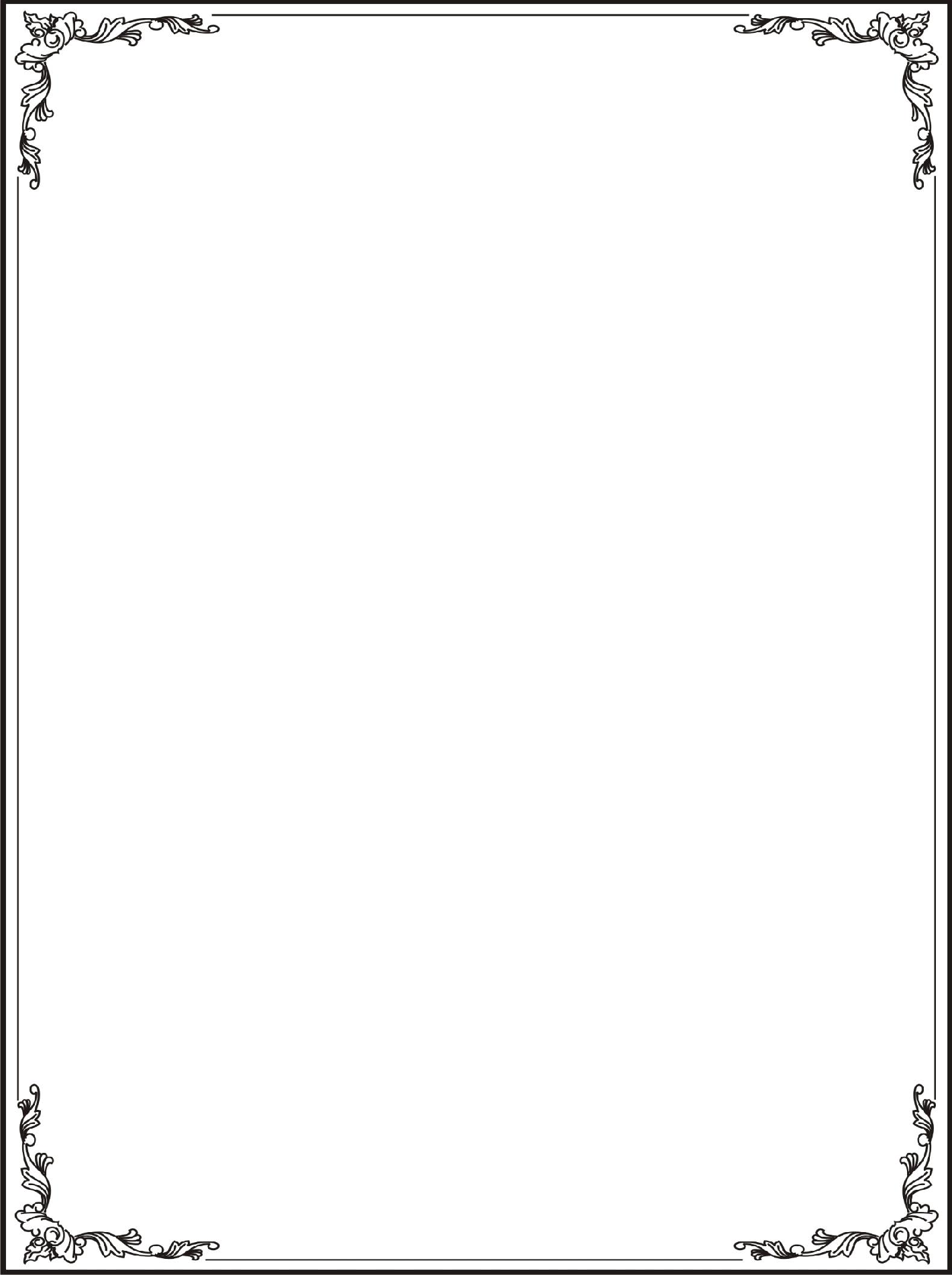
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**BÁO CÁO**

**TT Kiến Trúc Máy Tính**

**GVHD: Huỳnh Hoàng Hà**

**Mã học phần:**

**Ngày, tiết học:**

**HTSV: Mai Đông Thức**

**MSSV: 21139059**

**TP. Hồ Chí Minh - 11/2023**

Mục lục:

[**I.Giới Thiệu:** 3](#_Toc151082933)

[**I.1 Led ma trận là gì?** 3](#_Toc151082934)

[**I.2 Cấu tạo của led ma trận:** 3](#_Toc151082935)

[**II. Nguyên lý hoạt động:** 5](#_Toc151082936)

[**II.1 Nguyên lý hoạt động:** 5](#_Toc151082937)

[**II.2 Kỹ thuật quét led:** 5](#_Toc151082938)

[**III CODE** 6](#_Toc151082939)

[**III.1 Code tổng quang:** 6](#_Toc151082940)

[**III.2 Giải thích code:** 9](#_Toc151082941)

[**IV. Mô phỏng Proteus:** 12](#_Toc151082942)

[**IV.1 Sơ đồ mạch:** 12](#_Toc151082943)

[**IV.2 OUTPUT trên LED matrix 8x8:** 12](#_Toc151082944)

[**IV.2 Kit thực tế:** 13](#_Toc151082945)

# **I.Giới Thiệu:**

## **I.1 Led ma trận là gì?**

Led ma trận là ma trận điểm (mỗi điểm là một đèn LED) được xếp thành các hàng, cá cột nối tiếp nhau trên một màn hình lớn, độ phân giải thấp, được sử dụng để làm màn hình công nghiệp hoặc thương mại. Bên trong nó là ma trận diode hai chiều có cực dương sắp theo hàng và cực âm sắp theo cột. Có thể điều khiển từng điểm của led ma trận bằng cách điều khiển dòng điện đi qua mỗi cặp diode theo cột hoặc hàng. Loại ma trận này rất phổ biến trong sử dụng hiển thị thông tin, nó cho phép hiển thị hình ảnh và văn bản dạng tĩnh hoặc động. Bên dưới là hình 1 led ma trận:

A close-up of a dot-matrix

Description automatically generated

## **I.2 Cấu tạo của led ma trận:**

Để hiểu cấu trúc bên trong ma trận LED, trước tiên, hãy tìm hiểu một đèn LED đơn giản. Đèn LED có hai chân sử dụng nguồn DC để phát sáng. Trong ma trận này, có 8 hàng và 8 cột.

Chân dương của LED là chung cho các led trên cùng 1 hàng và chân âm của LED là chung cho các led trên cùng 1 cột. Tất cả các chân led hàng sẽ kết nối với cực dương của nguồn điện và nối đất của nguồn điện sẽ đi đến các chân led cột.

Bật nguồn cho một hàng và một cột sẽ chỉ bật một đèn LED duy nhất. Ví dụ, nguồn điện áp cấp cho hàng 3 và cột 3 thì đèn led trong ma trận 3 × 3 sẽ bật.

A diagram of a circuit

Description automatically generated

Trong ma trận, toàn bộ hàng có thể bật bằng cách cấp nguồn nhưng tất cả các cột phải có đầu nối mass. Quá trình tương tự sẽ diễn ra với cột. Ma trận LED có trên thị trường dưới dạng module được thiết kế sẵn nhưng chúng có chi phí cao hơn so với loại tự thiết kế.

# **II. Nguyên lý hoạt động:**

## **II.1 Nguyên lý hoạt động:**

Khi có một tín hiệu điều khiển ởcột và hàng, các chân Anode của cácled trên cột tương ứng đƣợc cấp điện áp cao, đồng thời các chân Cathode củacác led trên hàng tương ứng được cấp điện áp thấp. Tuy nhiên lúc đó chỉ có một led sáng, vì nó có đồng thời điện thếcao trên Anode và điện thế thấp trên Cathode. Như vậy khi có một tín hiệu điều khiển hàng và cột, thi tại một thờiđiểm chỉcó duy nhất một led tại chỗgặp nhau của một hàng và cột là sáng.Các bảng quang báo với số lượng led lớn hơn cũng được kết nối theo câu trúc như vậy

Trong trường hợp ta muốn cho sáng đồng thời một số led rời rạc trên ma trận, để hiển thị một ký tự nào đó, nếu trong hiển thị tĩnh ta phải cấp áp cao cho Anode và áp thấp cho Cathode, cho các led tương ứng mà ta muốn sáng. Nhưng khi đó một sốled ta không muốn cũng sẽ sáng, miễn là nó nằm tại vịtrí gặp nhau của các cột và hàng mà ta cấp nguồn. Vì vậy trong điều khiển led ma trận ta không thể sử dụng phương pháp hiện thị tĩnh mà phải sử dụng phương pháp quét ( hiển thị động ), có nghĩa là ta phải tiến hành cấp tín hiệu điều khiển theo dạng xung quét trên các hàng và cột có led cần hiển thị. Để đảm bảo cho mắt nhìn thấy các led không bịnháy, thì tần sốquét nhỏ nhất cho mỗi chu kỳ là khoảng 20hz(50ms). Trong lập trình điều khiển led ma trận bằng vi xử lý ta cũng phải sửdụng phương pháp quét như vậy.Ma trận led có thể là loại chỉ hiển thi được một màu hoặc hiển thị được 2 mày trên một điểm, khi đó led có số chân ra tƣơng ứng: đối với ma trận led8x8 hiển thị một màu, thi sốchân ra là 16, trong đó 8 chân dùng để điều khiển hàng và 8 chân còn lại dùng để điểu khiển cột. Đối với loại 8x8 có 2 màu thi số chân ra của led là 24 chân, trong đó 8 chân dùng để điều khiển hàng ( hoặc cột ) chung cho cả hai màu, 16 chân còn lại thi 8 chân dùng để điểu khiển hàng ( hoặc cột) màu thứ nhất, 8 chân còn lại dùng để điều khiển màu thứ 2

## **II.2 Kỹ thuật quét led:**

Phương pháp này khá đơn giản, đầu tiển điều khiển 8 con LED thứ nhất theo hàng ngang hoặc theo cột dọc. Sau đó tắt hết đi -> điều khiển tiếp 8 con LED tiếp theo -> tắt đi -> điều khiển tiếp 8 con tiếp theo …… cứ như thế cho đến hết 64 con LED thì lặp lại. Hãy nhớ rằng tốc độ bật tắt là rất nhanh, mắt người do có sự lưu ảnh ở mắt sẽ tự gép lại thành 1 hình ảnh hoàn chỉnh.

Khoảng thời gian mà bạn bật sáng 8 con LED đầu tiên cho tới khi bật sáng 8 con LED cuối cùng là 1 chu kì quét. Lấy nghịch đảo sẽ có được tần số quét. Theo tham khảo và thực tiễn, thì tần số để quét đi quét lại hết 1 tấm matrix là 60Hz thì mắt người nhìn sẽ thấy ổn ( không bị nháy nháy hay khó chịu – giống như ảnh trên được quét với tần số 20Hz). Tuy nhiên hầu hết các camera hiện nay có tốc độ chụp khá cao nên khi dùng camera quay lại vẫn sẽ thấy sự quét của màn hình led matrix. Do vậy tần số quét khuyến khích là 120Hz.

# **III CODE**

## **III.1 Code tổng quang:**

#include<reg51.h>

#include<intrins.h>

sbit SRCLK=P3^6;

sbit RCLK=P3^5;

sbit SER=P3^4;

#define COMMONPORTS     P0

unsigned char code TAB[8]  = {0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

unsigned char code CHARCODE[18][8]=

{

{0x00,0x00,0x3e,0x41,0x41,0x41,0x3e,0x00}, //0

{0x00,0x00,0x00,0x00,0x21,0x7f,0x01,0x00}, //1

{0x00,0x00,0x27,0x45,0x45,0x45,0x39,0x00}, //2

{0x00,0x00,0x22,0x49,0x49,0x49,0x36,0x00}, //3

{0x00,0x00,0x0c,0x14,0x24,0x7f,0x04,0x00}, //4

{0x00,0x00,0x72,0x51,0x51,0x51,0x4e,0x00}, //5

{0x00,0x00,0x3e,0x49,0x49,0x49,0x26,0x00}, //6

{0x00,0x00,0x40,0x40,0x40,0x4f,0x70,0x00}, //7

{0x00,0x00,0x36,0x49,0x49,0x49,0x36,0x00}, //8

{0x00,0x00,0x32,0x49,0x49,0x49,0x3e,0x00}, //9

{0x00,0x00,0x7F,0x48,0x48,0x30,0x00,0x00}, //P

{0x00,0x00,0x7F,0x48,0x4C,0x73,0x00,0x00}, //R

{0x00,0x00,0x7F,0x49,0x49,0x49,0x00,0x00}, //E

{0x00,0x00,0x3E,0x41,0x41,0x62,0x00,0x00}, //C

{0x00,0x00,0x7F,0x08,0x08,0x7F,0x00,0x00}, //H

{0x00,0x00,0x00,0xFF,0xFF,0x00,0x00,0x00}, //I

{0x00,0x7F,0x10,0x08,0x04,0x7F,0x00,0x00}, //N

{0x7C,0x48,0x48,0xFF,0x48,0x48,0x7C,0x00}  //��

};

void delay(unsigned int time)

{

  unsigned int i,j;

  for(i=0;i<time;i++)

    for(j=0;j<121;j++);

}

void Hc595SendByte(unsigned char dat)

{

    unsigned char a;

    SRCLK=0;

    RCLK=0;

    for(a=0;a<8;a++)

    {

        SER=dat>>7;

        dat<<=1;

        SRCLK=1;

        \_nop\_();

        \_nop\_();

        SRCLK=0;

    }

    RCLK=1;

    \_nop\_();

    \_nop\_();

    RCLK=0;

}

void main()

{

    unsigned char tab, j;

    unsigned int  i;

    while(1)

    {

        for(i= 0; i<50; i++

        {

            for(tab=0;tab<8;tab++)

            {

                Hc595SendByte(0x00);

                COMMONPORTS = TAB[tab];

                Hc595SendByte(CHARCODE[j][tab]);

                delay(2);

            }

        }

        j++;

        if(j == 18)

        {

            j = 0;

        }

    }

}

## **III.2 Giải thích code:**

A screen shot of a computer screen

Description automatically generated

Trong vòng lặp for , biến dat sẽ được dịch sang trái sau mỗi vòng lặp. Điều này có nghĩa là mỗi bit của biến dat sẽ được chuyển sang vị trí kế tiếp.

Sau đó, bit có trọng số cao nhất của biến dat (bit 7) sẽ được chuyển vào đầu vào SER (Serial Data Input) của thanh ghi.

Tiếp theo, tín hiệu SRCLK (Shift Register Clock) sẽ được gán giá trị 1 để thực hiện chuyển bit hiện tại trong SER vào bộ nhớ 8 bit của thanh ghi. Khi SRCLK là 1, dữ liệu sẽ được chuyển từ SER vào thanh ghi. Sau đó, SRCLK sẽ được gán giá trị 0 để chuẩn bị cho lần chuyển bit tiếp theo.

Sau khi đã chuyển 8 bit từ SER vào thanh ghi, tín hiệu RCLK (Register Clock) sẽ được gán giá trị 1 để thực hiện chốt dữ liệu trong thanh ghi. Khi RCLK là 1, dữ liệu trong thanh ghi sẽ được chốt, và các giá trị từ Q0 đến Q7 sẽ được đưa ra qua các ngõ ra tương ứng. Sau đó, RCLK sẽ được gán giá trị 0 để chuẩn bị cho lần chốt dữ liệu tiếp theo.

Quá trình này sẽ lặp lại trong vòng lặp for, cho phép dịch và chuyển dữ liệu từ biến dat vào thanh ghi 8 bit sử dụng các tín hiệu SRCLK và RCLK.

A black and white text on a black background

Description automatically generated

Mảng TAB chứa các cột trong led ma trận 8x8, chứa các bit 0.

A screenshot of a computer code

Description automatically generated

Mảng CHARCODE chứa các mã khi kết hợp với mảng TAB sẽ ra kết quả là các kí tự từ 0-9, A-N.

A computer screen shot of a program code

Description automatically generated

Trong hàm main():

Trong vòng lặp for, giá trị của mảng COMMONPORTS sẽ được gửi đến port 0. Để thực hiện việc gửi này, chúng ta sử dụng hàm Hc595SendByte() để gửi một chuỗi giá trị của một ký tự.

Khi kết hợp COMMONPORTS với các giá trị gửi bằng hàm Hc595SendByte(), nếu bit tương ứng trong mảng COMMONPORTS là 0 và bit CHARCODE[j][tab] là 1, thì đèn LED sẽ được bật.

Điều này có nghĩa là khi bit COMMONPORTS tại vị trí tương ứng là 0 và bit CHARCODE[j][tab] tại vị trí tương ứng là 1, thì ta có điều kiện để bật đèn LED.

Quá trình này sẽ được lặp lại trong vòng lặp for, để kiểm tra và điều khiển LED dựa trên giá trị của mảng COMMONPORTS và giá trị gửi bằng hàm Hc595SendByte().

# **IV. Mô phỏng Proteus:**

## **IV.1 Sơ đồ mạch:**

A diagram of a computer

Description automatically generated

Hình 6 – Mạch mô phỏng.

## **IV.2 OUTPUT trên LED matrix 8x8:**

A diagram of a circuit board

Description automatically generatedA diagram of a circuit board

Description automatically generated

A diagram of a circuit board

Description automatically generatedA diagram of a circuit board

Description automatically generated

## **IV.2 Kit thực tế:**

