LEDMatrix

Overview

In this project, we will use two 74HC595 to control a monochrome LEDMatrix (8*8) to make it display some graphics and characters.

Experimental Materials:

Raspberry Pi *1

T-type expansion board *1

220 ohm resistor *8

LEDMatrix *1

Breadboard*1

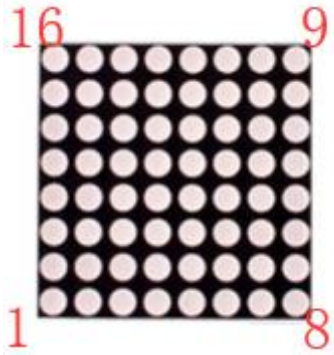
74HC595 *2

Some DuPont lines

Product description：

LED matrix is a rectangular display module that consists of several LEDs.

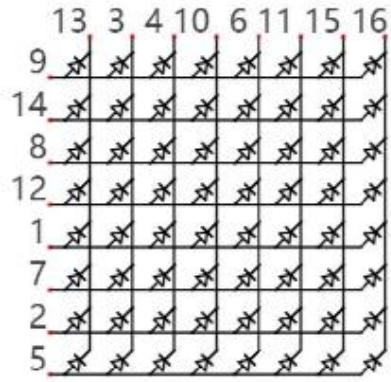
The following is an 8*8 monochrome LED matrix with 64 LEDs (8 rows and 8 columns).



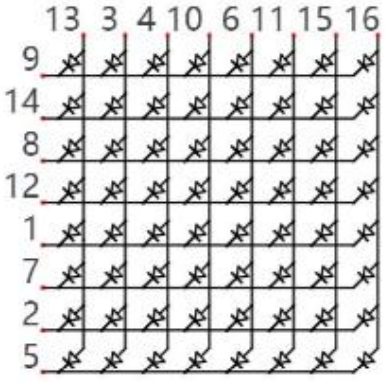
In order to facilitate the operation and save the ports, positive pole of LEDs in each row and negative pole of LEDs in each column are respectively connected together inside LED matrix module, which is called Common Anode. There is another form. Negative pole of LEDs in each row and positive pole of LEDs in each column are respectively connected together, which is called Common Cathode.

The one we use in this project is a common anode LEDMatrix.

Connection mode of common anode

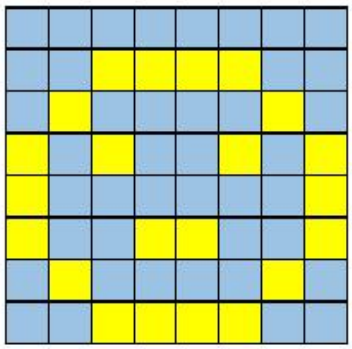


Connection mode of common cathode



Let us learn how connection mode of common anode works. Choose 16

ports on RPI board to connect to the 16 ports of LED Matrix. Configured one port in columns for low level, which make the column of the port selected. Then configure the eight ports in row to display content in the selected column. Delay for a moment. And then select the next column and outputs the corresponding content. This kind of operation to column is called scan. If you want to display the following image of a smiling face, you can display it in 8 columns, and each column is represented by one byte.



1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	0	1	0
1	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1
1	0	0	1	1	0	0	1
0	1	0	0	0	0	1	0
0	0	1	1	1	1	0	0

Column	Binary	Hexadecimal
1	0001 1100	0x1c
2	0010 0010	0x22
3	0101 0001	0x51
4	0100 0101	0x45
5	0100 0101	0x45
6	0101 0001	0x51
7	0010 0010	0x22
8	0001 1100	0x1c

First, display the first column, then turn off the first column and display the second column. .... turn off the seventh column and display the 8th

column, and then start from the first column again like the control of Graph LEDBar. The whole progress will be repeated rapidly and circularly. Due to afterglow effect of LED and visual residual effect of human eyes, we will see a picture of a smiling face directly rather than LED are turned on one column by one column (although in fact it is the real situation).

Scanning rows is another display way of dot matrix. Whether scanning line or column, 16 GPIO are required. In order to save GPIO of control board, two 74HC595 is used. Every piece of 74HC595 has eight parallel output ports, so two pieces has 16 ports in total, just enough. The control line and data line of two 74HC595 are not all connected to the RPi, but connect Q7 pin of first stage 74HC595 to data pin of second one, namely, two 74HC595 are connected in series. It is the same to using one "74HC595" with 16 parallel output ports.

Technical Parameters:

Product Name : LED Lattice Tube Matrix Module

LED Display : Red

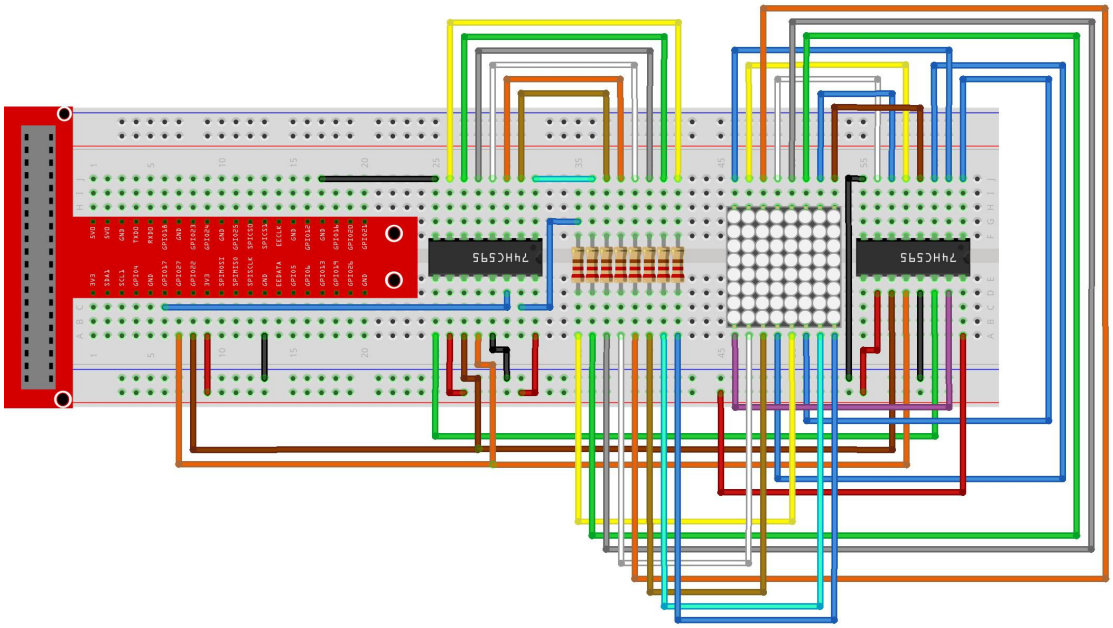
Dot Numbers : 8*8

Pin Numbers : 16 Pins

Material : Plastic

Color : Black, White

Wiring diagram：



C code：

```
#include <wiringPi.h>
#include <stdio.h>
#include <wiringShift.h>

#define   dataPin   0   //DS Pin of 74HC595(Pin14)
#define   latchPin  2   //ST_CP Pin of 74HC595(Pin12)
#define   clockPin 3    //SH_CP Pin of 74HC595(Pin11)
// data of smiling face
unsigned char pic[]={0x1c,0x22,0x51,0x45,0x45,0x51,0x22,0x1c};
unsigned char data[]={  // data of "0-F"
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // " "
    0x00, 0x00, 0x3E, 0x41, 0x41, 0x3E, 0x00, 0x00, // "0"
    0x00, 0x00, 0x21, 0x7F, 0x01, 0x00, 0x00, 0x00, // "1"
    0x00, 0x00, 0x23, 0x45, 0x49, 0x31, 0x00, 0x00, // "2"
    0x00, 0x00, 0x22, 0x49, 0x49, 0x36, 0x00, 0x00, // "3"
    0x00, 0x00, 0x0E, 0x32, 0x7F, 0x02, 0x00, 0x00, // "4"
```

```
    0x00, 0x00, 0x79, 0x49, 0x49, 0x46, 0x00, 0x00, // "5"
    0x00, 0x00, 0x3E, 0x49, 0x49, 0x26, 0x00, 0x00, // "6"
    0x00, 0x00, 0x60, 0x47, 0x48, 0x70, 0x00, 0x00, // "7"
    0x00, 0x00, 0x36, 0x49, 0x49, 0x36, 0x00, 0x00, // "8"
    0x00, 0x00, 0x32, 0x49, 0x49, 0x3E, 0x00, 0x00, // "9"
    0x00, 0x00, 0x3F, 0x44, 0x44, 0x3F, 0x00, 0x00, // "A"
    0x00, 0x00, 0x7F, 0x49, 0x49, 0x36, 0x00, 0x00, // "B"
    0x00, 0x00, 0x3E, 0x41, 0x41, 0x22, 0x00, 0x00, // "C"
    0x00, 0x00, 0x7F, 0x41, 0x41, 0x3E, 0x00, 0x00, // "D"
    0x00, 0x00, 0x7F, 0x49, 0x49, 0x41, 0x00, 0x00, // "E"
    0x00, 0x00, 0x7F, 0x48, 0x48, 0x40, 0x00, 0x00, // "F"
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // " "
};
void _shiftOut(int dPin,int cPin,int order,int val){
    int i;
    for(i = 0; i < 8; i++){
        digitalWrite(cPin,LOW);
        if(order == LSBFIRST){
            digitalWrite(dPin,((0x01&(val>>i)) == 0x01) ? HIGH : LOW);
            delayMicroseconds(10);
        }
        else {//if(order == MSBFIRST){
            digitalWrite(dPin,((0x80&(val<<i)) == 0x80) ? HIGH : LOW);
            delayMicroseconds(10);
        }
        digitalWrite(cPin,HIGH);
        delayMicroseconds(10);
    }
}
int main(void)
{
    int i,j,k;
    unsigned char x;
    if(wiringPiSetup() == -1){ //when initialize wiring failed,print messageto
screen
        printf("setup wiringPi failed !");
        return 1;
    }
    pinMode(dataPin,OUTPUT);
    pinMode(latchPin,OUTPUT);
    pinMode(clockPin,OUTPUT);
    while(1){
        for(j=0;j<500;j++){// Repeat enough times to display the smiling face
a period of time
```

```
            x=0x80;
            for(i=0;i<8;i++){
                digitalWrite(latchPin,LOW);
                _shiftOut(dataPin,clockPin,MSBFIRST,pic[i]);// first shift
data of line information to the first stage 74HC959
                _shiftOut(dataPin,clockPin,MSBFIRST,~x);//then shift data of
column information to the second stage 74HC959

                digitalWrite(latchPin,HIGH);//Output data of two stage 74HC595
at the same time
                x>>=1;// display the next column
                delay(1);
            }
        }
        for(k=0;k<sizeof(data)-8;k++){  //sizeof(data) total number of "0-F"
columns
            for(j=0;j<20;j++){// times of repeated displaying LEDMatrix in
every frame, the bigger the “j”, the longer the display time
               x=0x80;     // Set the column information to start from the first
column
                for(i=k;i<8+k;i++){
                    digitalWrite(latchPin,LOW);
                    _shiftOut(dataPin,clockPin,MSBFIRST,data[i]);
                    _shiftOut(dataPin,clockPin,MSBFIRST,~x);
                    digitalWrite(latchPin,HIGH);
                    x>>=1;
                    delay(1);
                }
            }
        }
    }
    return 0;
}
```

Python code：

```
#!/usr/bin/env python3
import RPi.GPIO as GPIO
import time
```

```
LSBFIRST = 1
MSBFIRST = 2
#define the pins connect to 74HC595
dataPin   = 11      #DS Pin of 74HC595(Pin14)
latchPin  = 13      #ST_CP Pin of 74HC595(Pin12)
clockPin = 15       #SH_CP Pin of 74HC595(Pin11)
pic = [0x1c,0x22,0x51,0x45,0x45,0x51,0x22,0x1c]# data of smiling face
data = [#data of "0-F"
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, # " "
    0x00, 0x00, 0x3E, 0x41, 0x41, 0x3E, 0x00, 0x00, # "0"
    0x00, 0x00, 0x21, 0x7F, 0x01, 0x00, 0x00, 0x00, # "1"
    0x00, 0x00, 0x23, 0x45, 0x49, 0x31, 0x00, 0x00, # "2"
    0x00, 0x00, 0x22, 0x49, 0x49, 0x36, 0x00, 0x00, # "3"
    0x00, 0x00, 0x0E, 0x32, 0x7F, 0x02, 0x00, 0x00, # "4"
    0x00, 0x00, 0x79, 0x49, 0x49, 0x46, 0x00, 0x00, # "5"
    0x00, 0x00, 0x3E, 0x49, 0x49, 0x26, 0x00, 0x00, # "6"
    0x00, 0x00, 0x60, 0x47, 0x48, 0x70, 0x00, 0x00, # "7"
    0x00, 0x00, 0x36, 0x49, 0x49, 0x36, 0x00, 0x00, # "8"
    0x00, 0x00, 0x32, 0x49, 0x49, 0x3E, 0x00, 0x00, # "9"
    0x00, 0x00, 0x3F, 0x44, 0x44, 0x3F, 0x00, 0x00, # "A"
    0x00, 0x00, 0x7F, 0x49, 0x49, 0x36, 0x00, 0x00, # "B"
    0x00, 0x00, 0x3E, 0x41, 0x41, 0x22, 0x00, 0x00, # "C"
    0x00, 0x00, 0x7F, 0x41, 0x41, 0x3E, 0x00, 0x00, # "D"
    0x00, 0x00, 0x7F, 0x49, 0x49, 0x41, 0x00, 0x00, # "E"
    0x00, 0x00, 0x7F, 0x48, 0x48, 0x40, 0x00, 0x00, # "F"
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, # " "
]
def setup():
    GPIO.setmode(GPIO.BOARD)    # Number GPIOs by its physical location
    GPIO.setup(dataPin, GPIO.OUT)
    GPIO.setup(latchPin, GPIO.OUT)
    GPIO.setup(clockPin, GPIO.OUT)

def shiftOut(dPin,cPin,order,val):
    for i in range(0,8):
        GPIO.output(cPin,GPIO.LOW);
        if(order == LSBFIRST):
            GPIO.output(dPin,(0x01&(val>>i)==0x01) and GPIO.HIGH or GPIO.LOW)
        elif(order == MSBFIRST):
            GPIO.output(dPin,(0x80&(val<<i)==0x80) and GPIO.HIGH or GPIO.LOW)
        GPIO.output(cPin,GPIO.HIGH);

def loop():
    while True:
```

```
        for j in range(0,500):# Repeat enough times to display the smiling face
a period of time
            x=0x80
            for i in range(0,8):
                GPIO.output(latchPin,GPIO.LOW)
                shiftOut(dataPin,clockPin,MSBFIRST,pic[i]) #first shift data
of line information to first stage 74HC959

                shiftOut(dataPin,clockPin,MSBFIRST,~x) #then shift data of
column information to second stage 74HC959
                GPIO.output(latchPin,GPIO.HIGH)# Output data of two stage
74HC595 at the same time
                time.sleep(0.001)# display the next column
                x>>=1
        for k in range(0,len(data)-8):#len(data) total number of "0-F" columns
            for j in range(0,20):# times of repeated displaying LEDMatrix in
every frame, the bigger the "j", the longer the display time.
                x=0x80      # Set the column information to start from the first
column
                for i in range(k,k+8):
                    GPIO.output(latchPin,GPIO.LOW)
                    shiftOut(dataPin,clockPin,MSBFIRST,data[i])
                    shiftOut(dataPin,clockPin,MSBFIRST,~x)
                    GPIO.output(latchPin,GPIO.HIGH)
                    time.sleep(0.001)
                    x>>=1
def destroy():   # When 'Ctrl+C' is pressed, the function is executed.
    GPIO.cleanup()
if __name__ == '__main__': # Program starting from here
    print ('Program is starting...' )
    setup()
    try:
        loop()
    except KeyboardInterrupt:
        destroy()
```

Experimental results:

In the directory where the code file is located, execute the following command

C：
gcc -Wall -o LEDMatrix LEDMatrix.c -lwiringPi
sudo ./LEDMatrix

Python:
python LEDMatrix.py

After the program is executed, LEDMatrix will display a smiling face, and then the display scrolling character "0-F", circularly.