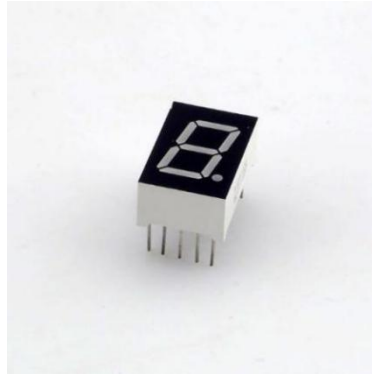


## 1 Digit LED Display (aka 8-Segment Digital Tube)



### Overview

“8-Segment Digital Tube” is the conventional name for a display capable of showing one decimal digit, where any possible digit, in turn, is composed of up to seven separate LED segments. (A decimal point potentially trailing the digit represents an eighth possible LED.) Such low-cost, low-power numeric displays are common in microwave ovens, alarm clocks, induction cookers, automatic washing machines, and similar devices. In this experiment, you’ll program the Raspberry Pi to show the digits 0 through 9, sequentially, on the LED display.

### Experimental Materials

Raspberry Pi	x1
Breadboard	x1
1 Digit LED Display	x1
Dupont jumper wires	
Resistors (330 $\Omega$ )	x8

### Experimental Procedure

1. If you have not done so already, prepare your development system by installing the Python interpreter, RPi.GPIO library, and wiringPi library as described in `READ_ME_FIRST.TXT`.
2. Install the 1 Digit LED Display on your breadboard, and use Dupont jumper wires and resistors to connect it to your Raspberry Pi as illustrated in the Wiring Diagram below.

3. Execute the sample stored in this experiment's subfolder.  
If using C, compile and execute the C code:

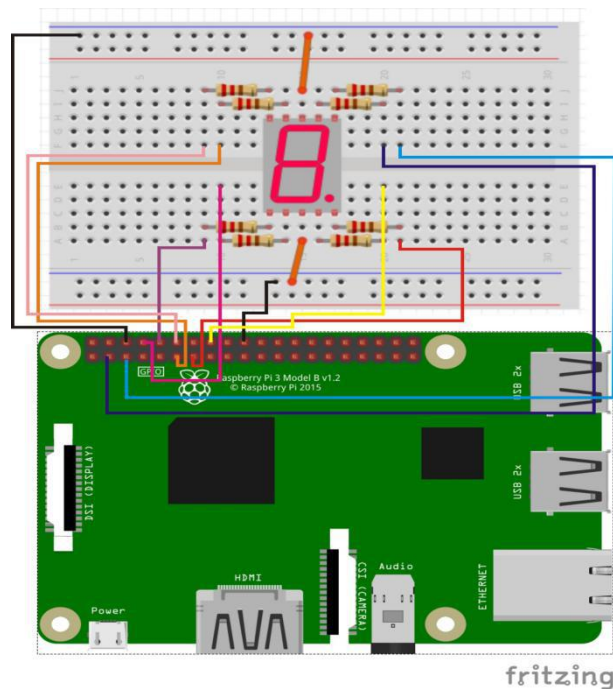
```
cd Code/C
gcc 1digitLEDdisplay.c -o 1digitLEDdisplay.out
-lwiringPi
./1digitLEDdisplay.out
```

If using Python, launch the Python script:

```
cd Code/Python
python 1digitLEDdisplay.py
```

4. Make experimental observations.  
The LED display cycles through 10 digits and the decimal point.

## Wiring Diagram



1-Digit LED display pin position:

- "a" ↔ Raspberry Pi pin 3 (through resistor)
- "b" ↔ Raspberry Pi pin 5 (through resistor)
- "c" ↔ Raspberry Pi pin 16 (through resistor)
- "d" ↔ Raspberry Pi pin 8 (through resistor)

"e" ↔ Raspberry Pi pin 10 (through resistor)

"f" ↔ Raspberry Pi pin 11 (through resistor)

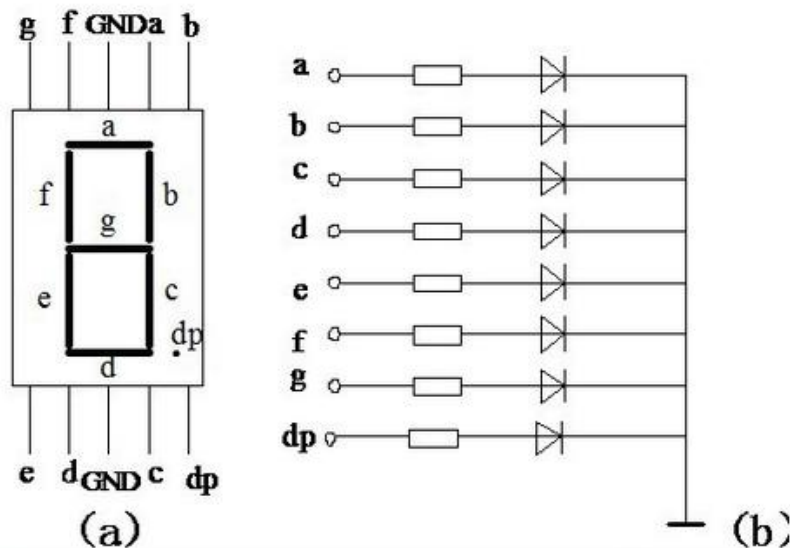
"g" ↔ Raspberry Pi pin 12 (through resistor)

"dp" ↔ Raspberry Pi pin 13 (through resistor)

GND ↔ Raspberry Pi GND

## Technical Background

The seven alphabetically-labeled data pins (**a-g**) correspond to the seven possible segments of the numeric digit display; the pin labeled **dp** corresponds to the decimal point. In the module's common cathode design, setting a pin HIGH illuminates the corresponding LED segment. Thus to display "1" requires setting only pins **b** and **c** high—for the two righthand vertical segments composing a "1", whereas to display "8." requires setting *all* pins (**a, b, c, d, e, f, g, dp**) high. Of course you can also set pin patterns that do not make up numeric digits at all! (For example, setting **b, c, e, f, g** would display an "H"-like shape.)



- ◆ Type: Common Cathode
- ◆ Number of Pins: 10
- ◆ Material: Plastic, Metal

- ◆ Color: Red
- ◆ Pin Spacing: 2mm / 0.08 inch
- ◆ Color: Black and White

## Sample Code

### 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!'
```

## C Code

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

//Set pin control all digital IO
int pinA = 8;
int pinB = 9;
int pinC = 4;
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 number 7
{
    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);
}
}
```