

1bit digital tube display

Overview

This course will use 8-segment digital tube to display the numbers 0-9

Materials:

Raspberry Pi *1

T-type expansion board *1

Color cable *1

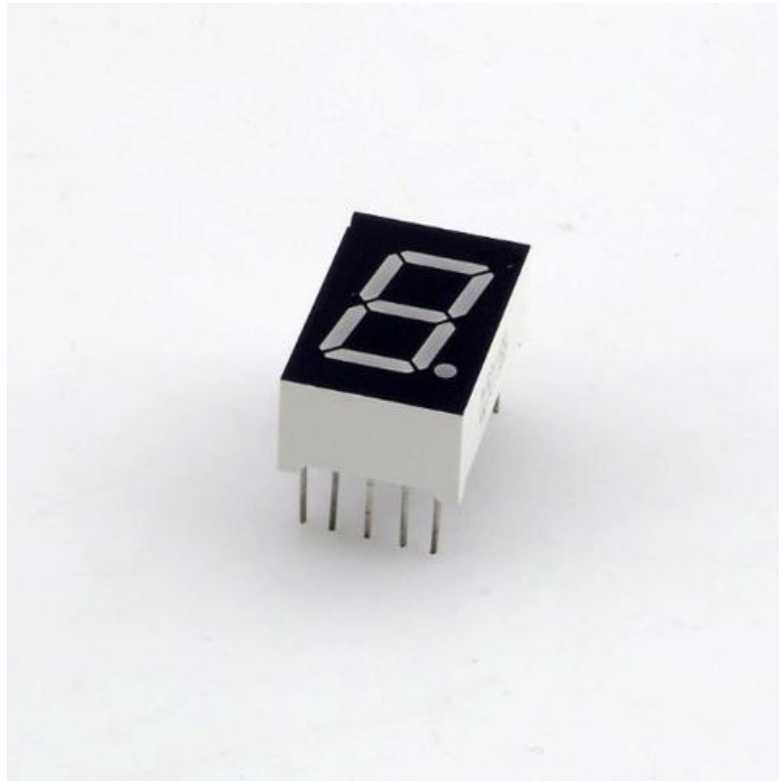
1bit digital tube *1

220 ohm resistor *8

Breadboard*1

Some DuPont lines

Product description:



●Function: digital tube display digital tube is a kind of semiconductor light emitting devices, the basic unit is the light-emitting diode, digital tube according to the number of segments is divided into seven segment digital tube and eight digital tube, eight digital tube than seven segment digital tube more than a light emitting diode unit (more than one decimal point), the use of eight digital tube.

●application: Digital tube is a common display digital display, daily life, for example: electromagnetic oven, full automatic washing machine, solar water temperature display, electronic clock

●According to the light emitting diode unit connection mode is divided into

common anode and cathode nixie tube. Yang digital tube is refers to all the light emitting diode anode received together form a common anode (COM) digital tube anode in the application should be public com received + 5V, when a light emitting diode for low electricity at ordinary times When a field of cathode for high electric power, the corresponding field is not bright. Common cathode digital tube is all led to form a common cathode (COM) digital tube. Common cathode digital tube in the application should be COM to the ground GND, when a field light-emitting diode anode for high power, the corresponding field is lit. When a field of anode for low power, the corresponding field is not bright.

Technical Parameters:

Product Name : LED Digital Tube;

Type : Common Cathode;

Emitted Color : Red

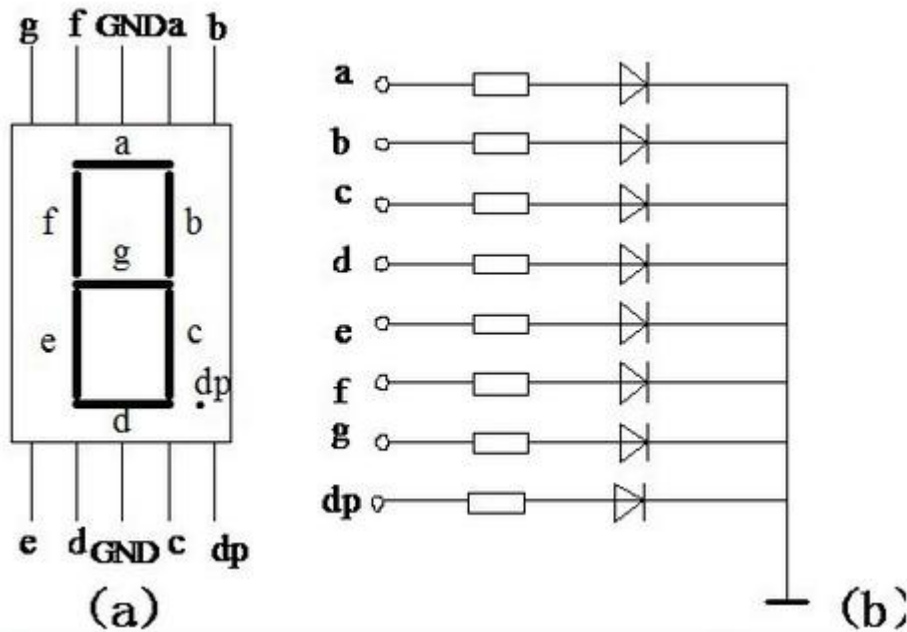
Pin Number : 10;

Pin Pitch : 2mm/0.08 inch

Material : Plastic, Metal;

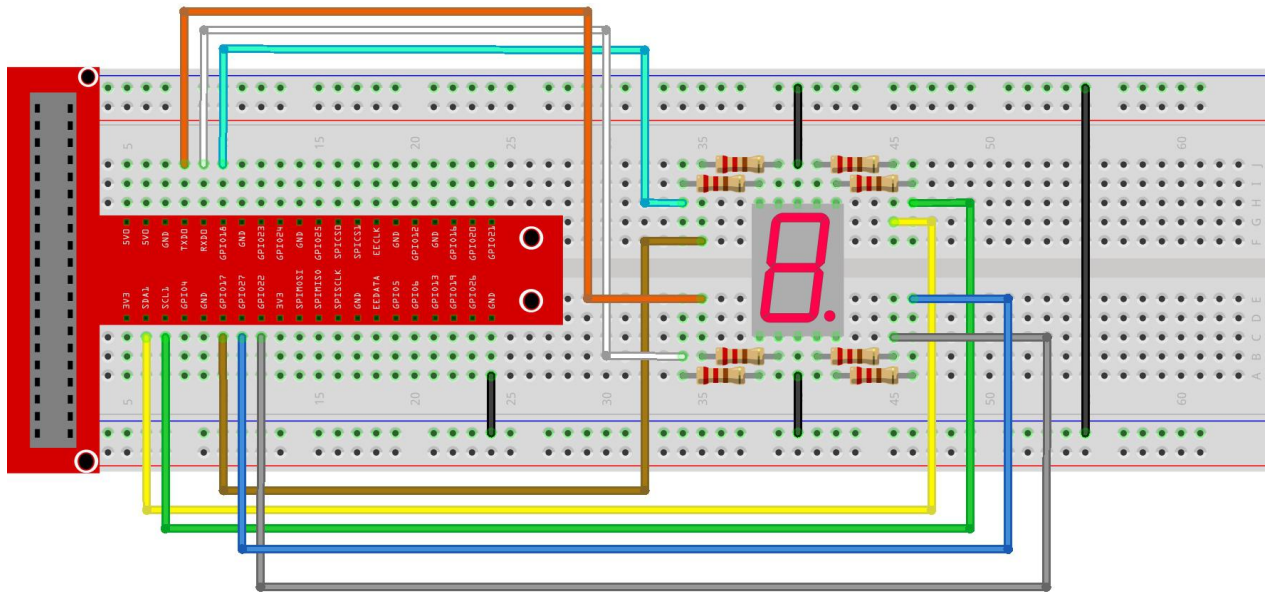
Color : Black, White

Net Weight : 38g;



Each of the digital tube is composed of light-emitting diodes, so when using the same as the light-emitting diode, but also to connect the current limit resistance, otherwise the current will burn the light emitting diode. This experiment with a common cathode digital tube, common cathode digital tube in the application should be GND, when a field light-emitting diode anode for high power, the corresponding field points out

Wiring diagram:



C code:

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

//Set pin control all digital IO
int pinA = 8;
int pinB = 9;
int pinC = 3;
int pinD = 15;
int pinE = 16;
int pinF = 0;
int pinG = 1;
int pinDP = 2;
```

```

void digital_0(void) //Display number 0
{
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, LOW);
    digitalWrite(pinDP, LOW);
}

void digital_1(void) //Display number 1
{
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
    digitalWrite(pinDP, LOW);
}

void digital_2(void) //Display number 2
{
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinDP, LOW);
}

void digital_3(void) //Display number3
{
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinDP, LOW);
}

```

```

}
void digital_4(void) //Display number 4
{
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinDP, LOW);
}
void digital_5(void) //Display number 5
{
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinDP, LOW);
}
void digital_6(void) //Display number 6
{
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinDP, LOW);
}
void digital_7(void) //Display number7
{
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
    digitalWrite(pinDP, LOW);
}

```

```

}
void digital_8(void) //Display number 8
{
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinDP, LOW);
}
void digital_9(void) //Display number 9
{
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, HIGH);
    digitalWrite(pinDP, LOW);
}

void digital_dp(void) //Display dp
{
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
    digitalWrite(pinDP, HIGH);
}

void clear(void) //clear
{
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);

```



```

        digitalWrite(pinG, LOW);
        digitalWrite(pinDP, LOW);
    }
    void init()
    {
        pinMode(pinA, OUTPUT);
        pinMode(pinB, OUTPUT);
        pinMode(pinC, OUTPUT);
        pinMode(pinD, OUTPUT);
        pinMode(pinE, OUTPUT);
        pinMode(pinF, OUTPUT);
        pinMode(pinG, OUTPUT);
        pinMode(pinDP, OUTPUT);
    }

    void main(void)
    {
        if(wiringPiSetup() == -1)
        {
            printf("wiringPi setup failed!\n");
            return -1;
        }

        init();
        while(1)
        {
            digital_0();
            delay(1000);
            digital_1();
            delay(1000);
            digital_2();
            delay(1000);
            digital_3();
            delay(1000);
            digital_4();
            delay(1000);
            digital_5();
            delay(1000);
            digital_6();
            delay(1000);
            digital_7();
            delay(1000);
            digital_8();
            delay(1000);

```

```
        digital_9();
        delay(1000);
        digital_dp();
        delay(1000);
        clear();
        delay(1000);
    }

}
```

Python Code:

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

pinA = 3
pinB = 5
pinC = 15
pinD = 8
pinE = 10
pinF = 11
pinG = 12
pinDP = 13

def init():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(pinA, GPIO.OUT)
    GPIO.setup(pinB, GPIO.OUT)
    GPIO.setup(pinC, GPIO.OUT)
    GPIO.setup(pinD, GPIO.OUT)
    GPIO.setup(pinE, GPIO.OUT)
    GPIO.setup(pinF, GPIO.OUT)
    GPIO.setup(pinG, GPIO.OUT)
    GPIO.setup(pinDP, GPIO.OUT)
    print 'gpio init completed!'

def display_0():
    GPIO.output(pinA, GPIO.HIGH)
    GPIO.output(pinB, GPIO.HIGH)
    GPIO.output(pinC, GPIO.HIGH)
```

```
GPIO.output(pinD, GPIO.HIGH)
GPIO.output(pinE, GPIO.HIGH)
GPIO.output(pinF, GPIO.HIGH)
GPIO.output(pinG, GPIO.LOW)
GPIO.output(pinDP, GPIO.LOW)
print 'display number 0'
```

```
def display_1():
    GPIO.output(pinA, GPIO.LOW)
    GPIO.output(pinB, GPIO.HIGH)
    GPIO.output(pinC, GPIO.HIGH)
    GPIO.output(pinD, GPIO.LOW)
    GPIO.output(pinE, GPIO.LOW)
    GPIO.output(pinF, GPIO.LOW)
    GPIO.output(pinG, GPIO.LOW)
    GPIO.output(pinDP, GPIO.LOW)
    print 'display number 1'
```

```
def display_2():
    GPIO.output(pinA, GPIO.HIGH)
    GPIO.output(pinB, GPIO.HIGH)
    GPIO.output(pinC, GPIO.LOW)
    GPIO.output(pinD, GPIO.HIGH)
    GPIO.output(pinE, GPIO.HIGH)
    GPIO.output(pinF, GPIO.LOW)
    GPIO.output(pinG, GPIO.HIGH)
    GPIO.output(pinDP, GPIO.LOW)
    print 'display number 2'
```

```
def display_3():
    GPIO.output(pinA, GPIO.HIGH)
    GPIO.output(pinB, GPIO.HIGH)
    GPIO.output(pinC, GPIO.HIGH)
    GPIO.output(pinD, GPIO.HIGH)
    GPIO.output(pinE, GPIO.LOW)
    GPIO.output(pinF, GPIO.LOW)
    GPIO.output(pinG, GPIO.HIGH)
    GPIO.output(pinDP, GPIO.LOW)
    print 'display number 3'
```

```
def display_4():
    GPIO.output(pinA, GPIO.LOW)
    GPIO.output(pinB, GPIO.HIGH)
```

```
GPIO.output(pinC, GPIO.HIGH)
GPIO.output(pinD, GPIO.LOW)
GPIO.output(pinE, GPIO.LOW)
GPIO.output(pinF, GPIO.HIGH)
GPIO.output(pinG, GPIO.HIGH)
GPIO.output(pinDP, GPIO.LOW)
print 'display number 4'
```

```
def display_5():
    GPIO.output(pinA, GPIO.HIGH)
    GPIO.output(pinB, GPIO.LOW)
    GPIO.output(pinC, GPIO.HIGH)
    GPIO.output(pinD, GPIO.HIGH)
    GPIO.output(pinE, GPIO.LOW)
    GPIO.output(pinF, GPIO.HIGH)
    GPIO.output(pinG, GPIO.HIGH)
    GPIO.output(pinDP, GPIO.LOW)
    print 'display number 5'
```

```
def display_6():
    GPIO.output(pinA, GPIO.HIGH)
    GPIO.output(pinB, GPIO.LOW)
    GPIO.output(pinC, GPIO.HIGH)
    GPIO.output(pinD, GPIO.HIGH)
    GPIO.output(pinE, GPIO.HIGH)
    GPIO.output(pinF, GPIO.HIGH)
    GPIO.output(pinG, GPIO.HIGH)
    GPIO.output(pinDP, GPIO.LOW)
    print 'display number 6'
```

```
def display_7():
    GPIO.output(pinA, GPIO.HIGH)
    GPIO.output(pinB, GPIO.HIGH)
    GPIO.output(pinC, GPIO.HIGH)
    GPIO.output(pinD, GPIO.LOW)
    GPIO.output(pinE, GPIO.LOW)
    GPIO.output(pinF, GPIO.LOW)
    GPIO.output(pinG, GPIO.LOW)
    GPIO.output(pinDP, GPIO.LOW)
    print 'display number 7'
```

```
def display_8():
    GPIO.output(pinA, GPIO.HIGH)
```

```
GPIO.output(pinB, GPIO.HIGH)
GPIO.output(pinC, GPIO.HIGH)
GPIO.output(pinD, GPIO.HIGH)
GPIO.output(pinE, GPIO.HIGH)
GPIO.output(pinF, GPIO.HIGH)
GPIO.output(pinG, GPIO.HIGH)
GPIO.output(pinDP, GPIO.LOW)
print 'display number 8'
```

```
def display_9():
    GPIO.output(pinA, GPIO.HIGH)
    GPIO.output(pinB, GPIO.HIGH)
    GPIO.output(pinC, GPIO.HIGH)
    GPIO.output(pinD, GPIO.HIGH)
    GPIO.output(pinE, GPIO.LOW)
    GPIO.output(pinF, GPIO.HIGH)
    GPIO.output(pinG, GPIO.HIGH)
    GPIO.output(pinDP, GPIO.LOW)
    print 'display number 9'
```

```
def display_dp():
    GPIO.output(pinA, GPIO.LOW)
    GPIO.output(pinB, GPIO.LOW)
    GPIO.output(pinC, GPIO.LOW)
    GPIO.output(pinD, GPIO.LOW)
    GPIO.output(pinE, GPIO.LOW)
    GPIO.output(pinF, GPIO.LOW)
    GPIO.output(pinG, GPIO.LOW)
    GPIO.output(pinDP, GPIO.HIGH)
    print 'display dp'
```

```
def clear():
    GPIO.output(pinA, GPIO.LOW)
    GPIO.output(pinB, GPIO.LOW)
    GPIO.output(pinC, GPIO.LOW)
    GPIO.output(pinD, GPIO.LOW)
    GPIO.output(pinE, GPIO.LOW)
    GPIO.output(pinF, GPIO.LOW)
    GPIO.output(pinG, GPIO.LOW)
    GPIO.output(pinDP, GPIO.LOW)
    print 'clear'
```

```
def loop():
```

```

while True:
    display_0()
    time.sleep(1)
    display_1()
    time.sleep(1)
    display_2()
    time.sleep(1)
    display_3()
    time.sleep(1)
    display_4()
    time.sleep(1)
    display_5()
    time.sleep(1)
    display_6()
    time.sleep(1)
    display_7()
    time.sleep(1)
    display_8()
    time.sleep(1)
    display_9()
    time.sleep(1)
    display_dp()
    time.sleep(1)
    clear()
    time.sleep(1)

if __name__ == '__main__':
    try:
        init()
        loop()
    except KeyboardInterrupt:
        GPIO.cleanup()
        print 'Key Board Interrupt!'

```

Experimental results:

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

C:

```

gcc -Wall -o 8segment_digital_tube 8segment_digital_tube.c -lwiringPi
sudo ./8segment_digital_tube

```

Python:

```
python 8segment_digital_tube.py
```

After the instruction is executed, the digital tube will display the number 0-9

