**8 Segment Digital Tube**

Overview

The purpose of this course is to use the Raspberry Pi to control the eight segment digital tubes, which will display the number 0~9 on the digital tube.

The Experimental Materials

Raspberry Pi \* 1

Breadboard \* 1

8 segment digital tube \* 1

Dupont Line

The Preparatory Work

1. Install the python interpreter in your Raspberry Pi system.

2. Install the rpi.gpio library in your Raspberry Pi system.

3. Install wiring Pi library in your Raspberry Pi system.

For more details, please refer to the attachment for installing the python interpreter and corresponding libraries in the Raspberry Pi system.

Product Description

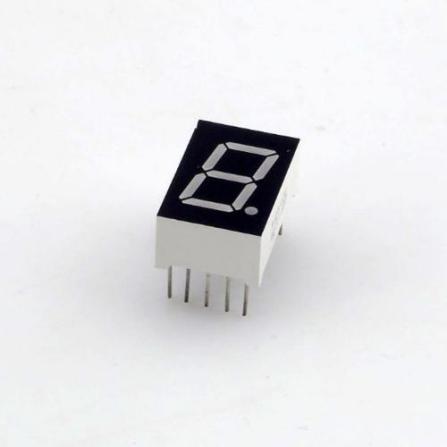
Brief Introduction

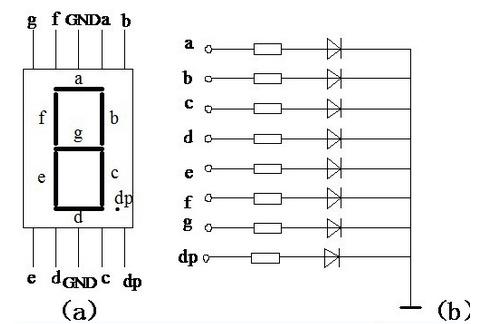
●Function: A digital tube is a semiconductor luminescent device whose basic unit is a light-emitting diode. The digital tube is divided into seven digital tubes and eight digital tubes. The eight-segment digital tube has more than one light-emitting diode unit (more than one decimal point) compared to the seven-segment digital tube. This experiment uses eight digital tubes.

●Application: Induction cooker, automatic washing machine, solar water temperature display and electronic clock.

●The light-emitting diode unit connection is divided into a common anode digital tube and a common cathode digital tube. The common anode digital tube is a digital tube that combines all the leds to form a common anode (COM).In the application of the common anode digital tube, the common pole COM should be connected to +5V. When the cathode of a field led is low, the corresponding field is lit.When the cathode of a field is high, the corresponding field is not bright.

The cathode digital tube is a digital tube that connects all the emitting diodes to a common cathode (COM).In the application, the cathode digital tube should connect the common pole COM to the ground line GND. When the anode of a field light-emitting diode is high, the corresponding field will be lit. When the anode of a field is low, the corresponding field is not bright.





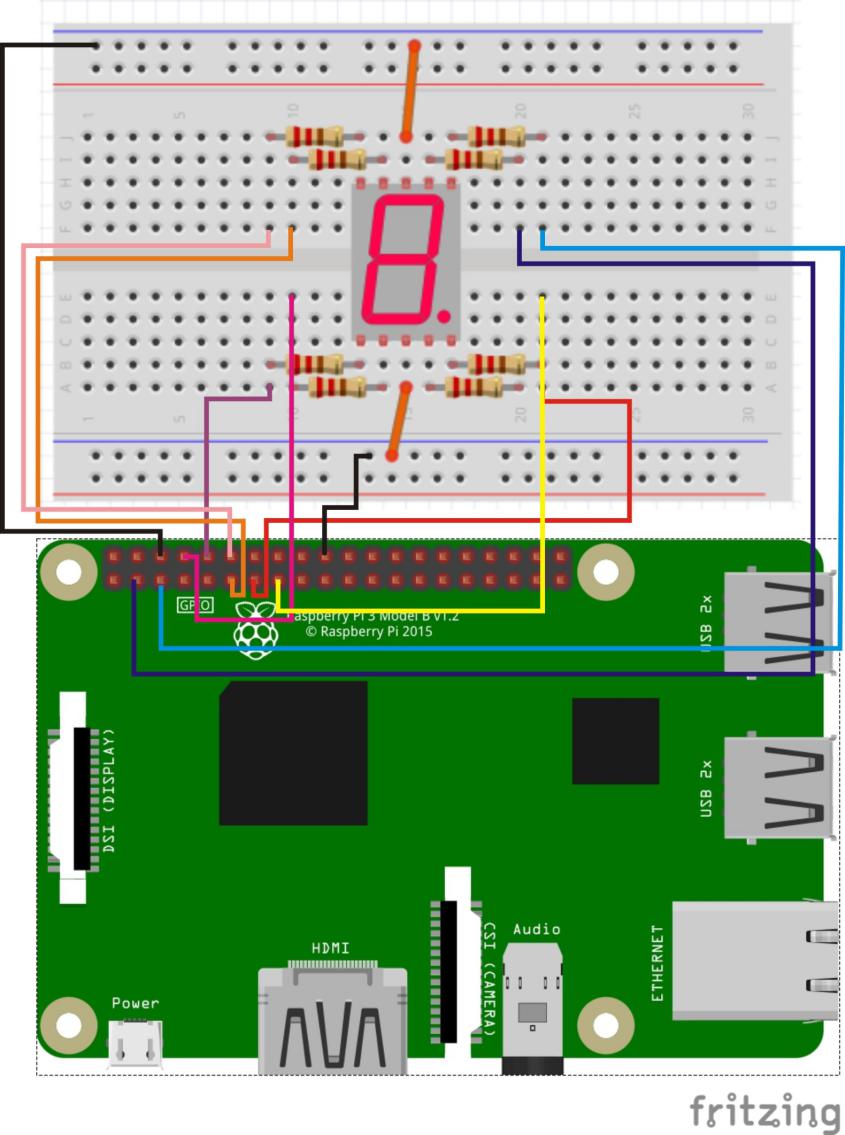
**Characteristic Parameters**

◆Type: Common Cathode ◆Color: Red

◆Number of Pins: 10 ◆Pin Spacing: 2mm / 0.08 inch

◆Material: Plastic, Metal ◆Color: Black and White

Wiring diagram



**The sample code**

1. **Python Code**

#!/usr/bin/env python

import RPi.GPIO as GPIO

import time

pinA = 3

pinB = 5

pinC = 16

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!'

1. **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 5

{

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);

}

}

**The experimental phenomena**

The digital tube displays the number 0~9 and the decimal point in cycle.