树莓派开发

15 用树莓派点亮数码管

萬和IT教育
因专业而精彩
我们·始于1993年
1993-2013
专注IT教育二十年

点亮单数码管

萬和 IT教育
因专业而精彩
我们·始于1993年

1993-2013
专注IT教育二十年

一、点亮单数码管

萬和 IT教育 因专业而精彩 我们·始于1993年

- 数码管的显示分为静态和动态两种。静态就是一个GPIO控制一个LED小灯管。
- 但是随着控制数码管数量的增加，GPIO口就占用太多了，所以多个数码管可以有共阴和共阳两种共享引脚。
- 这个时候如果采用静态点亮数码管的方式，共享引脚的数码管显示完全一样。
- 所谓动态方式，就是通过GPIO选择引脚，选择要点亮的某个数码管，然后通过共享引脚点亮LED小灯管。然后快速切换点亮其他数码管，由于点亮的切换频率非常快所以感觉上数码管一直亮着。
- 当然也可以使用辅助芯片，如MAX7129等，只需要用更加简化的方式控制显示数据。

20 1993-2013 专注IT教育二十年

二、单数码管显示时间

萬和 IT教育 因专业而精彩 我们·始于1993年

```
import RPi.GPIO as GPIO
import time, random
"""
Display date to LED lights
There are four lights, it displays 4 number
"""
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

def setp(n, status='on'):
    if status == 'on':
        GPIO.output(n, GPIO.HIGH)
    else:
        GPIO.output(n, GPIO.LOW)
```

1993-2013 专注IT教育二十年

三、代码

萬和 IT教育
因专业而精彩
我们 · 始于1993年

```
for i in pins + sels:
    GPIO.setup(i, GPIO.OUT)
    setp(i, 'off')
for i in sels:
    setp(i, 'on')
#
#     __2__
#    |     |    |  0 ->  011 1111 -> 0x3f
# 1 |     | 3  |  1 ->  010 0001 -> 0x21
#    |__7__|    |  2 ->  111 0110 -> 0x76
#    |     |    |  4 ->  ...
# 6 |     | 4  |       ...
#    |__5__|    |  9 ->  ...      -> 0x5f
#
```

1993-2013
专注IT教育二十年

三、代码

萬和 IT教育
因专业而精彩
我们 · 始于1993年

```
pins = [27, 17, 22, 10, 25, 24, 11] #GPIO ports
sels = [14, 15, 18, 23] #GPIO ports to select led, there are four led lights
nums = [0x3f, 0x21, 0x76, 0x5e, 0x4d, 0x5b, 0x7b, 0x0e, 0x7f, 0x5f]

def flush(sel, n):
    setp(sels[sel], 'off')
    n = nums[n]
    for i in sels:
        if i != sels[sel]:
            setp(i, 'on')
    for i in range(7):
        if (n & (1 << i)):
            setp(pins[i], 'on')
    for i in range(7):
        if (n & (1 << i)):
            setp(pins[i], ‘off‘)
```

1993-2013
专注IT教育二十年

三、代码

萬和 IT教育
因专业而精彩
我们 · 始于1993年

```
try:
    while True:
        t = time.gmtime()
        flush(3, t.tm_min / 10)
        flush(2, t.tm_min % 10)
        flush(1, t.tm_sec / 10)
        flush(0, t.tm_sec % 10)
except:
    GPIO.cleanup()
```

1993-2013
专注IT教育二十年

三、代码

萬和 IT教育 因专业而精彩 我们·始于1993年

- 其中灯管编号，引脚对应关系可以根据自己的连线方式自定义和修改：

```
#
#     __2__
#    |     |    |  0 ->  011 1111 -> 0x3f
# 1  |     | 3  |  1 ->  010 0001 -> 0x21
#    |__7__|    |  2 ->  111 0110 -> 0x76
#    |     |    |  4 ->  ...
# 6  |     | 4  |       ...
#    |__5__|    |  9 ->  ...      -> 0x5f
```

```
pins = [27, 17, 22, 10, 25, 24, 11] #GPIO ports
sels = [14, 15, 18, 23] #GPIO ports to select led, there are four led lights
nums = [0x3f, 0x21, 0x76, 0x5e, 0x4d, 0x5b, 0x7b, 0x0e, 0x7f, 0x5f]#0,1,2,3~9 对应编码
```

1993-2013 专注IT教育二十年

四、点亮四位共阴数码管

萬和 IT教育 因专业而精彩
我们 · 始于1993年

实物

一、模块描述

1、名称：4位数码管显示模块，采用74HC595+8550三极管驱动

2、数码管型号：0.56英寸共阳数码管

3、工作电压：3.3V-5V

4、设有两个固定螺栓孔方便安装，孔间距34MM

5、底板尺寸：5cm*2.7cm

1993-2013
专注IT教育二十年

四、点亮四位共阴数码管

萬和 IT教育 因专业而精彩
我们 · 始于1993年

实物背板
四针低电平I/O

二 模块接口说明：输入与输出对称，可拼接多块单元板。

1、两针电源接口，+5V可外接3.3V-5V电压（可以直接与5v单片机和3.3v单片机相连），GND为电源负极。

2、四针位选择1-4输入，低电平位选，可直接与单片机IO口相连。

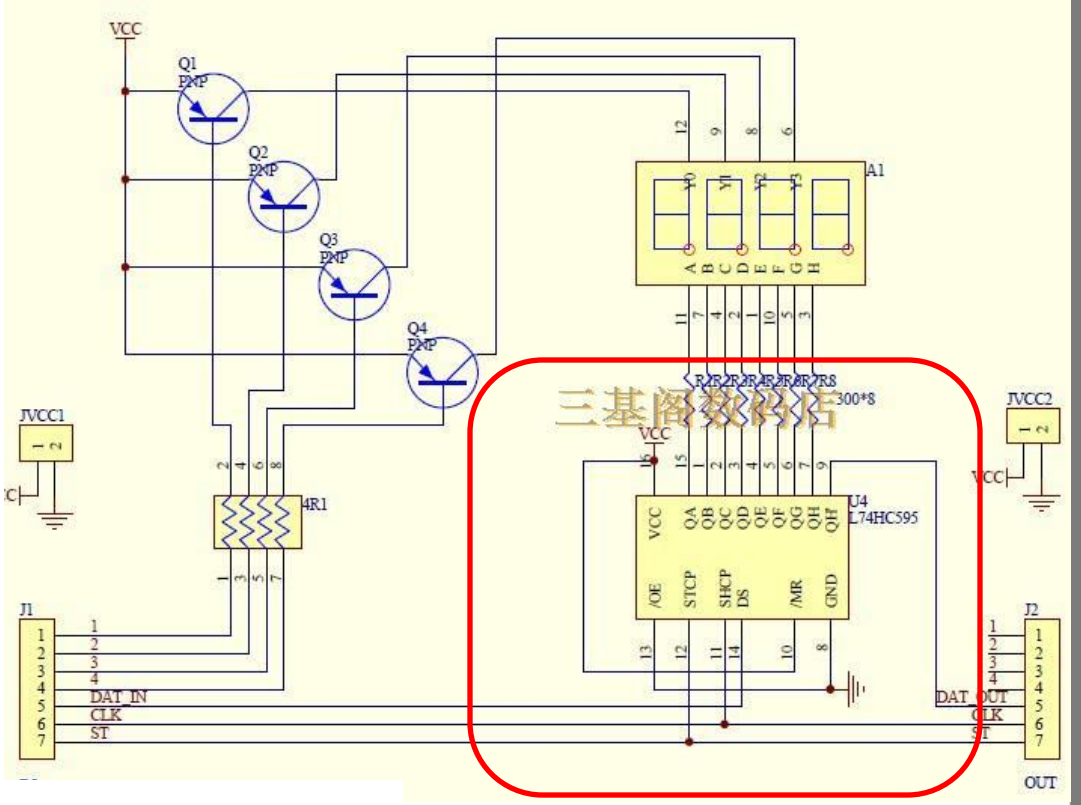
3、三针串入数据引脚，DAT,CLK,ST，为74HC595串行输入控制，可直接与单片机IO口相连。

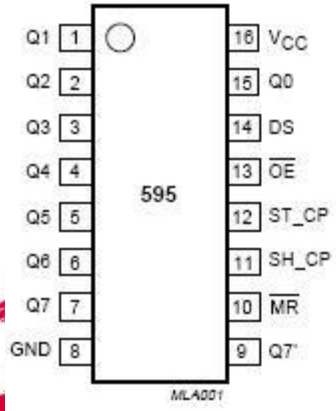
注：模块与单片机IO口相连需要7个IO口，不管你拼接多少块单元，都只要7个IO口连接，扩展灵活

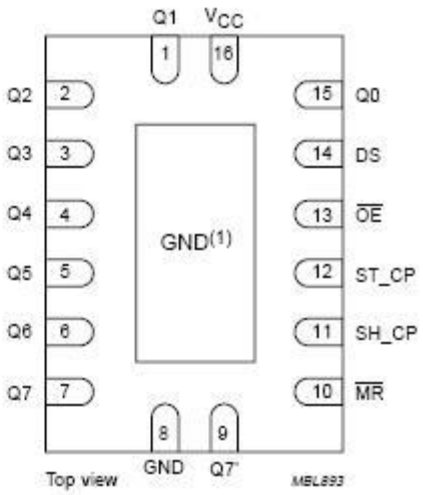
1993-2013
专注IT教育二十年

74HC595的控制逻辑

- 74HC595是8位串行输入并行输出移位寄存器，也就是串行转并行，来驱动数码管，否则，就需要如前例，每个数码管的每笔驱动。如右图：
- 各脚定义如下：





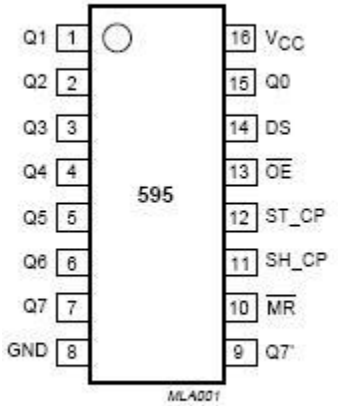


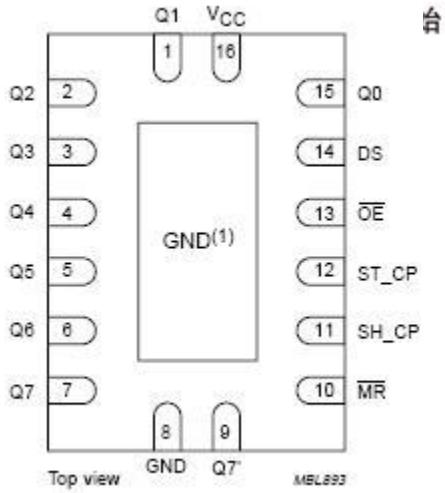
- 我们用到的输入分别接74HC595
- DAT_IN：DS
- CLK：SHCP
- ST：STCP

专注IT教育二十年

74HC595的控制逻辑

萬和 IT教育
因专业而精彩
台于1993年

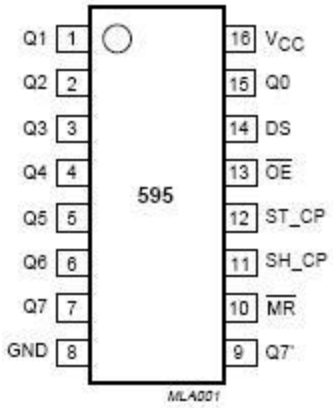


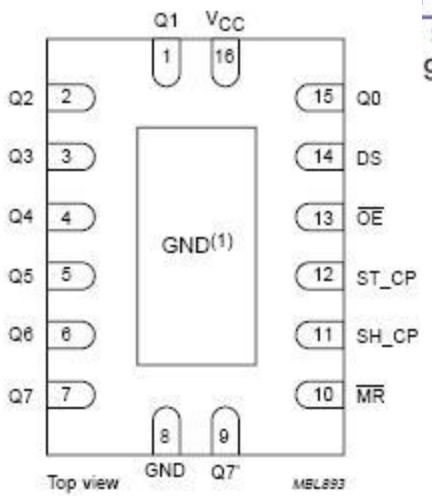


符号	引脚	描述
Q0…Q7	15， 1， 7	并行数据输出
GND	8	地
Q7’	9	串行数据输出
MR	10	主复位（低电平）
SH_{CP}	11	移位寄存器时钟输入
ST_{CP}	12	存储寄存器时钟输入
OE	13	输出有效（低电平）
D_S	14	串行数据输入
V_{CC}	16	电源

1993-2013
专注IT教育二十年

74HC595的控制逻辑





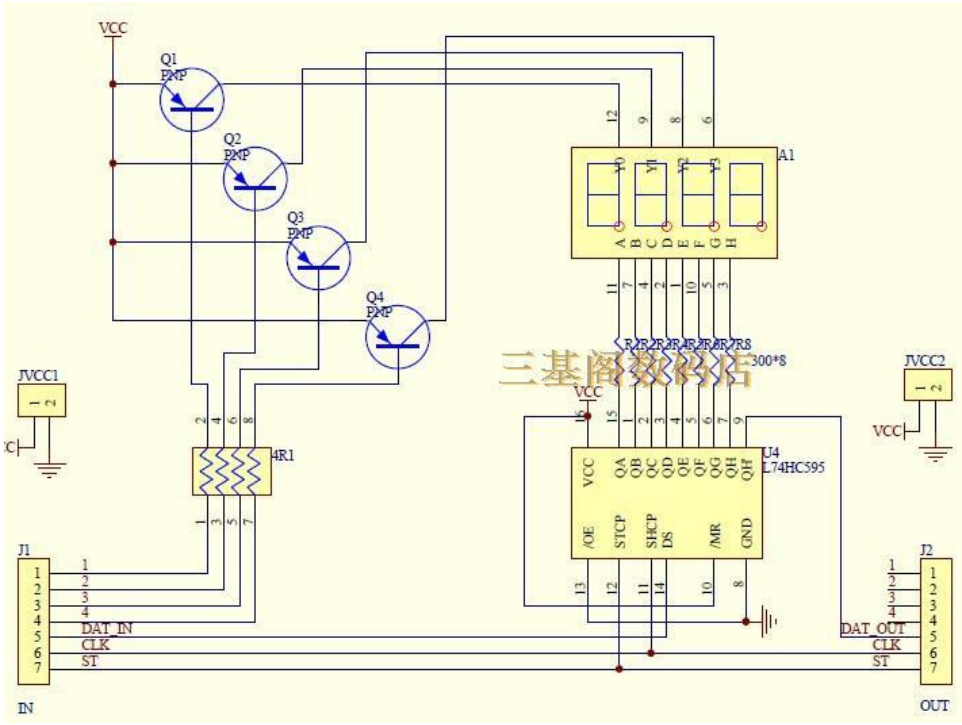
- DS（14）为串行数据输入口；
- SH_CP（11）为串行时钟输入口；
- ST_CP（12）为寄存器移位

- SH_CP每个上升沿到来时，芯片内部的移位寄存器会左移一位，最低位由DS决定，最高位移出丢失，次高位成为最高位，并在Q7'体现出来（根据Q7'可以看出，74HC595也有串行输出功能）；
- ST_CP每个上升沿会将移位寄存器的值输出到存储寄存器,存储寄存器直接和引脚Q0~Q7相连，所以存储寄存器的值会直接反映在引脚Q0~Q7上，从而实现串行转并行功能；
- OE是输出使能，高电平时Q0~Q7为高阻态，低电平时Q0~Q7为存储寄存器的值；MR为低电平时，移位寄存器会被清0，高电平时无效；VCC接电源；GND接地。

IT教育 业而精彩 993年
1993-2013 专注IT教育二十年

数码管的片选与刷新

- 数码管驱动方式：动态驱动。
- 每个数码管的a,b,c,d,e,f,g,dp同名片段连在一起，同时每个数码管有自己的独立的控制I/O,用于控制是否显示。比如4位数码管，总共有4个针脚用于控制每个数码管的显示。
- 3461BS共阴数码管，那么控制片段显示的针脚为LOW的时候，对应的数码管才会显示。



- 由于74HC595每次只能显示一个数字，如果需要每个数码管显示不同的数字，那么必须通过在1～2ms内反复刷新。

1993-2013
专注IT教育二十年

代码说明

```
// 0-9 10个显示数字与小数点（DP）
byte seven_seg_digits[10] =
                        { B00000011,  // = 0
                          B10011111,  // = 1
                          B00100101,  // = 2
                          B00001101,  // = 3
                          B10011001,  // = 4
                          B01001001,  // = 5
                          B01000001,  // = 6
                          B00011111,  // = 7
                          B00000001,  // = 8
                          B00001001,   // = 9
                          B111111110    // = dp
                         };
int latchPin = 6;  //ST_CP移位
int clockPin = 5;  //SH_CP时钟
int dataPin = 4;  //DS数据串口
```

数码管实现0-9的真值表：

数字	Pin7	Pin6	Pin5	Pin4	Pin3	Pin2	Pin1	Pin0
0	HIGH	HIGH	LOW	LOW	LOW	LOW	LOW	LOW
1	HIGH	HIGH	HIGH	HIGH	HIGH	LOW	LOW	HIGH
2	HIGH	LOW	HIGH	LOW	LOW	HIGH	LOW	LOW
3	HIGH	LOW	HIGH	HIGH	LOW	LOW	LOW	LOW
4	HIGH	HIGH	LOW	HIGH	HIGH	LOW	LOW	HIGH
5	HIGH	LOW	LOW	HIGH	LOW	LOW	HIGH	LOW
6	HIGH	HIGH	LOW	LOW	LOW	LOW	HIGH	HIGH
7	HIGH	HIGH	HIGH	HIGH	HIGH	LOW	LOW	LOW
8	HIGH	LOW	LOW	LOW	LOW	LOW	LOW	LOW
9	HIGH	LOW	LOW	HIGH	HIGH	LOW	LOW	LOW
DP	LOW	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH	HIGH

1993-2013
专注IT教育二十年

代码说明

萬和 IT教育
因专业而精彩
我们 · 始于1993年

```
void setup() {
  pinMode(latchPin, OUTPUT);  //移位输出
  pinMode(clockPin, OUTPUT); //时钟输出
  pinMode(dataPin, OUTPUT); //数码管串口输出
  pinMode(8, OUTPUT); //片选1输出
  pinMode(9, OUTPUT); //片选2输出
  pinMode(10, OUTPUT); //片选3输出
  pinMode(11, OUTPUT); //片选4输出
  Serial.begin(9600); //树莓派串口传输速率
//  pinMode(2, INPUT);  //温度传感器采集口
}
```

1993-2013
专注IT教育二十年

代码说明

萬和 IT教育 因专业而精彩 我们·始于1993年

```
void sevenSegWrite(byte digit, int b) {

  digitalWrite(8, HIGH);  //全暗
  digitalWrite(9, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(11, HIGH);

  digitalWrite(latchPin, LOW); //移位低
  shiftOut(dataPin, clockPin, LSBFIRST, seven_seg_digits[digit]); //移位
  digitalWrite(latchPin, HIGH); //移位高，完成写入
  digitalWrite(b, LOW); //点亮b（片选）
}
```

- SH_CP每个上升沿到来时，芯片内部的移位寄存器会左移一位，最低位由DS决定，最高位移出丢失，次高位成为最高位，并在Q7‘体现出来（根据Q7'可以看出，74HC595也有串行输出功能）；
- ST_CP每个上升沿会将移位寄存器的值输出到存储寄存器,存储寄存器直接和引脚Q0~Q7相连，所以存储寄存器的值会直接反映在引脚Q0~Q7上，从而实现串行转并行功能；

1993-2013 专注IT教育二十年

代码说明

萬和 IT教育 因专业而精彩 我们·始于1993年

- shiftOut(dataPin,clockPin,bitOrder,val)
- shiftOut函数能够将数据通过串行的方式在引脚上输出，相当于一般意义上的同步串行通信，这是控制器与控制器、控制器与传感器之间常用的一种通信方式。
- shiftOut函数无返回值，有4个参数：dataPin、clockPin、bitOrder、val，具体说明如下：
- dataPin：数据输出引脚，数据的每一位将逐次输出。引脚模式需要设置成输出。
- clockPin：时钟输出引脚，为数据输出提供时钟，引脚模式需要设置成输出。
- bitOrder：数据位移顺序选择位，该参数为byte类型，有两种类型可选择，分别是高位先入MSBFIRST和低位先入LSBFIRST。
- val：所要输出的数据值。

1993-2013 专注IT教育二十年

我们的代码：

```
digitalWrite(latchPin, LOW); //移位低
 shiftOut(dataPin, clockPin, LSBFIRST, seven_seg_digits[digit]);
 digitalWrite(latchPin, HIGH); //移位高，完成写入
 digitalWrite(b, LOW); //点亮b（片选）
```

代码说明

萬和 IT教育
因专业而精彩
我们 · 始于1993年

- shiftOut函数原型如下：

```
void shiftOut(uint8_t dataPin, uint8_t clockPin, uint8_t bitOrder, uint8_t
val)
{
    uint8_t i;
    for (i = 0; i < 8; i++)
    {
        if (bitOrder == LSBFIRST)
            digitalWrite(dataPin, !!(val & (1 << i)));
        else
            digitalWrite(dataPin, !!(val & (1 << (7 - i))));
        digitalWrite(clockPin, HIGH);
        digitalWrite(clockPin, LOW);
    }
}
```

1993-2013
专注IT教育二十年

代码说明

萬和 IT教育
因专业而精彩
我们 · 始于1993年

```
void showNum(int num) {
  int b = 11; //数码管移位操作次数-实际以数据位数决定
  while(1) {
    sevenSegWrite(num % 10, b); //将num逐位写到b位数码管
    b -= 1;  //数码管移一位
    if(num/10 > 0){ //数值移一位
      num /= 10;
    }
    else {
      break;
    }
  }
}

 void loop() {
  showNum(analogRead(A5)); //显示树莓派A5口的采集信息，反复写（刷新）
}
```

1993-2013
专注IT教育二十年