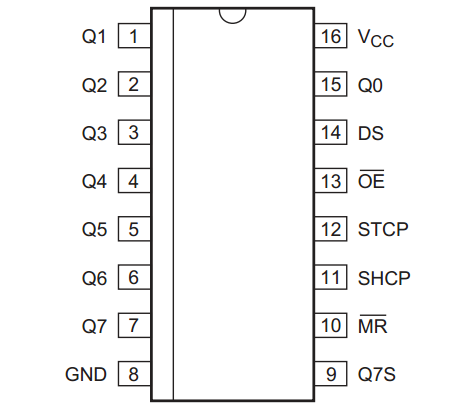
# SN74HC595 Drive 4-digit Digital Tube Experiment

**Introduction**

74HC595 is a CMOS shift register with 8-bit serial input and parallel open-drain output. The shift register provides data to a register with three-state output. The shift register and storage register have independent clocks respectively. The shift register of 74HC595 has the highest priority direct clear terminal (SRCLR) and the serial input terminal (DS) is used to cascade the serial output of the up one level. While the output enable terminal (OE) is high level voltage, the 74HC595 parallel output is in a high-impedance state. While it is low level voltage, it is used to enable parallel output.

Both the shift register clock SHCP and the storage register clock STCP are triggered by rising edges.

## Pin Definition

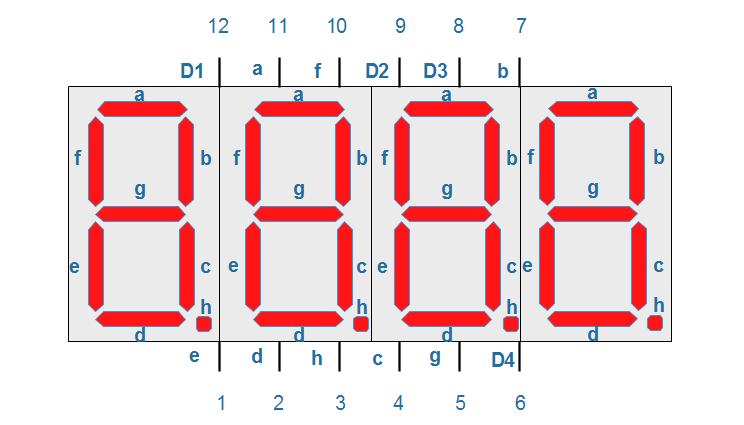


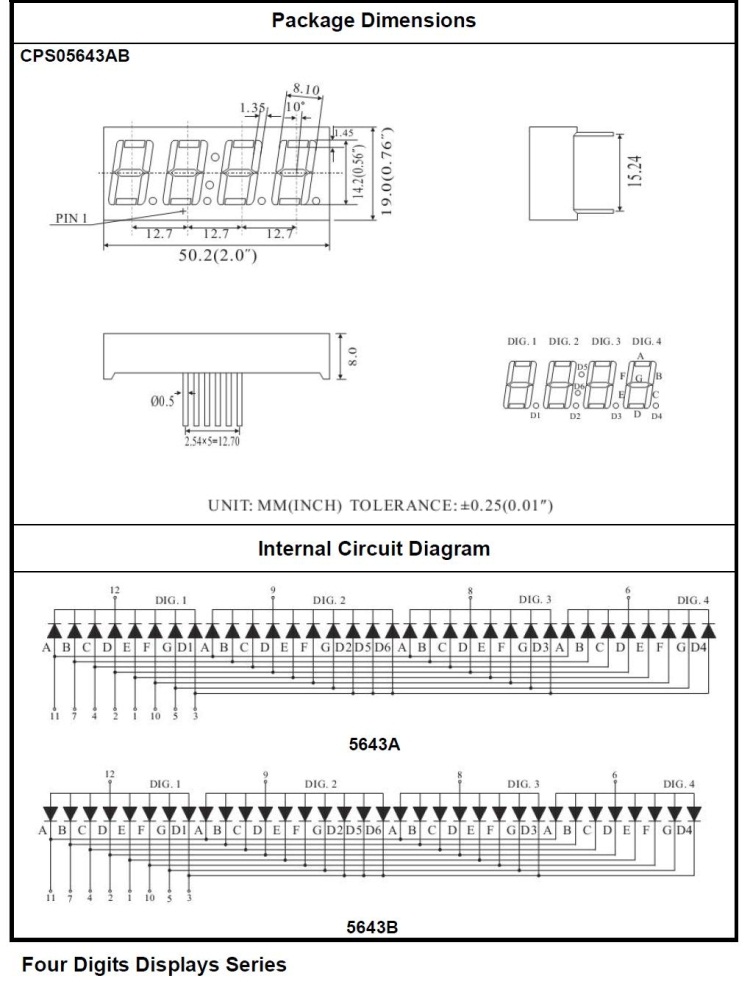
## Pin Description

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Pin** | **Description** |
| Q0 | 15 | parallel data output 0 |
| Q1 | 1 | parallel data output 1 |
| Q2 | 2 | parallel data output 2 |
| Q3 | 3 | parallel data output 3 |
| Q4 | 4 | parallel data output 4 |
| Q5 | 5 | parallel data output 5 |
| Q6 | 6 | parallel data output 6 |
| Q7 | 7 | parallel data output 7 |
| GND | 8 | ground(0V) |
| Q7S | 9 | serial data output |
| MR | 10 | master reset(active low) |
| SHCP | 11 | shift register clock input |
| STCP | 12 | storage register clock input |
| OE | 13 | output enable input(active low) |
| DS | 14 | serial data input |
| Vcc | 16 | supply voltage |

## 4-digit 7-segment Digital Tube Introduction

We have used 7-segment digital tubes before, so we know that when we want to display multiple numbers, we need a multi-digit digital tube. In fact, each 7-segment digital tube is almost the same as the tube used before. In this experiment, we will use Raspberry Pi as the main board to drive 4-digit 7-segment common anode digital tubes.





This digital tube has a total of 12 pins and the pin number in the upper left corner is 12. Eight of its pins are used to light up the “abcdefgh”segment tube and the remaining four pins D1, D2, D3 and D4 are used to represent “bit” pins. When the corresponding “bit” pin of these four digital tubes is high level voltage, it will be lit. The display principle of these four digital tubes is to continuously scan the D1, D2, D3 and D4 pins, and then light the corresponding eight-segment tubes in sequence. Since the speed of lighting up the digital tubes is so fast that humans eyes can not see it, so it looks like four digital tubes are displayed at the same time.

Based on the principles introduced above, we will now make a simulated countdown time bomb similar to the movie. This bomb will explode in a minute.

## Experimental Purpose

In this experiment, we will use the Raspberry Pi serial drive to output to 74HC595 and then drive a common anode four-digit digital tube by 74HC595 parallel port. Raspberry Pi then drives the bit selection pin of the digital tube directly. This experiment uses 8\* 220Ω resistors to limit the current so that the digital tube dynamically display number from 0 and 9.

## Component List

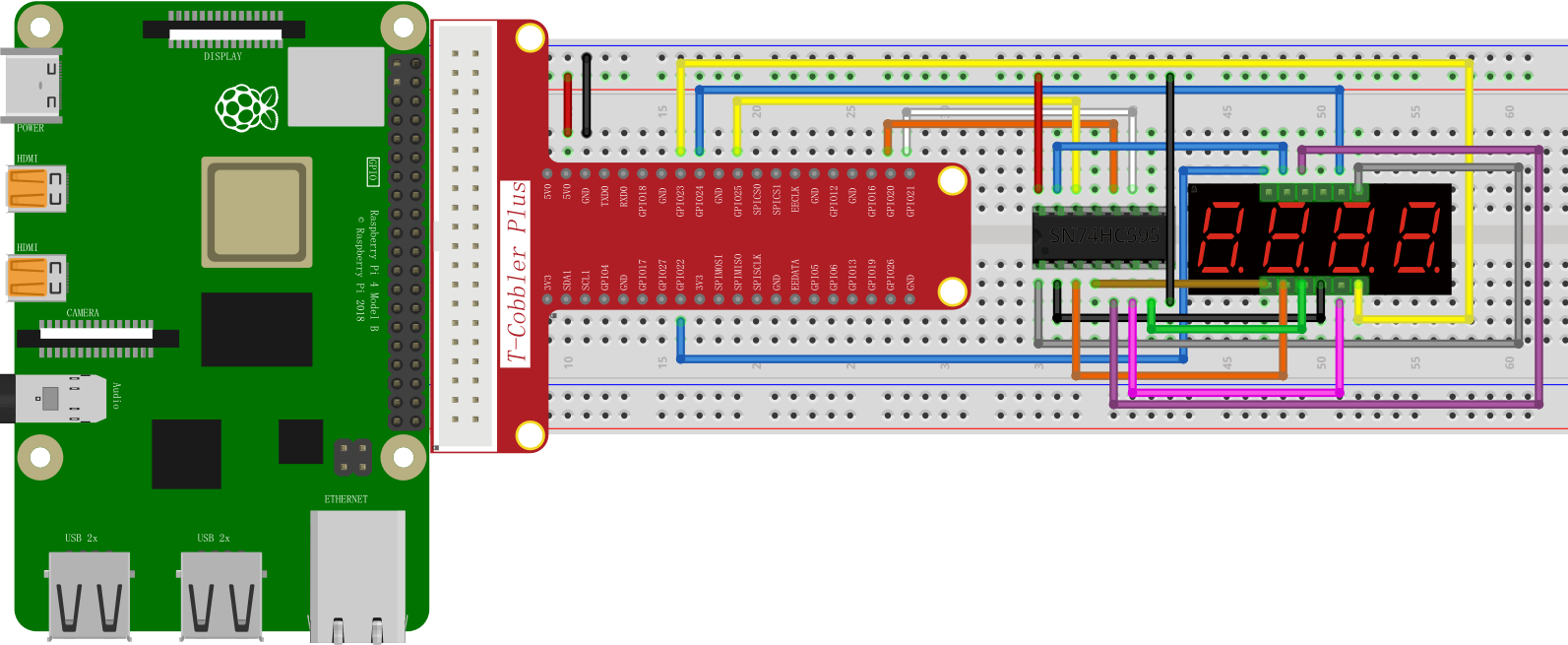
* Raspberry Pi main board \*1
* T-Cobbler Plus expansion board
* Four-digit common anode digital tube \*1
* SN74HC595 \*1
* Resistor 220Ω \*8
* Several wires

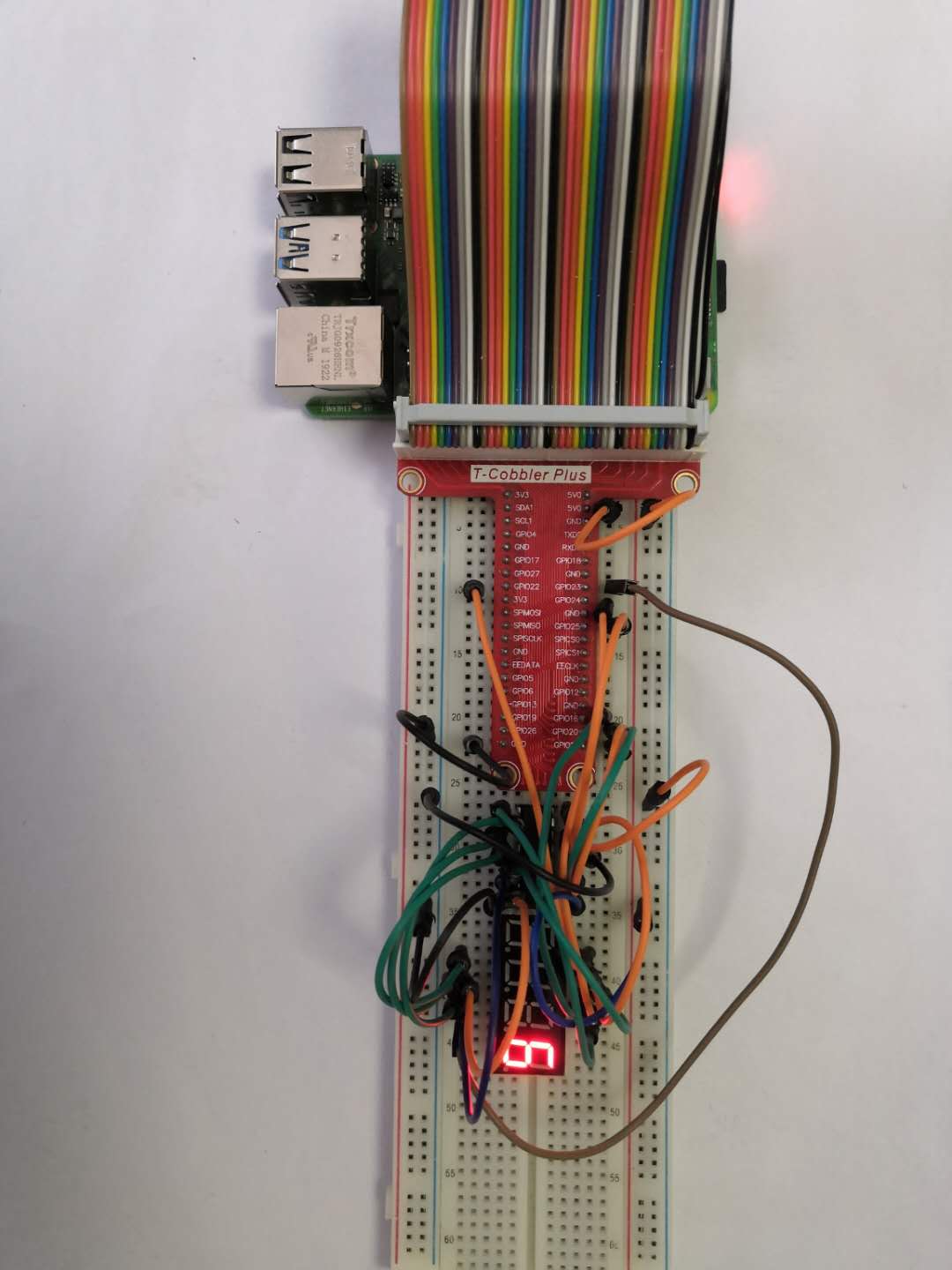
## Experimental connection schematic diagram

|  |  |
| --- | --- |
| Raspberry Pi | SN74HC595 |
| IO28(wiringPi)/20(BCM) | 12 STCP |
| IO29(wiringPi)/21(BCM) | 11 SHCP |
| IO6(wiringPi)/25(BCM) | 14 DS |

|  |  |
| --- | --- |
| Raspberry Pi | 7 Segment nixie tube |
| IO3(wiringPi)/22(BCM) | 12 |
| IO27(wiringPi)/2(BCM) | 9 |
| IO5(wiringPi)/24(BCM) | 8 |
| IO4(wiringPi)/23(BCM) | 6 |

|  |  |
| --- | --- |
| SN74HC595 | 7 Segment nixie tube |
| 15 | 11 |
| 1 | 7 |
| 2 | 4 |
| 3 | 2 |
| 4 | 1 |
| 5 | 10 |
| 6 | 5 |
| 7 | 3 |





C++ program

|  |
| --- |
| #include <wiringPi.h>  #include <stdio.h>  #define BIT\_CHOICE\_1 3  #define BIT\_CHOICE\_2 27  #define BIT\_CHOICE\_3 5  #define BIT\_CHOICE\_4 4  #define STCP\_PIN 28  #define SHCP\_PIN 29  #define DATA\_PIN 6 // Define the stcp shcp ds pin  #define uint8\_t unsigned char  int BIT\_CHOICE**[**4**]=** **{**BIT\_CHOICE\_1 **,** BIT\_CHOICE\_2 **,** BIT\_CHOICE\_3**,** BIT\_CHOICE\_4**};**// 4x8bit  unsigned char DisplayNumble**[**10**]=**  **{**0xC0**,** 0xF9**,** 0xA4**,** 0xB0**,** 0x99**,** 0x92**,** 0x82**,** 0xF8**,** 0x80**,** 0x90**};**  void shiftOut**(**uint8\_t dataPin**,** uint8\_t clockPin**,** uint8\_t val**)**  **{**  uint8\_t i**;**  **for** **(**i **=** 0**;** i **<** 8**;** i**++)**  **{**  digitalWrite**(**dataPin**,** **!!(**val **&** **(**1 **<<** **(**7**-**i**))));**  printf**(**"%d"**,!!(**val **&** **(**1 **<<** i**)));**  digitalWrite**(**clockPin**,** HIGH**);**  digitalWrite**(**clockPin**,** LOW**);**  **}**  printf**(**"\n"**);**  **}**  int main**()**  **{**  wiringPiSetup**();**  pinMode**(**STCP\_PIN**,** OUTPUT**);**  pinMode**(**SHCP\_PIN**,** OUTPUT**);**  pinMode**(**DATA\_PIN**,** OUTPUT**);** // Set stcp shcp ds pin to output mode  **for(**int i**=**0**;**i**<**4**;**i**++)**  **{**  pinMode**(**BIT\_CHOICE**[**i**],** OUTPUT**);**  digitalWrite**(**BIT\_CHOICE**[**i**],** LOW**);**  **}**  digitalWrite**(**SHCP\_PIN**,** HIGH**);**  **while(**1**)**  **{**  **for(**int i**=**0**;**i**<**10**;**i**++)**  **{**  digitalWrite**(**STCP\_PIN**,** LOW**);**  shiftOut**(**DATA\_PIN**,**SHCP\_PIN**,**DisplayNumble**[**i**]);** //serial shift out put display numble  digitalWrite**(**STCP\_PIN**,** HIGH**);**  digitalWrite**(**BIT\_CHOICE**[**3**],** HIGH**);**  delay**(**1000**);**  **}**  **}**  **}** |

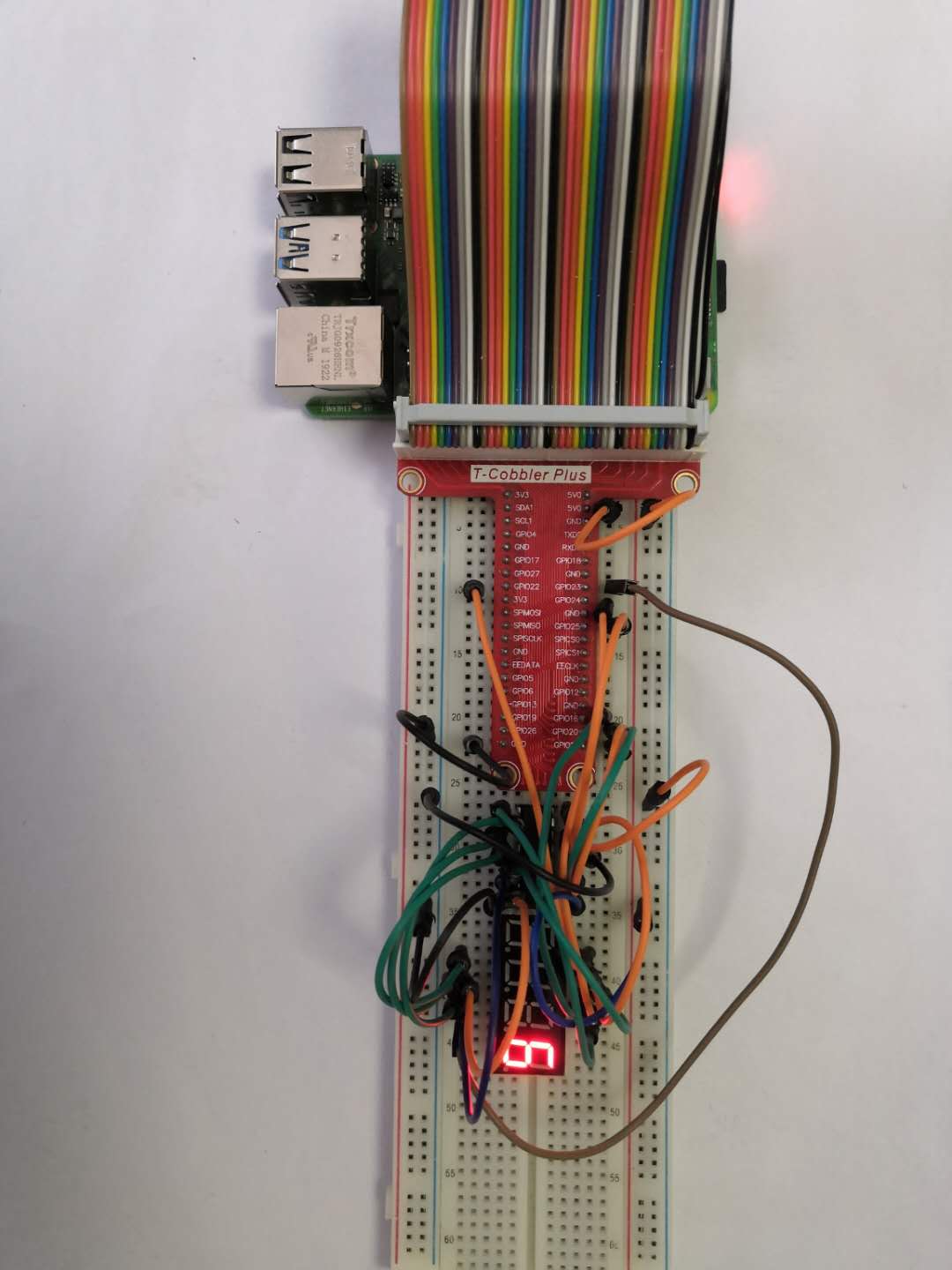
Python program

|  |
| --- |
| import RPi**.**GPIO as GPIO  import time  BIT\_CHOICE\_1 **=** 22  BIT\_CHOICE\_2 **=** 2  BIT\_CHOICE\_3 **=** 24  BIT\_CHOICE\_4 **=** 23  STCP\_PIN **=** 20  SHCP\_PIN **=** 21  DATA\_PIN **=** 25 # Define the stcp shcp ds pin  BIT\_CHOICE **=** **[**22 **,** 2 **,** 24**,** 23**]**  DisplayNumble**=[**0xC0**,** 0xF9**,** 0xA4**,** 0xB0**,** 0x99**,** 0x92**,** 0x82**,** 0xF8**,** 0x80**,** 0x90**]**  def setup**():**  GPIO**.**setmode**(**GPIO**.**BCM**)**  GPIO**.**setup**(**STCP\_PIN**,** GPIO**.**OUT**)**  GPIO**.**setup**(**SHCP\_PIN**,** GPIO**.**OUT**)**  GPIO**.**setup**(**DATA\_PIN**,** GPIO**.**OUT**)**  GPIO**.**output**(**SHCP\_PIN**,** GPIO**.**HIGH**)**  **for** pin in BIT\_CHOICE**:**  GPIO**.**setup**(**pin**,** GPIO**.**OUT**)** #set all pins' mode is output  GPIO**.**output**(**pin**,** GPIO**.**LOW**)**  def shiftOut**(**val**):**  **for** dat in range**(**8**):**  print**((**val **&** **(**1**<<(**7**-**dat**))** **))**  **if** **(**val **&** **(**1**<<(**7**-**dat**))):**  GPIO**.**output**(**DATA\_PIN**,** GPIO**.**HIGH**)**  **else:**  GPIO**.**output**(**DATA\_PIN**,** GPIO**.**LOW**)**  GPIO**.**output**(**SHCP\_PIN**,** GPIO**.**HIGH**)**  GPIO**.**output**(**SHCP\_PIN**,** GPIO**.**LOW**)**  print**(**"\n"**)**  **while** True**:**  setup**()**  **for** i in range**(**10**):**  GPIO**.**output**(**STCP\_PIN**,** GPIO**.**LOW**)**  shiftOut**(**DisplayNumble**[**i**])**  GPIO**.**output**(**STCP\_PIN**,** GPIO**.**HIGH**)**  GPIO**.**output**(**BIT\_CHOICE**[**3**],** GPIO**.**HIGH**)**  time**.**sleep**(**1**)** |

Java program

|  |
| --- |
|  |

Experimental Effect



The fourth digital tube loop displays the Numbers 0-9.