy đến chương trình con số 1.

Nút số 2 được bấm sẽ nhảy đến chương trình con số 2...

➤ **Sơ đồ thuật toán**



➤ **Lập trình:**

```
; Khai báo hằng số LED_PORT cho cổng P2
LED_PORT EQU P2

; Địa chỉ bắt đầu thực hiện chương trình
ORG 00H

; Tắt tất cả các đèn
MOV LED_PORT, #0FFH

; Nhãn của chương trình chính
MAIN:

; Nếu nút bấm được nối ở chân P1.0 không được bấm thì nhảy
```

; đến nhãn EXIT_1 để kiểm tra nút tiếp theo. Còn nếu được
; bấm thì gọi chương trình con tương ứng CHUONG_TRINH_1.

```
JB P1.0, EXIT_1  
    CALL CHUONG_TRINH_1
```

```
EXIT_1:
```

```
JB P1.1, EXIT_2  
    CALL CHUONG_TRINH_2
```

```
EXIT_2:
```

```
JB P1.2, EXIT_3  
    CALL CHUONG_TRINH_3
```

```
EXIT_3:
```

```
JB P1.3, EXIT_4  
    CALL CHUONG_TRINH_4
```

```
EXIT_4:
```

```
JB P1.4, EXIT_5  
    CALL CHUONG_TRINH_5
```

```
EXIT_5:
```

```
JB P1.5, EXIT_6  
    CALL CHUONG_TRINH_6
```

```
EXIT_6:
```

```
JB P1.6, EXIT_7  
    CALL CHUONG_TRINH_7
```

```
EXIT_7:
```

```
JB P1.7, EXIT_8  
    CALL CHUONG_TRINH_8
```

```
EXIT_8:
```

```
MOV P2, #0FFH
```

```
JMP MAIN
```

```

; Định nghĩa các chu trình sáng khác nhau.
; Chu trình 1: Có 7 led sáng, 1 led tắt chạy từ trái qua phải
CHUONG_TRINH_1:
    MOV A, #00000001B      ; Khởi tạo giá trị cho thanh ghi A
    MOV R1, #8             ; Tạo vòng lặp 8 lần.
    LOOP:
        MOV LED_PORT, A   ; Xuất giá trị trong thanh
                            ; ghi A ra cổng
        RL A               ; Quay các bit của A sang trái 1 bit
        CALL DELAY        ; gọi chương trình tạo trễ
    DJNZ R1, LOOP
    RET

; Chu trình 2: Có 7 led sáng, 1 led tắt chạy từ phải qua trái
CHUONG_TRINH_2:
    MOV A, #10000000B
    MOV R2, #8
    LOOP1:
        MOV LED_PORT, A
        RR A
        CALL DELAY
    DJNZ R2, LOOP1
    RET

; Chu trình 3: Các led sáng đan xen
CHUONG_TRINH_3:
    MOV A, #01010101B
    MOV R3, #8
    LOOP2:
        MOV LED_PORT, A

```

```

RR A

CALL DELAY

DJNZ R3, LOOP2

RET

; Chu trình 4: Led tắt dần từ trong ra ngoài
CHUONG_TRINH_4:

MOV LED_PORT, #00011000B

CALL DELAY

MOV LED_PORT, #00111100B

CALL DELAY

MOV LED_PORT, #01111110B

CALL DELAY

MOV LED_PORT, #11111111B

CALL DELAY

RET

; Chu trình 5: Led tắt dần từ ngoài vào trong
CHUONG_TRINH_5:

MOV LED_PORT, #10000001B

CALL DELAY

MOV LED_PORT, #11000011B

CALL DELAY

MOV LED_PORT, #11100111B

CALL DELAY

MOV LED_PORT, #11111111B

CALL DELAY

RET

; Chu trình 6: Led chạy sáng - tắt, sáng - tắt.
CHUONG_TRINH_6:

MOV LED_PORT, #00H

```

```

CALL DELAY
MOV LED_PORT, #0FFH
CALL DELAY
MOV LED_PORT, #00H
CALL DELAY
MOV LED_PORT, #0FFH
CALL DELAY
RET

```

; Chu trình 7: 7 Led tắt, 1 Led sáng chạy từ trái qua phải

```

CHUONG_TRINH_7:
MOV A, #11111110B
MOV R1, #8
LOOP3:
MOV LED_PORT, A
RL A
CALL DELAY
DJNZ R1, LOOP3

```

```
RET
```

; Chu trình 1: 7 Led tắt, 1 Led sáng chạy từ phải qua trái

```

CHUONG_TRINH_8:
MOV A, #01111111B
MOV R1, #8
LOOP4:
MOV LED_PORT, A
RR A
CALL DELAY
DJNZ R1, LOOP4

```

```
RET
```

```
;***** CHUONG TRINH TAO TRE 0.5S *****
```

```
DELAY:
```

```
MOV TMOD, #01h
```

```
MOV R7, #10
```

```
LOOP_DELAY:
```

```
MOV TH0, #3CH
```

```
MOV TL0, #0B0H
```

```
SETB TR0
```

```
JNB TF0, $
```

```
CLR TR0
```

```
CLR TF0
```

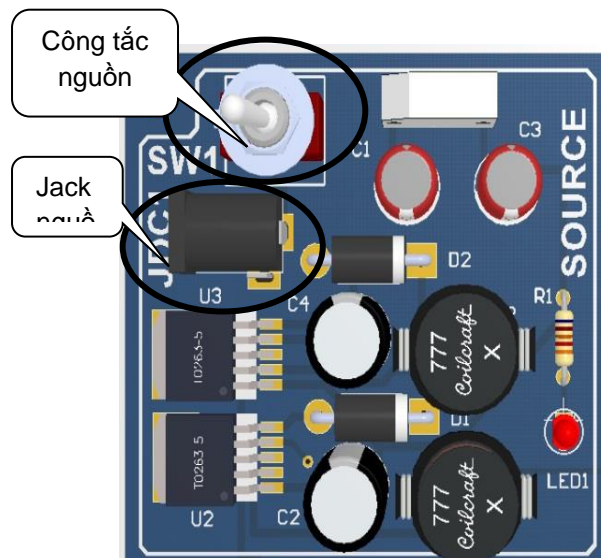
```
DJNZ R7, LOOP_DELAY
```

```
RET
```

```
END
```

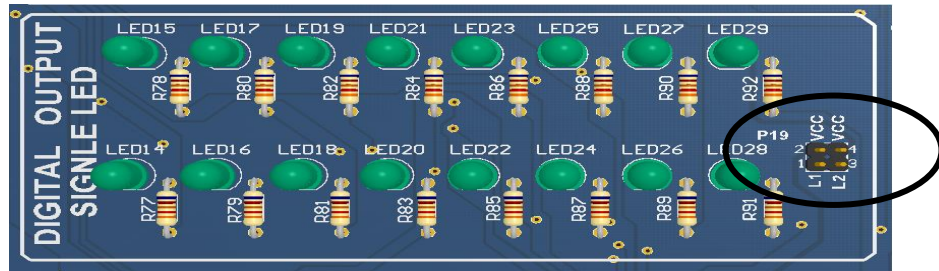
➤ Kết nối thiết bị:

- Cắm jack nguồn cho Module thí nghiệm 8051

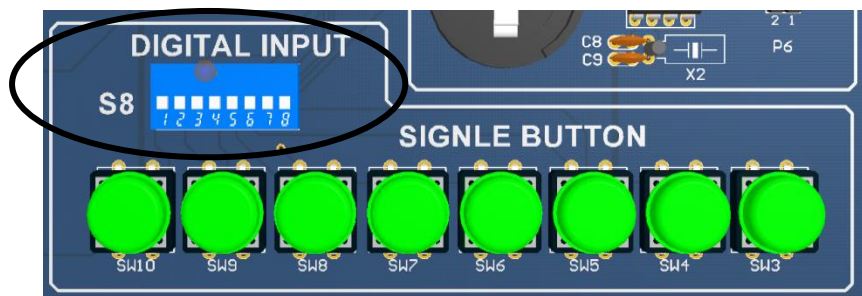


- Để công tắc ở vị trí OFF
- Cắm jump P19 kết nối dây Led với VCC
- Cắm Jump nối 1 với 2: Nối nguồn cho dây led 1 – Kết nối với vi điều khiển trung tâm qua PORT P0

- Cắm Jump nối 3 với 4: Nối nguồn cho dãy led 2 – Kết nối với vi điều khiển trung tâm qua PORT P2



- Chuyển các DipSW S1, S2, S3, S4, S5, S6, S7, S8 ở vị trí OFF
- Chuyển S8 về vị trí ON (Tương đương với việc kết nối các chân dữ liệu với PORT P1).



- Chuyển S1, S2, S3, S4, S5, S6, S7 ở vị trí OFF.

➤ **Nạp chip và kiểm thử**

2.3.2. Thiết kế và lập trình điều khiển led 7 thanh và nút bấm

- Viết chương trình để khi bấm mỗi nút Led 7 thanh sẽ sáng các giá trị từ 1 đến 8.
- Trình tự thực hành

➤ **Đọc sơ đồ nguyên lý để biết các tín hiệu điều khiển**

➤ **Các kiến thức cần có**

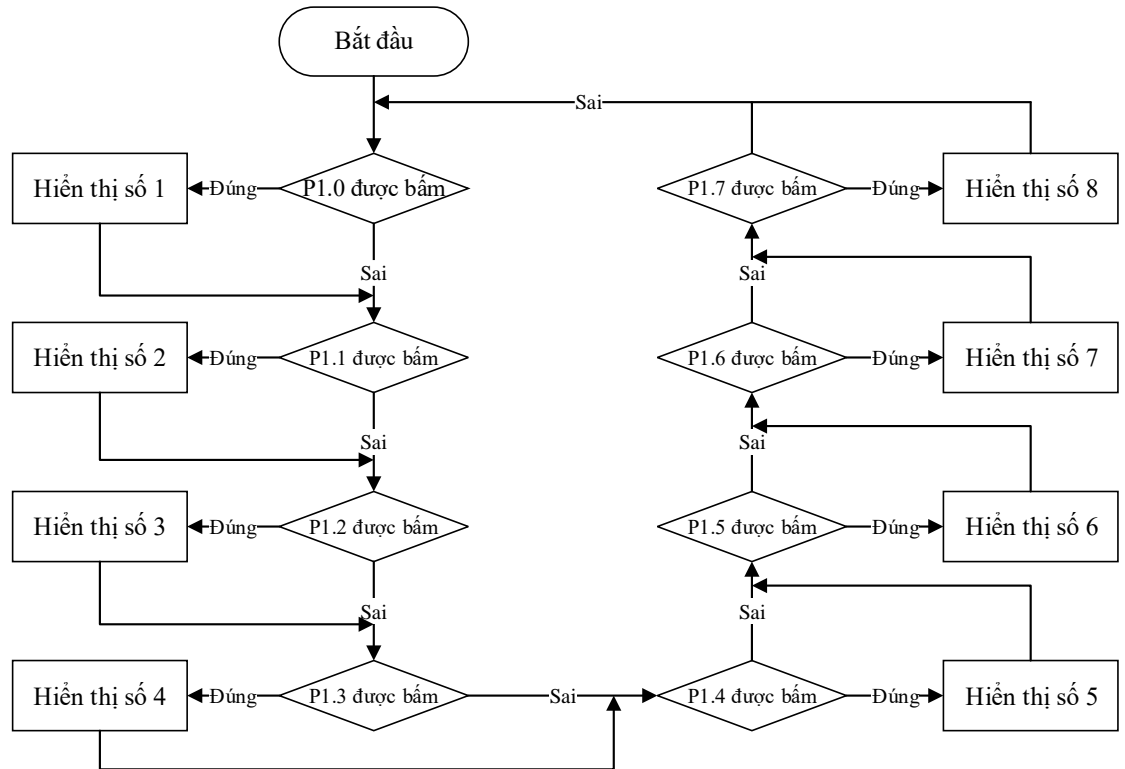
Trong các kiến thức cần chuẩn bị, cần chú ý nắm rõ các kiến thức:

- Cấu trúc, cách viết một chương trình bằng ngôn ngữ Assembly.
- Nguyên lý hoạt động của Led 7 thanh. Phân biệt cách nối chung Anode chung, chung Katot.
- Cách kiểm tra trạng thái của nút bấm.
- Cách viết, gọi một chương trình con.
- Các lệnh JB, JNB.

➤ **Phân tích cách giải**

Lần lượt kiểm tra trạng thái của từng nút bấm, nếu nút không được bấm thì kiểm tra nút kế tiếp, ngược lại, xuất ra công giá trị tương ứng với nút được bấm.

➤ **Sơ đồ thuật toán**



➤ **Lập trình:**

LED EQU P0

ORG 00H

; Cấp nguồn cho 3 led

MOV P2, #0FFH

; Xuất mã led của số 0 để hiển thị số 0

MOV LED, #3FH

MAIN:

; Nếu nút bấm ở chân P1.0 không được bấm thì nhảy

; đến nhãn để kiểm tra nút tiếp theo. Nếu không

; thì xuất mã Led của số 1 ra cổng.

JB P1.0, CHECK_1

MOV LED, #06H

CHECK_1:

JB P1.1, CHECK_2

MOV LED, #05BH

CHECK_2:

JB P1.2, CHECK_3

MOV LED, #4FH

CHECK_3:

JB P1.3, CHECK_4

MOV LED, #66H

CHECK_4:

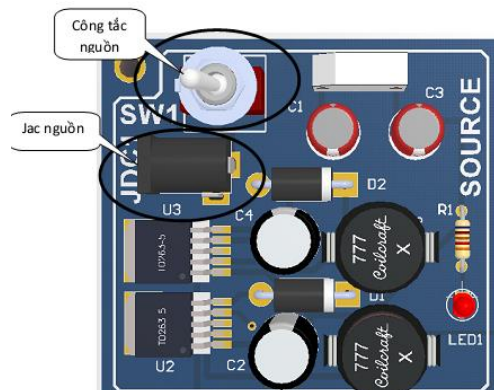
```

JB P1.4, CHECK_5
    MOV LED, #6DH
CHECK_5:
JB P1.5, CHECK_6
    MOV LED, #7DH
CHECK_6:
JB P1.6, CHECK_7
    MOV LED, #07H
CHECK_7:
JB P1.7, EXIT
    MOV LED, #7FH
EXIT:
JMP MAIN
END

```

➤ **Kết nối thiết bị:**

- Cắm Jack nguồn cho Module thí nghiệm 8051



- Để công tắc ở vị trí OFF
- Chuyển **S1** về vị trí ON (Tương đương với việc kết nối các chân dữ liệu với PORT P0 và các chân quét với PORT P2 của Led 7 thanh).



- Chuyển **S2, S3, S4, S5, S6, S7, S8** ở vị trí OFF.

➤ **Nạp chip và kiểm thử:**

2.4. Nội dung thực hành

Sinh viên thực hiện 2 trong 6 bài sau: (**Bài 1 là 1 trong 3 bài 1, 2, 3. Bài 2 là 1 trong 3 bài 4, 5, 6**)

Đề số 1. Viết chương trình cho phép mỗi khi nhấn một trong các nút bấm thì một Led đơn sẽ sáng tương đương với vị trí của nút bấm (*Nút 1 nhấn - Led tại vị trí 1 sáng, nút 2 nhấn - Led tại vị trí số 2 sáng, ..., nút 8 nhấn - Led tại vị trí 8 sáng.*)

Đề số 2. Viết chương trình điều khiển để mỗi lần bấm nút kết nối tới chân P1.0 thì các Led ở cổng P0 sáng theo quy luật sau:

- Lần 1: Led ở chân P0.0 sáng, các Led còn lại tắt
- Lần 2: Led ở chân P0.0 và P0.1 sáng, các Led còn lại tắt
- Lần 3: Led từ chân P0.0 đến P0.2 sáng, các Led còn lại tắt
- Lần 4: Led từ chân P0.0 đến P0.3 sáng, các Led còn lại tắt
- Lần 5: Led từ chân P0.0 đến P0.4 sáng, các Led còn lại tắt
- Lần 6: Led từ chân P0.0 đến P0.5 sáng, các Led còn lại tắt
- Lần 7: Led từ chân P0.0 đến P0.6 sáng, các Led còn lại tắt
- Lần 8: Led từ chân P0.0 đến P0.7 sáng

Đề số 3. Viết chương trình điều khiển theo yêu cầu sau:

- Nếu nút ở chân P1.0 được bấm thì các đèn Led ở cổng P0 sẽ sáng lần lượt từ chân P0.0 đến P0.7.
- Nếu nút ở chân P1.1 được bấm thì các đèn Led ở cổng P0 sẽ sáng lần lượt từ chân P0.7 về P0.0.

Đề số 4. Viết chương trình điều khiển để khi bấm một nút bất kỳ thì các Led 7 đoạn hiển các số đếm chạy từ 00 đến 50, khi nhả nút các Led sẽ chỉ hiển thị số tại thời điểm đó.

Đề số 5. Viết chương trình điều khiển để khi bấm nút ở chân P1.0 thì các Led 7 đoạn hiển thị các số đếm chẵn trong khoảng từ 00-99 và bấm nút ở chân P1.1 thì hiển thị các số đếm lẻ trong khoảng từ 00-99.

Đề số 6. Viết chương trình điều khiển để khi

- Bấm nút ở chân P1.0 thì các Led 7 đoạn hiển thị các số đếm từ 0 đến 9.
- Bấm nút ở chân P1.1 thì các Led 7 đoạn hiển thị các số đếm từ 1 đến 9.
- Bấm nút ở chân P1.2 thì các Led 7 đoạn hiển thị các số đếm từ 2 đến 9.

BÀI 3: GIAO TIẾP BÀN PHÍM VÀ ĐIỀU KHIỂN ĐỘNG CƠ

3.1. Mục đích của phần thực hành

Qua bài thí nghiệm này, giúp sinh viên làm chủ được:

- Nắm được cách thức làm việc của LED 7 thanh
- Biết lập trình giao tiếp bàn phím
- Biết cách điều khiển động cơ

3.2. Các kiến thức cần chuẩn bị

- Nắm vững tập lệnh của vi điều khiển MCS-51
- Nắm vững các kỹ thuật lập trình I/O, Timer/Counter
- Nắm vững các kỹ thuật thiết kế và lập trình hiển thị thiết bị ngoại vi với MCS-51

3.3. Bài tập thực hành mẫu

3.3.1. Thiết kế và lập trình điều khiển Động cơ và nút bấm

a. Viết chương trình để khi nhấn nút kết nối đến chân P1.0 động cơ quay theo chiều kim đồng hồ và nhấn nút bấm kết nối đến chân P1.1 động cơ dừng.

b. Trình tự thực hành

➤ **Đọc sơ đồ nguyên lý để biết các tín hiệu điều khiển**

➤ **Các kiến thức cần có**

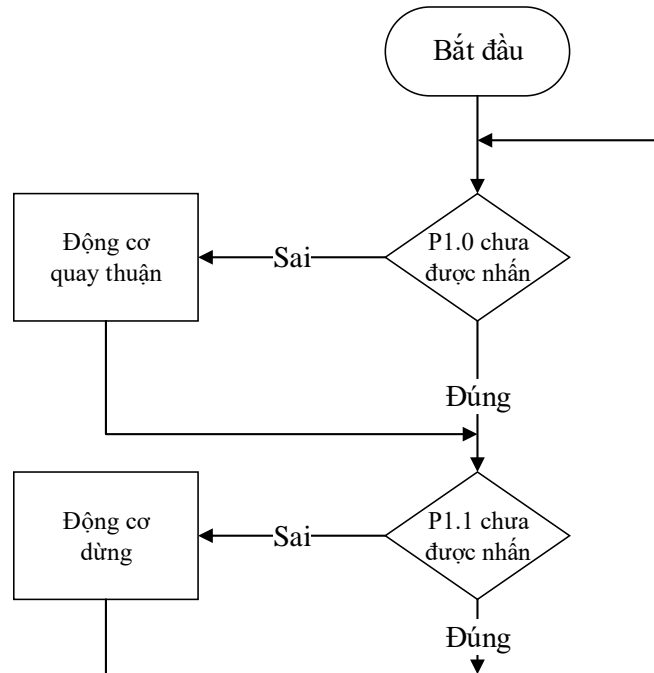
Để làm bài thực hành, cần chú ý:

- Cấu trúc, cách viết một chương trình viết bằng ngôn ngữ Assembly.
- Cách sử dụng động cơ, mô phỏng trên phần mềm Proteus.
- Cách kiểm tra trạng thái của nút bấm.
- Các lệnh JB, JNB, SETB và CLR.

➤ **Phân tích cách giải**

- Kiểm tra trạng thái của hai nút bấm nối ở hai chân P1.0 và P1.1. Nếu nút bấm ở chân P1.0 được nhấn thì nhảy đến nhãn quay thuận, tại đây, ta điều khiển động cơ quay thuận bằng cách điều khiển hai chân P2.0 và P2.1. Tương tự, nếu nút bấm ở chân P1.1 được nhấn thì nhảy đến nhãn quay ngược. Nếu không nút nào được bấm thì quay lại nhãn MAIN để tiếp tục kiểm tra trạng thái của các nút bấm.

➤ **Sơ đồ thuật toán**

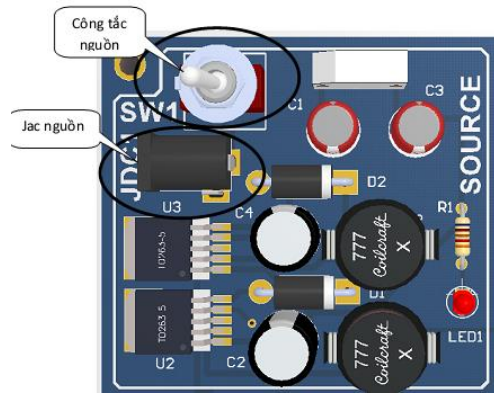


➤ **Lập trình:**

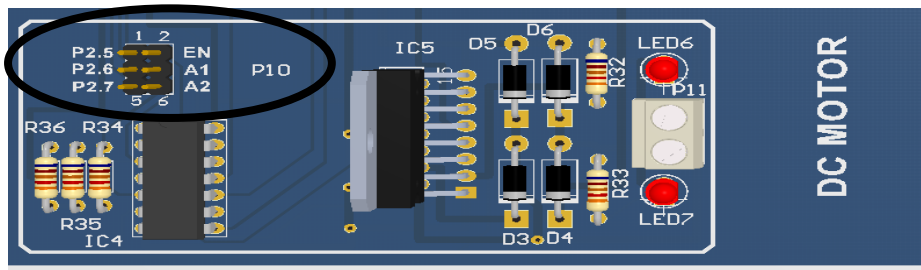
```
BUTTON_ONE EQU P1.0
BUTTON_TWO EQU P1.1
DC_LEFT EQU P2.6
DC_RIGHT EQU P2.7
ORG 00H
MAIN:
; Nếu P1.0 không được bấm thì kiểm tra nút tiếp theo, được
; bấm thì động cơ quay thuận
JB BUTTON_ONE, IGNORE_1
CLR DC_RIGHT
SETB DC_LEFT
IGNORE_1:
; Nếu P1.1 không được bấm thì thoát, được bấm thì dừng
; động cơ
JB BUTTON_TWO, EXIT
CLR DC_LEFT
CLR DC_RIGHT
EXIT:
JMP MAIN
END
```

➤ **Kết nối thiết bị:**

- Cắm Jack nguồn cho Module thí nghiệm 8051



- Để công tắc ở vị trí OFF
- Cắm chọn 3 Jump ở vị trí P10 để kết nối tín hiệu EN, A1, A2 với 3 chân P2.5, P2.6, P2.7 của vi điều khiển.



➤ **Nạp chip và kiểm thử**

3.3.2. Thiết kế và lập trình điều khiển Keypad và Led 7 thanh

- Viết chương trình hiển thị số thứ tự của nút được bấm trong bàn phím trên Led 7 thanh.
- Trình tự thực hành
 - **Đọc sơ đồ nguyên lý để biết các tín hiệu điều khiển**
 - **Các kiến thức cần có**
 Trong các kiến thức cần chuẩn bị, cần chú ý nắm rõ các kiến thức:
 - Cấu trúc, cách viết một chương trình viết bằng ngôn ngữ Assembly.
 - Nguyên lý hoạt động của Led 7 thanh, Keypad.
 - Cách kiểm tra trạng thái của nút bấm.
 - Cách viết, gọi một chương trình con.
 - Các lệnh JB, JNB và DPTR.
 - **Phân tích cách giải**
 Liên tục quét từng hàng của Keypad, kiểm tra từng phím bấm của hàng. Nếu phím nào được bấm thì xuất giá trị tương ứng của phím đó ra Led 7 thanh.

➤ Lập trình

; Địa chỉ bắt đầu tục hiện chương trình

ORG 00H

; Hiển thị số 0 lên Led 7 thanh

MOV R0,#0

CALL CHTT

SETB P2.0

MAIN:

;QUET HANG 1

MOV P1,#11111110B

JNB P1.4,PHIM7

JNB P1.5,PHIM8

JNB P1.6,PHIM9

JNB P1.7,PHIM10

;QUET HANG 2

MOV P1,#11111101B

JNB P1.4,PHIM4

JNB P1.5,PHIM5

JNB P1.6,PHIM6

JNB P1.7,PHIM11

;QUET HANG 3

MOV P1,#11111011B

JNB P1.4,PHIM1

JNB P1.5,PHIM2

JNB P1.6,PHIM3

JNB P1.7,PHIM12

;QUET HANG 4

MOV P1,#11110111B

JNB P1.4,PHIM13

JNB P1.5,PHIM14

JNB P1.6,PHIM15

JNB P1.7,PHIM16

JMP EXIT

PHIM1:

MOV R0,#1

JMP EXIT

PHIM2:

MOV R0,#2

JMP EXIT

PHIM3:

MOV R0,#3

JMP EXIT

PHIM4:

MOV R0,#4

JMP EXIT

```

PHIM5:
    MOV R0,#5
    JMP EXIT
PHIM6:
    MOV R0,#6
    JMP EXIT
PHIM7:
    MOV R0,#7
    JMP EXIT
PHIM8:
    MOV R0,#8
    JMP EXIT
PHIM9:
    MOV R0,#9
    JMP EXIT
PHIM10:
    MOV R0,#10
    JMP EXIT
PHIM11:
    MOV R0,#11
    JMP EXIT
PHIM12:
    MOV R0,#12
    JMP EXIT
PHIM13:
    MOV R0,#13
    JMP EXIT
PHIM14:
    MOV R0,#0
    JMP EXIT
PHIM15:
    MOV R0,#15
    JMP EXIT
PHIM16:
    MOV R0,#16
EXIT:
    CALL CTHT
LJMP MAIN

```

; Chương trình hiển thị giá trị trong R0 lên Led 7 thanh

```

CTHT:
    MOV DPTR,#MALED
    MOV A,R0
    MOVC A,@A+DPTR
    MOV P0,A
RET

```

; Chương trình tạo trễ

```

DELAY:
    MOV R1,#100

```



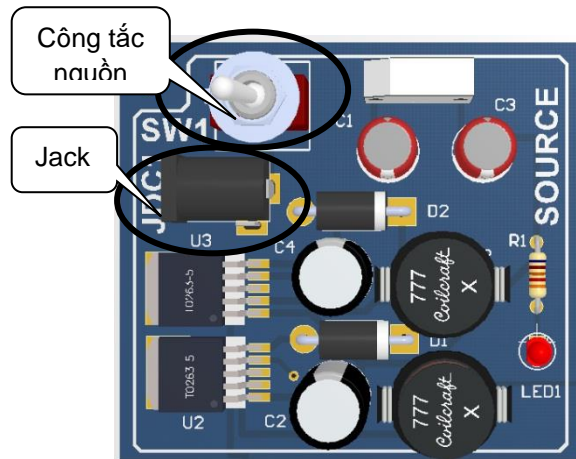
```

LAP:
    NOP
    NOP
    DJNZ R1, LAP
RET
; Mã Led 7 thanh
MALED: DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH
END

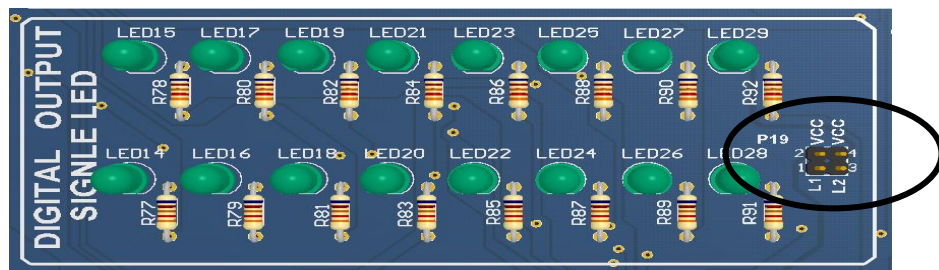
```

➤ **Kết nối thiết bị:**

- Cắm jack nguồn cho Module thí nghiệm 8051



- Để công tắc ở vị trí OFF
- Cắm Jump P19 kết nối dây Led với VCC
- Jump nối 1 với 2: Nối nguồn cho dây led 1 – Kết nối với vi điều khiển trung tâm qua PORT P0
- Jump nối 3 với 4: Nối nguồn cho dây led 2 – Kết nối với vi điều khiển trung tâm qua PORT P2



- Chuyển các DipSW S1, S2, S3, S4, S5, S6, S7, S8 ở vị trí OFF
- Chuyển S5 về vị trí ON (Tương đương với việc kết nối các chân dữ liệu với PORT P1).



- Chuyển **S1, S2, S3, S4, S6, S7, S8** ở vị trí **OFF**.

➤ **Nạp chip và kiểm thử**

3.4. Nội dung thực hành

Sinh viên thực hiện 1 trong 3 bài sau:

- Đề số 1. Viết chương trình để nếu 1 trong các nút bấm từ SW11 đến SW18 được bấm thì động cơ quay thuận, nếu 1 trong các nút từ SW19 đến SW26 được bấm thì động cơ quay ngược.
- Đề số 2. Viết chương trình để nếu 1 trong các nút bấm trên bàn phím được bấm thì động cơ quay thuận trong 5s rồi dừng.
- Đề số 3. Viết chương trình để nếu 1 trong các nút bấm trên bàn phím được bấm và giữ trong 1s thì động cơ quay thuận.

BÀI 4: HIỂN THỊ LED MATRIX VÀ LCD

4.1. Mục đích của phần thực hành:

Qua bài thí nghiệm này, giúp sinh viên nắm được:

- Phương pháp lập trình giao tiếp LED matrix và LCD

4.2. Các kiến thức cần chuẩn bị:

- Phương pháp lập trình vào ra số
- Phương pháp quét LED matrix
- Phương pháp lập trình giao tiếp LCD

4.3. Bài thực hành mẫu

4.3.1. Thiết kế và lập trình điều khiển Led Matrix

a. Viết chương trình hiển thị chữ “LAP TRINH” trên led Matrix.

b. Trình tự thực hành

➤ **Đọc sơ đồ nguyên lý để biết các tín hiệu điều khiển**

➤ **Các kiến thức cần có**

Trong các kiến thức cần chuẩn bị, cần chú ý nắm rõ các kiến thức:

- Cấu trúc, cách viết một chương trình viết bằng ngôn ngữ Assembly.
- Nguyên lý hoạt động của Led Matrix. Cách quét led để hiển thị ký tự bất kỳ.
- Cách viết, gọi một chương trình con.
- Cách viết chương trình tạo trễ

➤ **Phân tích cách giải**

➤ **Sơ đồ thuật toán**

➤ **Lập trình:**

```
ORG 00H
```

```
MAIN:
```

```
    MOV DPTR, #MASO1
```

```
    CALL CTHT
```

```
LJMP MAIN
```

```
CTHT:;CHUONG TRINH HIEN THI
```

```
    MOV R7, #64;Moi chu la 1 ma tran 8x8, co 8 chu nhu vay  
se la 8x8 = 64
```

```
    CHAYCHU:
```

```
        MOV R5, #100;toc do chay chu
```

```
    QUET:
```

```
        MOV R0, #00
```

```
        MOV R1, #01111111B
```

```
    LAPS:
```

```
        MOV P3, R1
```

```
        MOV A, R1
```

```

                RL A
                MOV R1,A
                MOV P3,R1
                MOV A,R0
                MOVC A,@A+DPTR
                MOV P2,A
                CALL DELAY
                INC R0
                CJNE R0,#8,LAPS;
            DJNZ R5,QUET
            INC DPTR
            DJNZ R7,CHAYCHU
RET
;-----Hien thi Chu LAP TRINH-----
MASO1:
DB    0FFH,0FFH,0H,7FH,7FH,7FH,7FH,0FFH    ;L
DB    0FFH,3H,0F5H,0F6H,0F6H,0F5H,3H,0FFH   ;A
DB    0FFH,0FFH,0H,0F6H,0F6H,0F6H,0F9H,0FFH ;P
DB    0FFH,0FFH,0FEH,0FEH,0H,0FEH,0FEH,0FFH ;T
DB    0FFH,0FFH,0H,0EEH,0D6H,0B6H,79H,0FFH  ;R
DB    0FFH,0FFH,7EH,0H,7EH,0FFH,0FFH,0FFH   ;I
DB    0FFH,0FFH,0H,0FDH,0FBH,0F7H,0H,0FFH   ;N
DB    0FFH,0FFH,0H,0F7H,0F7H,0F7H,0H,0FFH   ;H

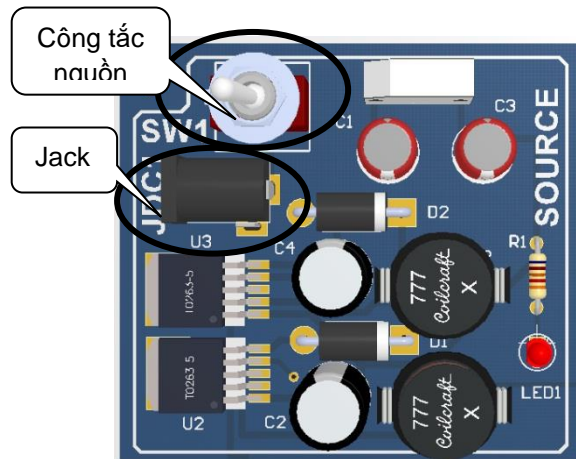
;-----Chuong trinh tao thoi gian tre-----

DELAY:
MOV R4,#50
LAPF:
    NOP
    NOP
    NOP
    DJNZ R4,LAPF
RET
END

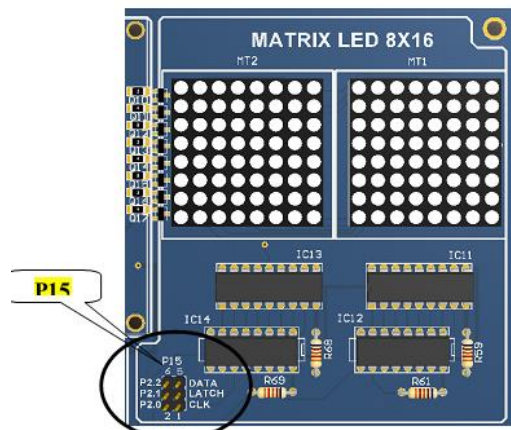
```

➤ **Kết nối thiết bị:**

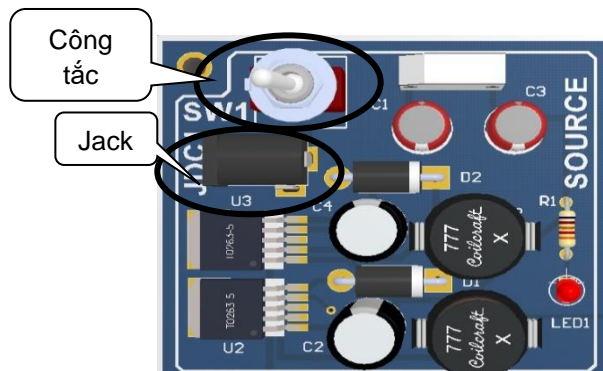
- Cắm jack nguồn cho Module thí nghiệm 8051



- Để công tắc ở vị trí OFF
- Chuyển S4 về vị trí ON (Tương đương với việc kết nối các chân dữ liệu với PORT P0).
- Chuyển S1, S2, S3, S5, S6, S7, S8 ở vị trí OFF.
- Cắm 3 jump của P15 để kết nối 3 chân điều khiển với mạch Vi điều khiển trung tâm (P2.0 với CLK; P2.1 với LATCH; P2.2 với DATA).



➤ **Nạp chip và kiểm thử**



4.3.2. Thiết kế và lập trình điều khiển LCD

- Viết chương trình hiển thị một chuỗi ký tự bất kỳ trên LCD.
- Trình tự thực hành

- **Đọc sơ đồ nguyên lý để biết các tín hiệu điều khiển**
- **Các kiến thức cần có**
 - Trong các kiến thức cần chuẩn bị, cần chú ý nắm rõ các kiến thức:
 - Cấu trúc, cách viết một chương trình viết bằng ngôn ngữ Assembly.
 - Nguyên lý hoạt động của LCD. Cách hiển thị ký tự bất kỳ lên LCD
 - Cách viết, gọi một chương trình con.
 - Cách viết chương trình tạo trễ
- **Phân tích cách giải**
- **Sơ đồ thuật toán**

➤ **Lập trình:**

```

RS EQU P3.0
RW EQU P3.1
ENABLE EQU P3.2

ORG 00H

MAIN:
    ACALL LCD_INIT
    MOV    DPTR, #MYDATA

    MOV R7, #0
    DISPLAY:
        MOV A, R7
        MOVC A, @A+DPTR
        ACALL DATAW
        ACALL DELAY
        INC R7
        CJNE R7, #16, DISPLAY

LJMP MAIN

LCD_INIT:
    MOV A, #38H
    ACALL W
    ACALL DELAY
    MOV A, #0EH
    ACALL W
    ACALL DELAY
    MOV A, #0CH
    ACALL W
    ACALL DELAY
    MOV A, #80H
    ACALL W

```

```

        ACALL DELAY
RET

W:
        MOV P2,A
        CLR ENABLE
        CLR RW
        SETB RS
        ACALL DELAY
        CLR RS
RET

DATAW:
        MOV P2,A
        SETB ENABLE
        CLR RW
        SETB RS
        ACALL DELAY
        CLR RS
RET

DELAY:
        MOV TMOD,#01H
            MOV TH0, #HIGH(-100)
            MOV TL0, #LOW(-100)
            SETB TR0
            JNB TF0,$
            CLR TR0
            CLR TF0
RET

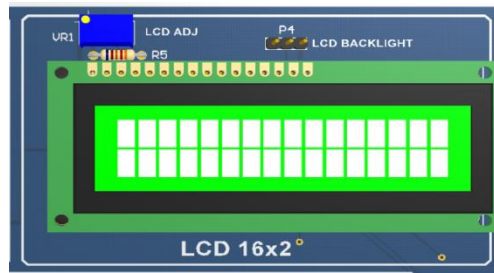
MYDATA: DB "DHCN THAI NGUYEN"

END

```

➤ **Kết nối thiết bị:**

- Cắm jack nguồn cho Module thí nghiệm 8051
- Để công tắc ở vị trí OFF
- Chuyển S2 về vị trí ON (Tương đương với việc kết nối các chân dữ liệu với PORT P0).
- Chuyển S1, S3, S4, S5, S6, S7, S8 ở vị trí OFF.



➤ **Nạp chip và kiểm thử**

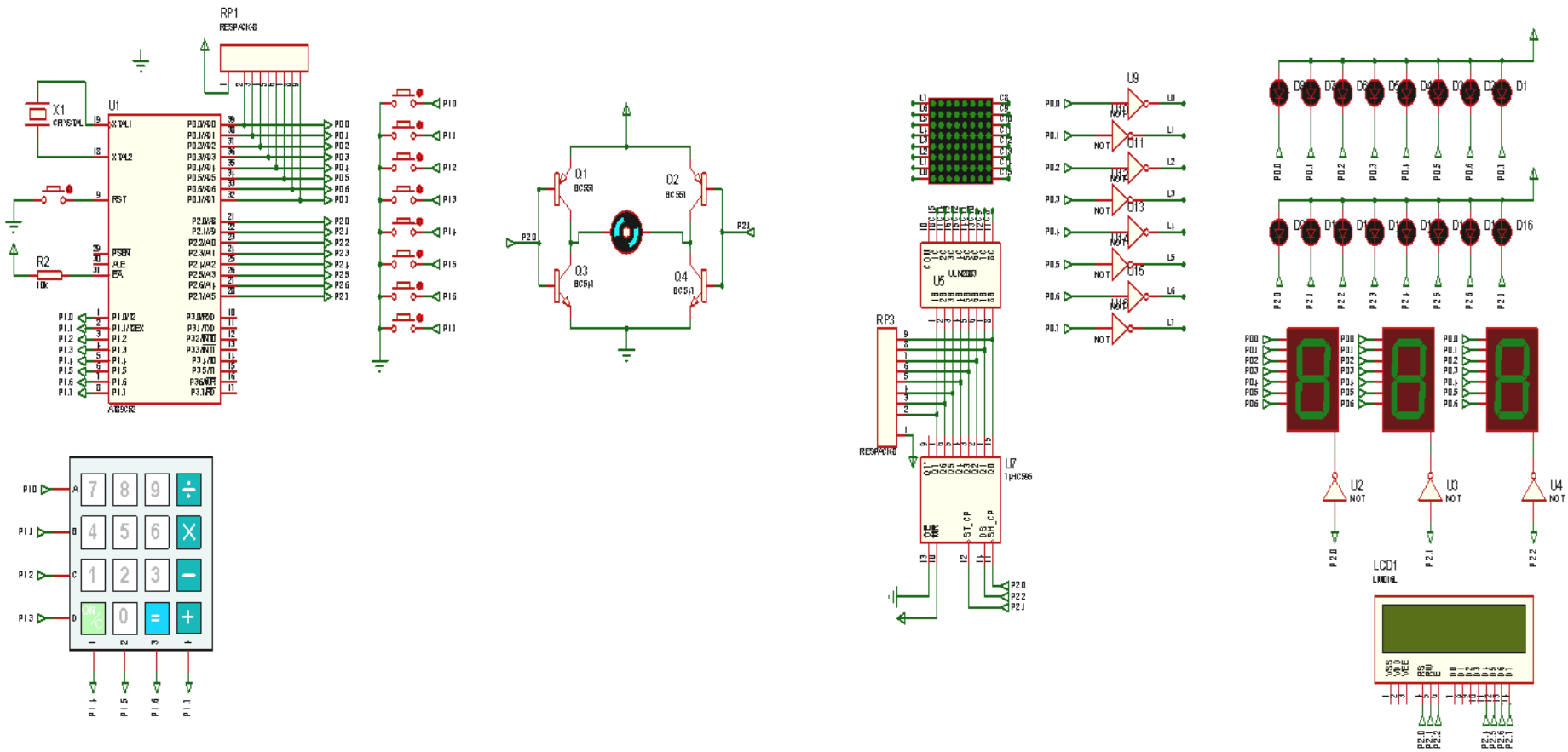
4.4. Nội dung thực hành

Sinh viên thực hiện nội dung sau: Viết chương trình hiển thị

- Chữ số cuối cùng trong mã số sinh viên của mình lên Led matrix
- Họ tên của mình lên dòng thứ nhất, mã số sinh viên lên dòng thứ 2 của LCD.

PHỤ LỤC

PHỤ LỤC 1: SƠ ĐỒ MẠCH NGUYÊN LÝ TỔNG THỂ

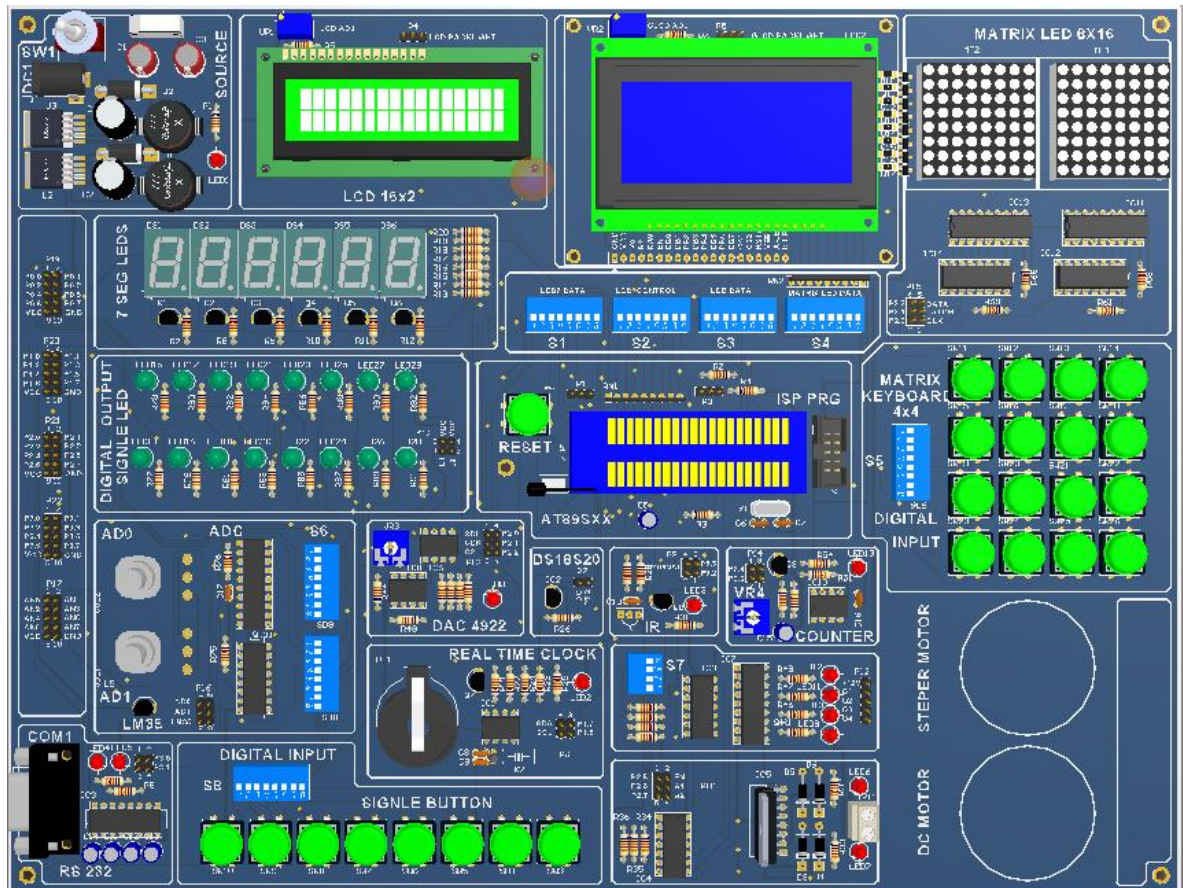


PHỤ LỤC 2: GIỚI THIỆU MODULE ĐÀO TẠO VI ĐIỀU KHIỂN 8051

1.1. Giới thiệu

Module đào tạo vi điều khiển 8051 là kit phát triển hoàn chỉnh, đầy đủ tính năng và dễ sử dụng cho ATMEL 89xxx Microcontroller. Với Module đào tạo vi điều khiển 8051 Người dùng không phải bận tâm về phần cứng, mà tập trung vào các ứng dụng phần mềm. Đây là một công cụ rất tuyệt vời đáp ứng mọi nhu cầu cho những kỹ sư giàu kinh nghiệm cũng như người mới bắt đầu.

Module đào tạo vi điều khiển 8051 thiết kế đặc biệt không dùng dây cắm. Các module được kết nối hoặc ngắt khỏi MCU một cách linh hoạt bằng Jumper hoặc DipSW rất tiện lợi và gọn gàng.



Hình 1: Sơ đồ tổng thể Module đào tạo vi điều khiển 8051.

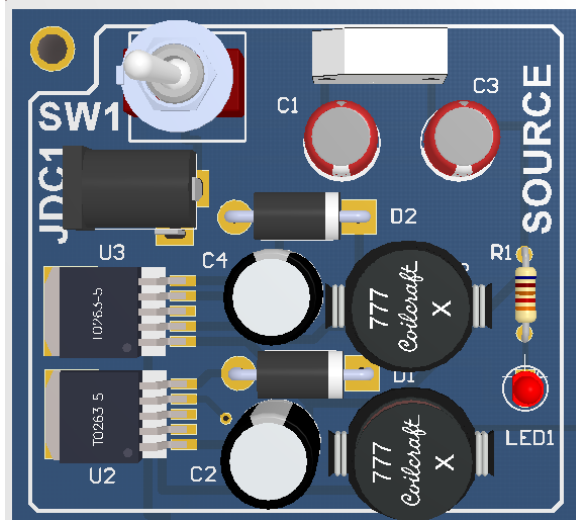
1.2. Nhưng tính năng của Kit vi điều khiển 8051

1. Power Supply: Sử dụng nguồn ngoài AC/DC 7 ~ 12V có công tắc nguồn ON/OFF
2. Có đường ISP programmer sẵn sàng sử dụng cho các mạch nạp ISP (Ví dụ: Mạch nạp USB 89Sxxx ISP programmer)
3. Suport 89xxx DIP40 (có thể dùng cho DIP20 bằng cách sử dụng Adapter chuyển chân)
4. LCD 16x2 4 BIT có biến trở chỉnh độ tương phản
5. Khối phím đơn gồm 8 nút bấm up/down
6. Ma trận bàn phím 4x4 (16 nút) bấm ma trận kết nối qua DipSW
7. RS-232 Communication giao tiếp truyền dữ liệu với PC
8. Khối thời gian thực (RTC DS1307 Real time clock) kết nối hoặc ngắt khỏi 8051 qua DIPSW
9. DS1820 Digital thermometer dùng để đo nhiệt độ từ -55°C to 125°C.
10. Điều khiển thu phát hồng ngoại - IR
11. Bộ chuyển đổi ADC 2 kênh – 8 bit
- 12.6 Led 7seg Anode chung multiplex mode. kết nối hoặc ngắt khỏi 8051 qua DIPSW
13. LED ma trận hiển thị số - ký tự
14. 16 led đơn anod chung
15. Điều khiển động cơ bước
16. Điều khiển động cơ một chiều
17. Graphic LCD 128x64 / Charracter LCD 16x2 8-bit có biến trở chỉnh độ tương phản
18. Xtal sử dụng socket dễ dàng thay đổi
19. RESET Switch - loại lớn cho phép dễ dàng Reset mạch bằng tay
20. Mạch in xuyên lớp chất lượng cao kích thước 350x250 mm
21. Có Mica bảo vệ bên dưới khối sợ ngắn mạch

1.3. Cấu hình chi tiết các khối module

1.3.1. Khối nguồn:

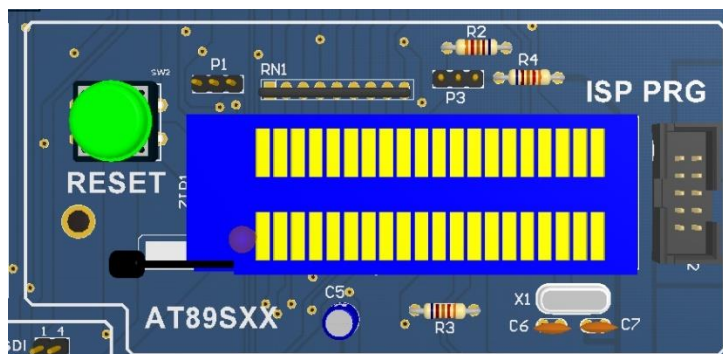
- Cung cấp nguồn DC5V cho toàn bộ các module của KIT VDK8051
- Cung cấp nguồn DC12V cho khối công suất của module điều khiển DC Motor và STEP Motor



Hình 2: Khối nguồn

1.3.2. Khối Vi điều khiển

- Sử dụng 89xxx DIP40 (có thể dùng cho DIP20 bằng cách sử dụng Adapter chuyển chân)
- Vi điều khiển AT89S52 lưu trữ, điều khiển hoạt động các Module theo chương trình đã được lập trình sẵn. Vi điều khiển AT89S52 có thể chạy bằng bộ nhớ trong (internal memory, dung lượng 8Kb) hoặc chạy bằng bộ nhớ ngoài (external memory) bằng cách thiết lập phần cứng trên KIT VDK8051
- Đường ISP programmer sẵn sàng sử dụng cho các mạch nạp ISP (Ví dụ: Mạch nạp USB 89Sxxx ISP programmer)



Hình 3: Khối vi điều khiển trung tâm

1.3.3. Khối hiển thị LCD 16x2:

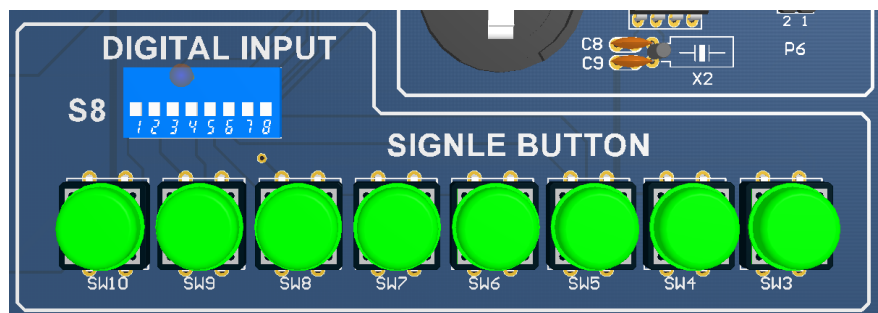
- Màn hình LCD 16x2 được thiết kế ở chế độ giao tiếp 4bit (Giao tiếp với VDK qua Port P2)
- Biến trở VR1 điều chỉnh độ sáng LCD



Hình 4: Khối hiển thị LCD 16x2

1.3.4. Khối phím đơn

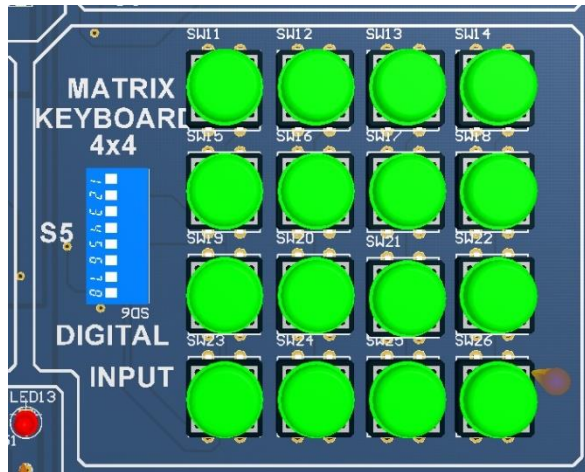
- 8 phím đơn giao tiếp trực tiếp với VDK qua PORT P1
- Thực hiện điều khiển kết nối thông qua Dip S8



Hình 5: Khối phím đơn

1.3.5. Khối phím ma trận

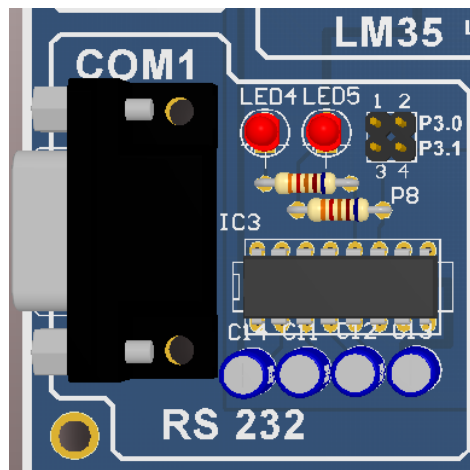
- Ma trận phím bấm là ma trận 4x4.
- Thực hiện điều khiển kết nối thông qua Dip S5



Hình 6: Khối ma trận bàn phím

1.3.6. Khối giao tiếp RS232

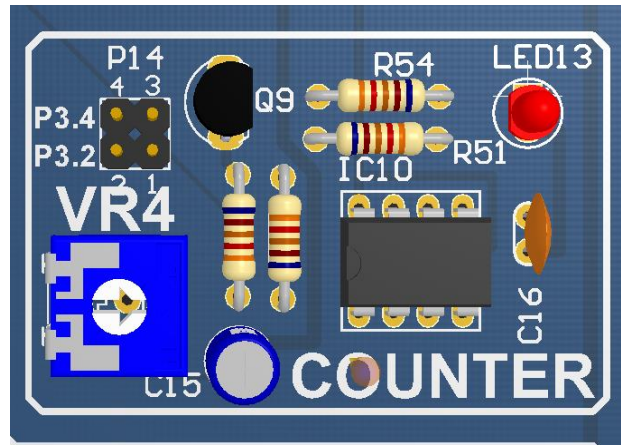
Giao tiếp giữa VDK với thiết bị ngoài thông qua chuẩn giao tiếp RS232 Ngoài ra còn có các chân cắm để kết nối giữa Module điều khiển trung tâm với các Module khác



Hình 7: Khối giao tiếp RS232

1.3.7. Khối tạo xung (Bộ đếm)

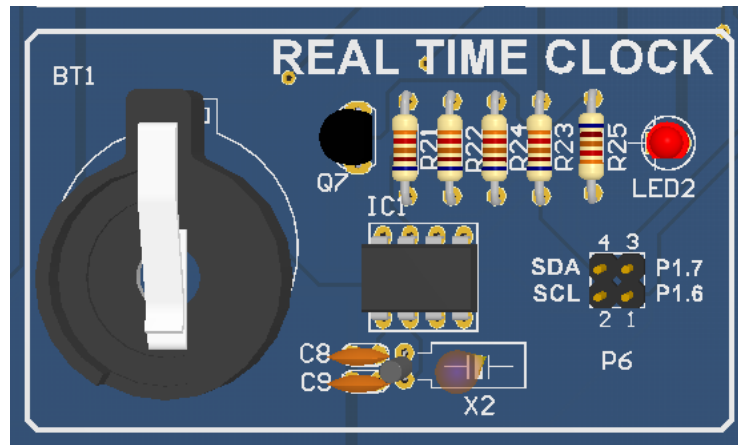
- Khối tạo xung vuông có thể điều chỉnh tần số bằng biến trở để tạo các tín hiệu như: ngắt theo chu kỳ, ... sử dụng làm nguồn ngắt ngoài cho VDK trong các ứng dụng lập trình External Interrupts, Counter
- Giao tiếp I²C với khối điều khiển trung tâm thông qua P3.4 và P3.2



Hình 8: Khối tạo xung vuông

1.3.8. Khối thời gian thực (Real Time Clock)

- Sử dụng DS1307 với nguồn Pin dự phòng
- Giao tiếp với IC thời gian thực qua chuẩn I²C sử dụng P1.7 và P1.6



Hình 9: Khối thời gian thực

1.3.9. Khối đo nhiệt độ - DS18S20

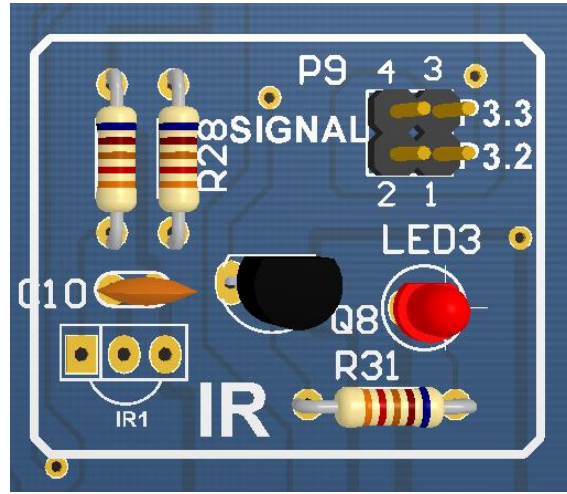
- Cho phép VDK giao tiếp với cảm biến nhiệt độ DS18S20 qua chuẩn giao tiếp ONE-WIRE



Hình 10: Khối đo nhiệt độ

1.3.10. Khối thu tín hiệu hồng ngoại – IR

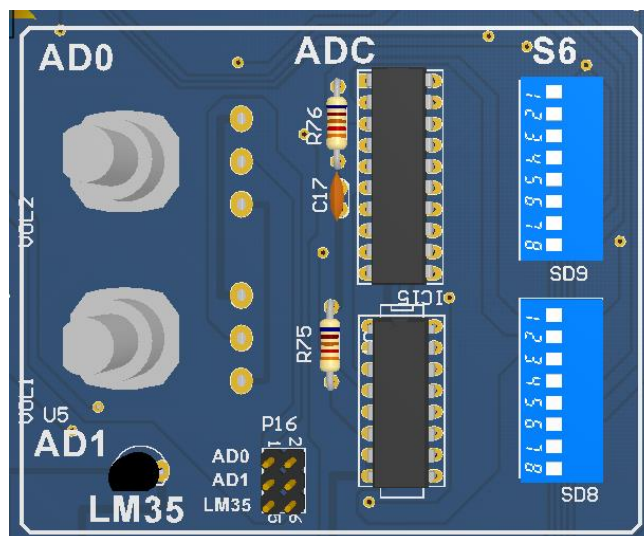
- Cho phép VDK nhận tín hiệu điều khiển từ xa dùng hồng ngoại để xử lý
- Kết nối với VDK trung tâm thông qua P3.2 và P3.3



Hình 11: Khối thu tín hiệu hồng ngoại

1.3.11. Khối chuyển đổi analog – digital (ADC)

- Thực hiện chức năng chuyển đổi 8 kênh tín hiệu Analog sang giá trị số 8bit.
- Trong Module có sẵn 2 biến trở VR1 , VR2 để điều chỉnh độ lớn tín hiệu Analog, cảm biến nhiệt độ LM35



Hình 12: Khối chuyển đổi ADC

1.3.12. Khối hiển thị LED 7 thanh

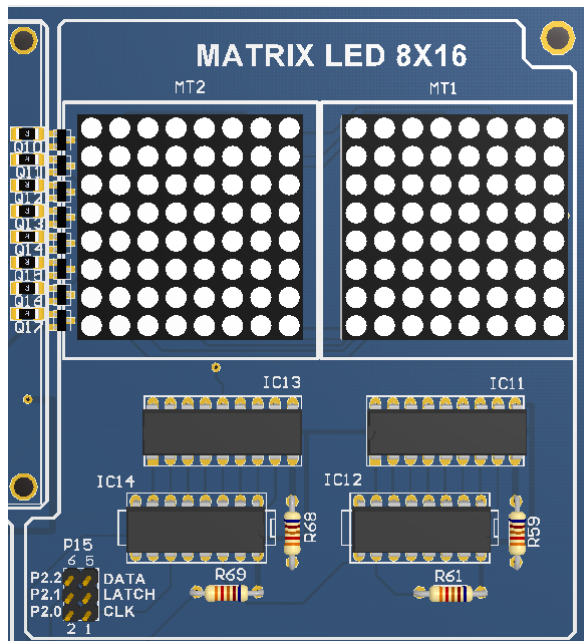
Bao gồm 6 led 7 thanh được mắc chung đường dữ liệu thực hiện các bài toán hiển thị số từ vi điều khiển.



Hình 13: Khối hiển thị LED 7 thanh

1.3.13. Khối led Matrix

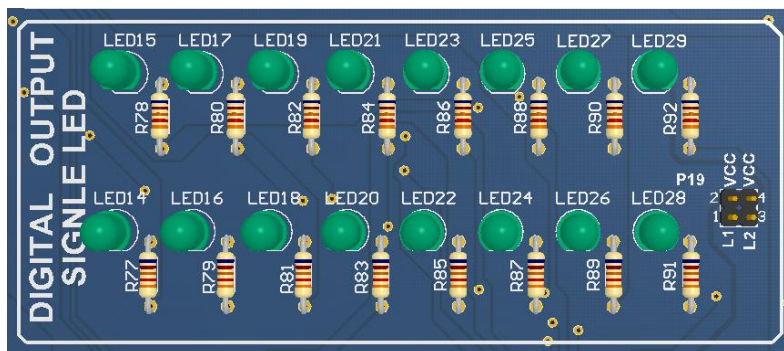
Gồm 2 led matrix được thiết kế sẵn để hiển thị theo phương pháp quét cột



Hình 14: Khối LED ma trận

1.3.14. Khối led đơn

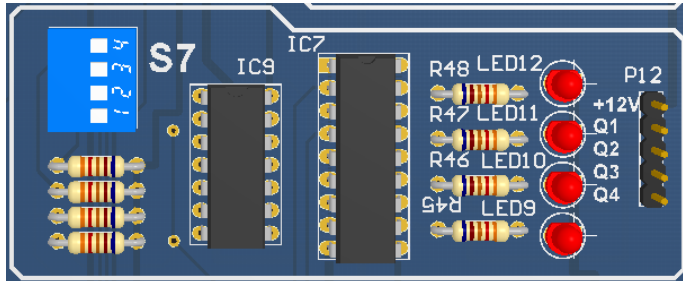
- Gồm 2 hàng ,mỗi hàng 8 led đơn mắc chung chân Anot giúp hiển thị trạng thái đầu ra vi điều khiển
- P19 kết nối dây led với VCC



Hình 15: Khối Led đơn

1.3.15. Khối điều khiển STEP Motor

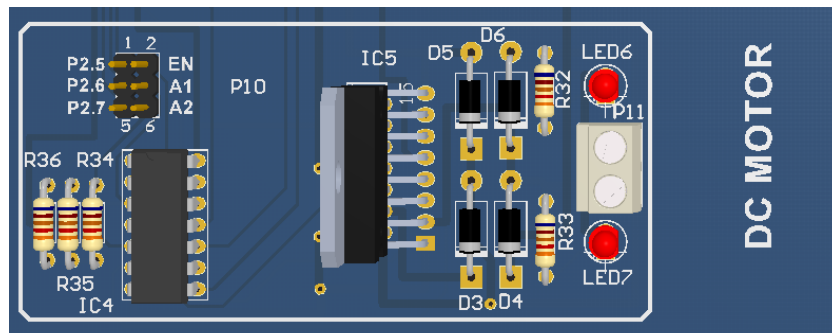
- Sử dụng mạch khuếch đại công suất cho động cơ bước 12V
- Kết nối với vi điều khiển qua chân P2.5, P2.6, P2.7 sử dụng Jump chọn.



Hình 16: Khối điều khiển động cơ bước

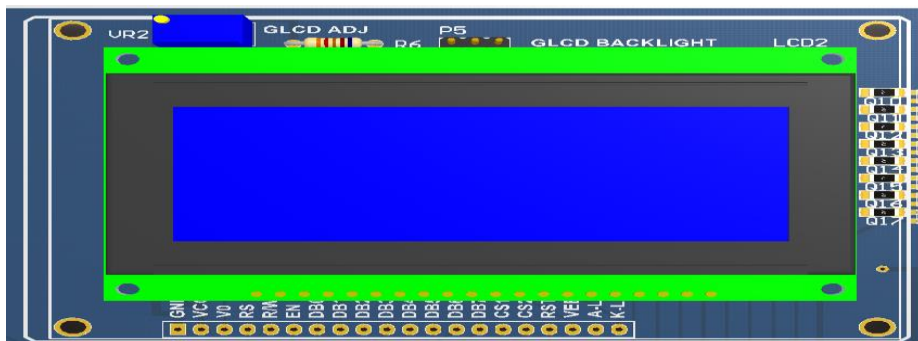
1.3.16. Khối điều khiển động cơ một chiều (DC motor)

Gồm 1 động cơ DC 12V và mạch điều khiển , mạch công suất chuyên dụng cho động cơ DC



Hình 17: Khối điều khiển động cơ một chiều

1.3.17. Khối hiển thị GLCD 128x64



PHỤ LỤC 3: HƯỚNG DẪN SỬ DỤNG MODULE ĐÀO TẠO VI ĐIỀU KHIỂN 8051

2.1. Hướng dẫn sử dụng Module đào tạo vi điều khiển 8051

2.1.1. Khi kết nối mạch

- Tên của từng port xuất nhập trên mô hình.
- Thứ tự các bit (từ LSB đến MSB) tại các port xuất nhập trên mô hình.
- Khi kết nối phải đảm bảo sao cho bit 0 của port vi điều khiển đúng vị trí bit 0 của đối tượng cần điều khiển.
- Khi kết nối đúng vị trí bit 0 thì các bit còn lại sẽ đúng vị trí.
- Tất cả các chương trình mẫu trong hệ thống này đều được kiểm tra rất kỹ theo đúng như kết nối mạch được trình bày.
- Nếu một yêu cầu nào đó không đúng thì hãy xem lại phần kết nối và chương trình.

2.1.2. Khi viết chương trình:

- Số 0 thường được đánh nhầm là chữ O.
- Thường đánh thiếu tiền tố # và hậu tố H đi kèm trong một số trường hợp.
- Sau lệnh END thì không còn một hàng hay một ký tự nào (kể cả ký tự trắng) nếu không chương trình biên dịch sẽ báo lỗi. Lỗi này có thể bỏ qua.
- Hãy dùng phím TAB để viết chương trình cho thẳng hàng. Điều này rất có ích cho bạn khi cần xem lại và kiểm tra lỗi chương trình được nhanh chóng.
- Nếu nhập một chương trình nào đó trong tài liệu mà chương trình chạy không đúng như yêu cầu thì hãy xem kỹ lại có đánh đầy đủ tất cả các lệnh trong chương trình hay chưa? Có thiếu sót gì không? Kết nối mạch có theo như hướng dẫn hay không? Tất cả các chương trình trong tài liệu hướng dẫn đã được chạy thử và luôn luôn đúng.

2.2. Nạp chương trình cho Vi điều khiển dùng ISP Prog

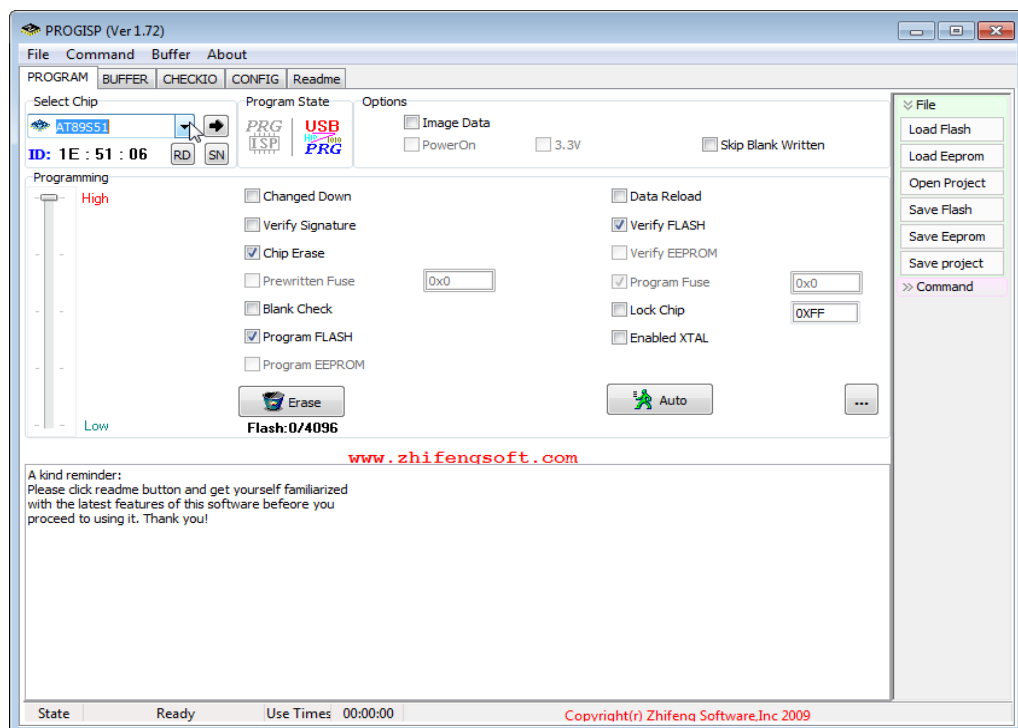
B1. Mở chương trình Prog ISP



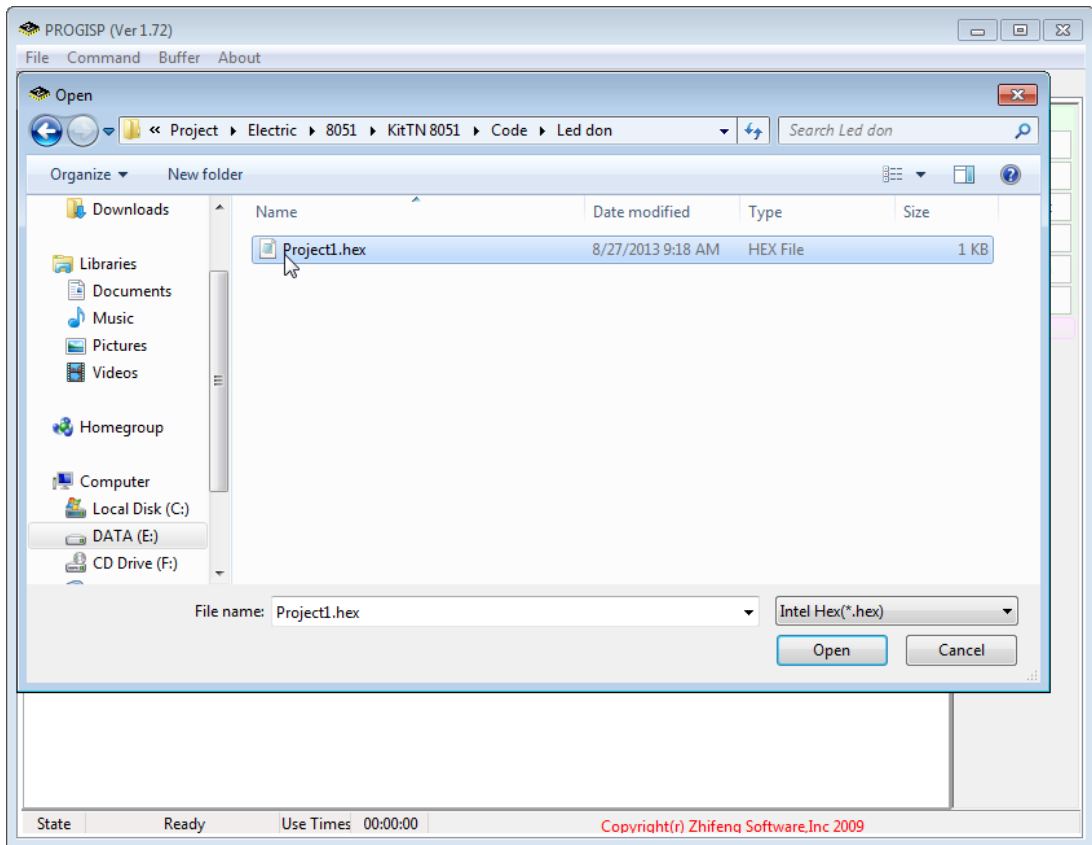
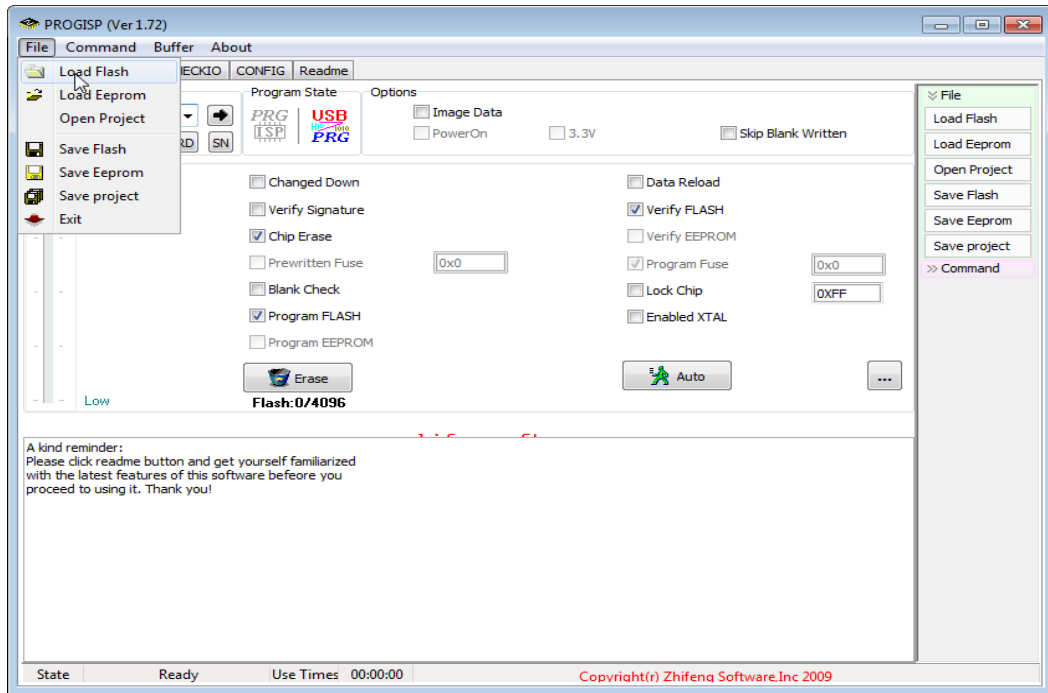
B2.Trong Tab PROGRAM/Select Chip : Chọn loại MCU cần nạp tương ứng với chương trình đã viết (Ở đây chọn AT89S51).

(Chú ý: Để nạp chương trình vào chip vi điều khiển cần cắm mạch nạp ISP vào main điều khiển và kết nối với máy tính ở cổng USB)

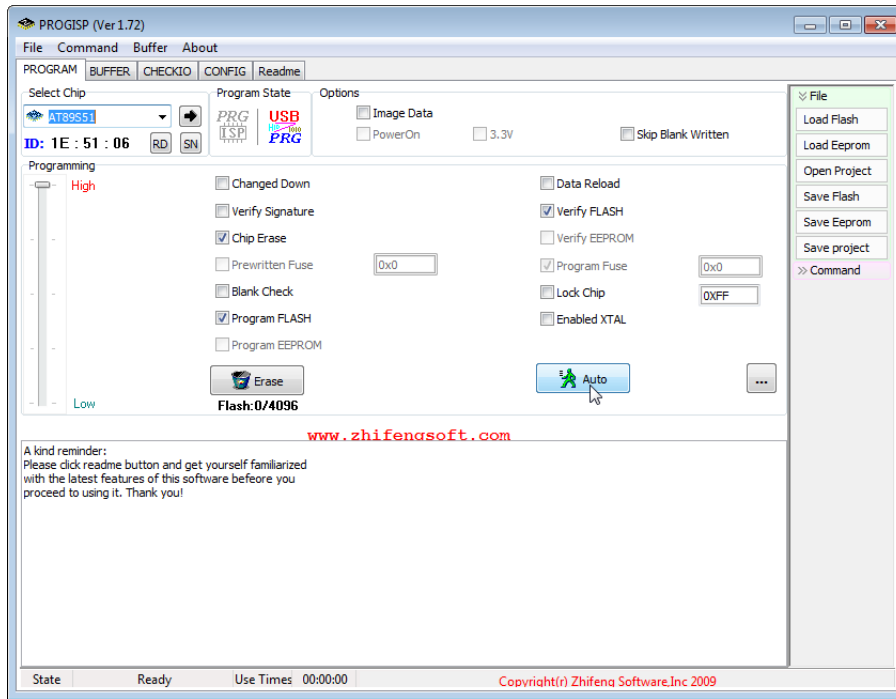
Trong phần Programming : Chọn Chip Erase,Program Flash,Verify FLASH



B3. Load File Hex cần nạp



B4. Nạp chương trình : Click Auto



B5. Chương trình báo nạp thành công