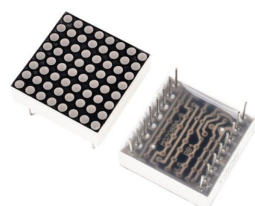


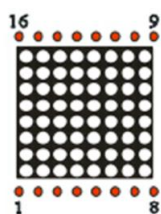
# 点阵模块

## 硬件概述



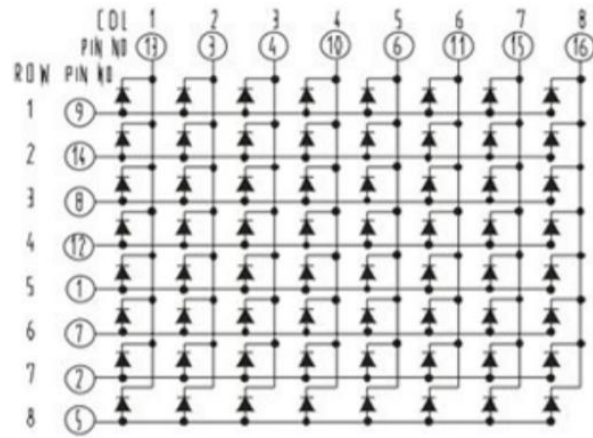
LED 点阵模块指的是利用封装 8\*8 的模块组合点元板形成模块，它连接微处理器与 8 位数字的 7 段数字 LED 显示，也可以连接条线图显示器或者 64 个独立的 LED。其上包括一个片上的 B 型 BCD 编码器、多路扫描回路，段字驱动器，而且还有一个 8\*8 的静态 RAM 用来存储每一个数据。只有一个外部寄存器用来设置各个 LED 的段电流。每个数据可以寻址在更新时不需要改写所有的显示。LED 点阵显示模块可显示汉字、图形、动画及英文字符等；显示方式有静态、横向滚动、垂直滚动和翻页显示等。

## 引脚定义

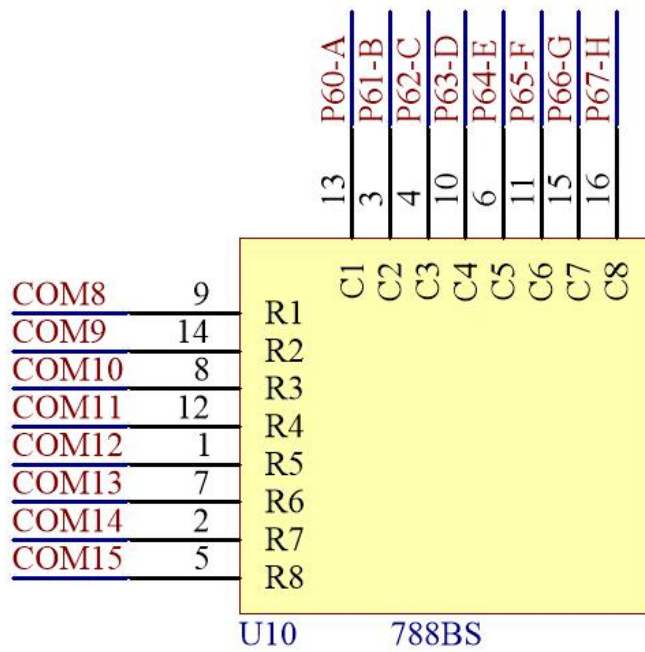


序号	符号	管脚名	功能描述
1	1、2、5、7、8、9、12、14	COM	公共正极
2	3、4、6、10、11、13、15、16	负极	电源负极

## 点阵内部连线图



## 电路原理图



## 图形化模块

1. 点阵初始化。

点阵初始化

2. 点阵扫描回调函数。

## 点阵扫描回调函数

3. 点阵清屏函数。

## 点阵清屏

4. 点阵显示数字。

点阵屏显示数字

5. 点阵显示字符串。

点阵屏显示字符串

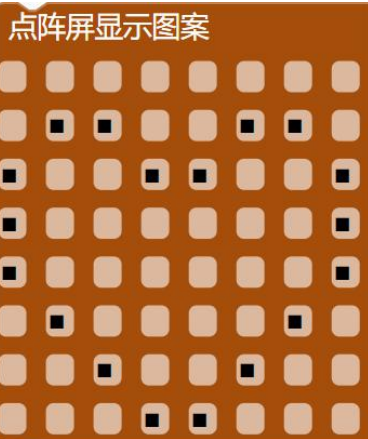
6. 设置点阵在第几行，第几列显示点。

点阵屏显示点在第  行第  列

7. 设置点阵在第几行，第几列清除点。

点阵屏清除点在第  行第  列

8. 设置点阵显示图案。



9. 点阵更新显示缓存

点阵更新显示缓存

10. 设置点阵显示自带图案

点阵屏显示图案

## 示例代码 1

设置点阵在第 0 行，第 0 列显示一个点。



### 示例代码 2

设置点阵显示数字 123。



### 示例代码 3

设置点阵显示字符串 abcd。



### 示例代码 4

设置点阵显示大爱心，300 毫秒后切换小爱心，依次循环。



### 示例代码 5

设置数组数据 0xe3,0xc1,0x81,0x03,0x03,0x81,0xc1,0xe3, 点阵更新显示数据从 mylist 数组中调用数据。



### 调用函数代码

```
void matrix_init()//点阵屏初始化, 参数无
```

```
void matrix_display_num(uint8 data)
```

```
// 点阵屏显示数
```

```
void matrix_scan_callback()//点阵屏扫描回调函数, 参数无
```

```

matrix_set_pixel(uint8 x,uint8 y,uint8 state)

// 点阵屏显示点.

// 参数: x 坐标 (0-7) , y 坐标 (0-7) , 1: 亮; 0: 灭.

// (0,7)------(7,7)

// |           |

// |           |

// |           |

// (0,0)------(7,0)

```

```

void matrix_clear()//点阵屏清屏, 参数无

```

```

void matrix_update_buf(uint8 *from) //点阵屏显示缓存更新数据, 参数:字模数组指针

```

```

void matrix_display_char(char c) // 描述: 点阵屏显示字符, 参数: 字符

```

#### 示例代码 1

```

#include <STC8HX.h>

uint32 sys_clk = 24000000;

//系统时钟确认

#include "lib/matrix.h"//调用 点阵 头文件

#include "lib/hc595.h"

#include "lib/rgb.h"

#include "lib/delay.h"

#include "lib/led8.h"

```

```
void twen_board_init()
{
    hc595_init();//HC595 初始化

    hc595_disable();//HC595 禁止点阵和数码管输出

    rgb_init();//RGB 初始化

    delay(10);

    rgb_show(0,0,0,0);//关闭 RGB

    delay(10);
}

void setup()
{
    matrix_init();//点阵初始化

    twen_board_init();

    led8_disable();//关闭 8 个 LED 流水灯电源
}

void loop()
{
    matrix_scan_callback();//点阵扫描回调函数

    matrix_set_pixel(0,0,1);//点阵显示一个点
```

```
}  
  
void main(void)  
{  
  
    setup();  
  
    while(1){  
  
        loop();  
  
    }  
  
}
```

#### 示例代码 2

```
#include <STC8HX.h>  
  
uint32 sys_clk = 24000000;  
  
//系统时钟确认  
  
#include "lib/matrix.h"//引用 点阵 头文件  
  
#include "lib/hc595.h"  
  
#include "lib/rgb.h"  
  
#include "lib/delay.h"  
  
#include "lib/led8.h"  
  
#include "lib/oled.h"  
  
void twen_board_init()  
{  
  
    hc595_init();
```



```
hc595_disable();

rgb_init();

delay(100);

rgb_show(0,0,0,0);//熄灭 RGB

delay(100);
}

void Timer0Init(void) //100 微秒@24.000MHz
{

    TMOD |= 0x00; //模式 0

    TL0 = 0x37; //设定定时初值

    TH0 = 0xff; //设定定时初值
}

void T_IRQ0(void) interrupt 1 using 1{

    matrix_scan_callback();//点阵扫描回调函数
}

void setup()
{

    matrix_init();//点阵初始化

    twen_board_init();
```

```

led8_disable();//关闭 8 个 LED 流水灯电源

Timer0Init();

EA = 1; // 控制总中断

ET0 = 1; // 控制定时器中断

TR0 = 1;// 启动定时器
}

void loop()
{

matrix_display_num(123);//点阵显示数字 123

}

void main(void)
{

setup();

while(1){

loop();

}

}

```

示例代码 3

```

#include <STC8HX.h>

uint32 sys_clk = 24000000;

//系统时钟确认

```

```
#include "lib/matrix.h"//调用 点阵 头文件

#include "lib/hc595.h"

#include "lib/rgb.h"

#include "lib/delay.h"

#include "lib/led8.h"

#include "lib/oled.h"

void twen_board_init()

{

    hc595_init();

    hc595_disable();

    rgb_init();

    delay(100);

    rgb_show(0,0,0,0);//熄灭 RGB

    delay(100);

}

void Timer0Init(void) //100 微秒@24.000MHz

{

    TMOD |= 0x00; //模式 0

    TL0 = 0x37; //设定定时初值

    TH0 = 0xff; //设定定时初值
```

```
}

void T_IRQ0(void) interrupt 1 using 1{

    matrix_scan_callback();//点阵扫描回调函数

}

void setup()

{

    matrix_init();//点阵初始化

    twen_board_init();

    led8_disable();//关闭 8 个 LED 流水灯电源

    Timer0Init();

    EA = 1; // 控制总中断

    ET0 = 1; // 控制定时器中断

    TR0 = 1;// 启动定时器

}

void loop()

{

    matrix_display_string("abcd");//点阵显示字符 abcd

}
```

```
void main(void)
{
    setup();
    while(1){
        loop();
    }
}
```

#### 示例代码 4

```
#include <STC8HX.h>

uint32 sys_clk = 24000000;

//系统时钟确认

#include "lib/matrix.h"//调用 点阵 头文件

#include "lib/hc595.h"

#include "lib/rgb.h"

#include "lib/delay.h"

#include "lib/led8.h"

void twen_board_init()
{
    hc595_init();

    hc595_disable();

    rgb_init();

    delay(100);
```

```
rgb_show(0,0,0,0);//熄灭 RGB

delay(100);
}

void Timer0Init(void) //100 微秒@24.000MHz
{
    TMOD |= 0x00; //模式 0

    TL0 = 0x37; //设定定时初值

    TH0 = 0xff; //设定定时初值
}

void T_IRQ0(void) interrupt 1 using 1{

    matrix_scan_callback();//点阵扫描回调函数
}

uint8_t matrix[8]; //自定义数组缓存区

void setup()
{
    matrix_init();//点阵初始化

    twen_board_init();

    led8_disable();//关闭 8 个 LED 流水灯电源
```

```
Timer0Init();

EA = 1; // 控制总中断

ET0 = 1; // 控制定时器中断

TR0 = 1; // 启动定时器
}

void loop()
{
    matrix[0] = 0xe3;matrix[1] = 0xc1; // 缓存数据到数组
    matrix[2] = 0x81;matrix[3] = 0x03; // 缓存数据到数组
    matrix[4] = 0x03;matrix[5] = 0x81; // 缓存数据到数组
    matrix[6] = 0xc1;matrix[7] = 0xe3; // 缓存数据到数组
    matrix_update_buf(matrix); // 点阵更新显示缓存数据
    delay(300);

    matrix[0] = 0xff;matrix[1] = 0xe7; // 缓存数据到数组
    matrix[2] = 0xc3;matrix[3] = 0x87; // 缓存数据到数组
    matrix[4] = 0x87;matrix[5] = 0xc3; // 缓存数据到数组
    matrix[6] = 0xe7;matrix[7] = 0xff; // 缓存数据到数组
    matrix_update_buf(matrix); // 点阵更新显示缓存数据
    delay(300);
}
```

```
void main(void)
{
    setup();
    while(1){
        loop();
    }
}
```

#### 示例代码 5

```
#include <STC8HX.h>

uint32 sys_clk = 24000000;

//系统时钟确认

#include "lib/matrix.h"//调用 点阵 头文件

#include "lib/hc595.h"

#include "lib/rgb.h"

#include "lib/delay.h"

#include "lib/led8.h"

uint8 mylist[8]={ 0xe3,0xc1,0x81,0x03,0x03,0x81,0xc1,0xe3};//自定义数组

void twen_board_init()
{
    hc595_init();

    hc595_disable();
}
```



```
rgb_init();

delay(100);

rgb_show(0,0,0,0);//熄灭 RGB

delay(100);
}

void Timer0Init(void) //100 微秒@24.000MHz
{

    TMOD |= 0x00; //模式 0

    TL0 = 0x37; //设定定时初值

    TH0 = 0xff; //设定定时初值
}

void T_IRQ0(void) interrupt 1 using 1{

    matrix_scan_callback();//点阵扫描回调函数
}

void setup()

{

    matrix_init();//点阵初始化

    twen_board_init();

    led8_disable();//关闭 8 个 LED 流水灯电源
```

```
Timer0Init();

EA = 1; // 控制总中断

ET0 = 1; // 控制定时器中断

TR0 = 1; // 启动定时器
}

void loop()
{

matrix_update_buf(mylist); // 点阵更新显示数组缓存数据
}

void main(void)
{

setup();

while(1){

loop();

}

}
```