



Lesson6_7segments digital LED score board

7 segments digital LED score board is a digital LED display, and made by 7 LED lights, there are share a common ground pin, so we can control each pin to show different digital number, such as 0-9 and A to H. we man make a score board by using 7 segments digital LED and button.

Principal

The Raspberry Pi will detect the level of the GPIO pins When the button is pressed, it can be summed and display on the 7 segments digital LED.

Experiment Content

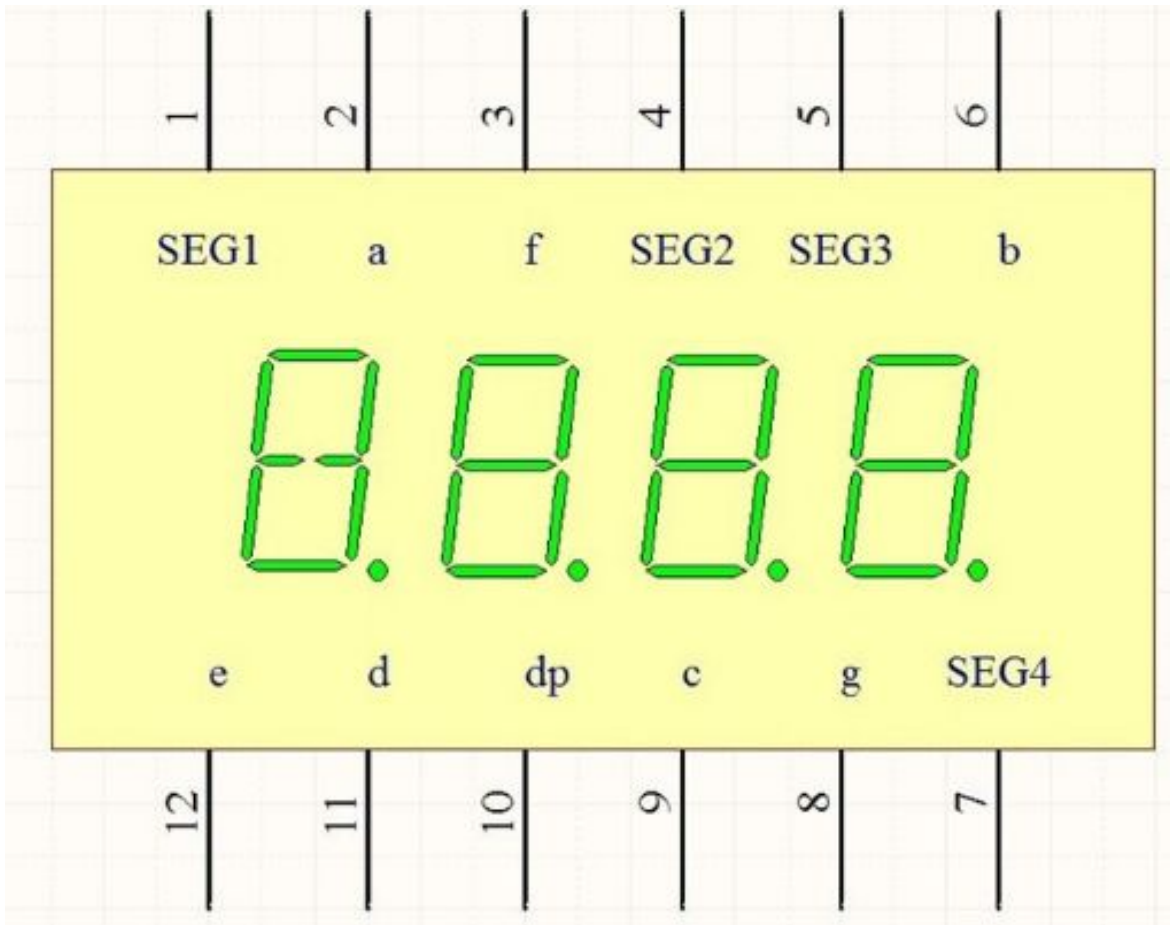
Learning how to use 7 segments digital LED
Learning how to detect button status

Components

Raspberry Pi 3, mode B, 1GB RAM	x1
Breadboard	x1
Button	x2
4bit 7 segments digital LED display	x1
220 Resister	x6
T extend board or Easy Multiplexing Board	x1
Cable wire	x n

All of those stuff can be purchased from: <https://52pi.taobao.com>

Step.1 Wire everything up as following chart, and following picture show you the pin's function of 4bits 7segments digital LED display.



We should connect each pin to the GPIO pin of raspberry Pi and make sure its connection is fine as following chart:

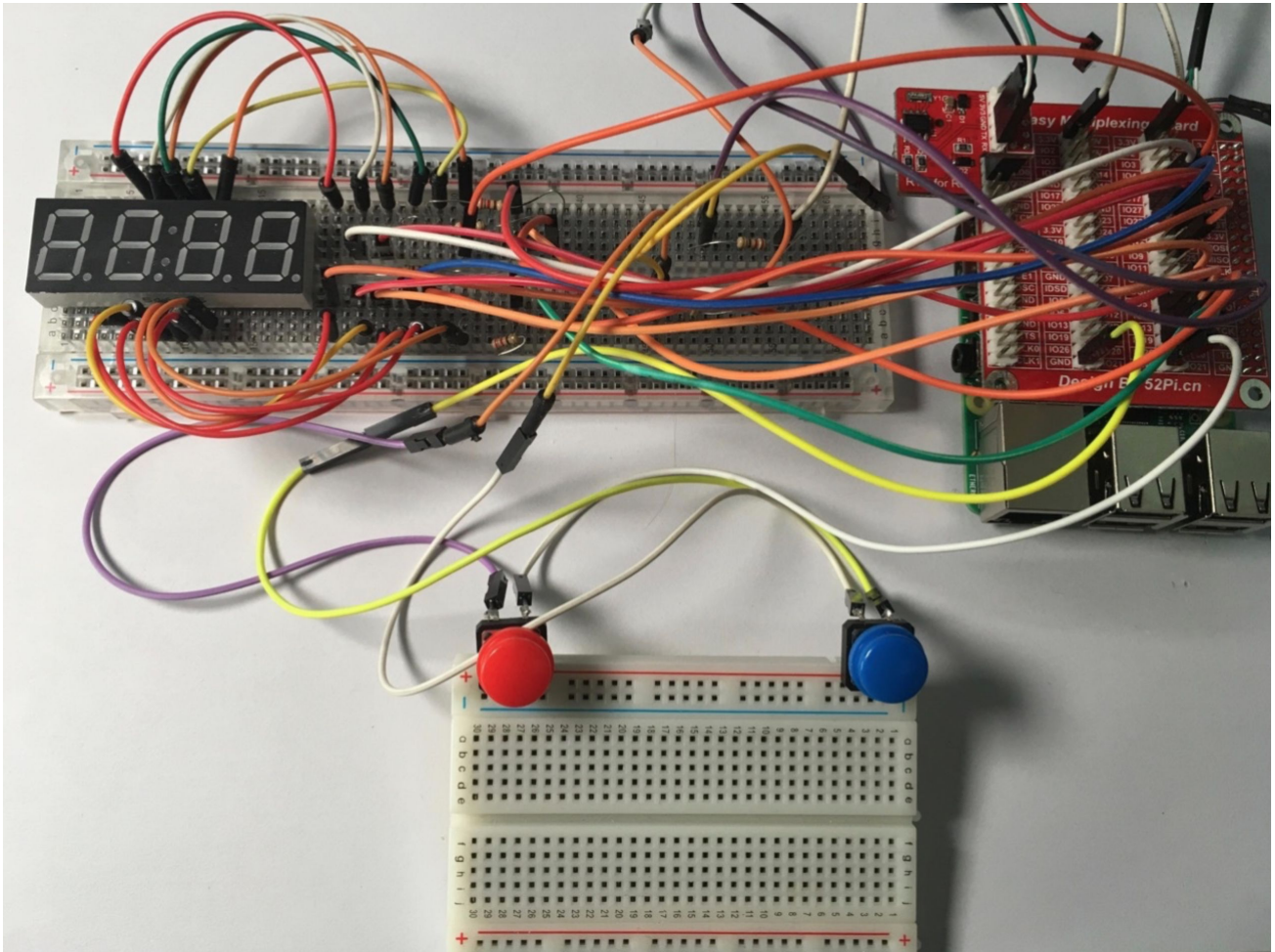
Name of 7segments digital LED pin	GPIO physical number	WiringPi	BCM
A	11	0	17
B	12	1	18
C	13	2	27
D	15	3	22
E	16	4	23
F	18	5	24
G	22	6	25
dp	29	21	5
Seg1	31	22	6
Seg2	33	23	13
Seg3	35	24	19
Seg4	37	25	26

You can extend the wire to breadboard so that you can wire everything up properly.

Please pay attention to that each 7segments digital LED pin must connect to a $220\ \Omega$ resistor before you connect them to GPIO pin on Raspberry Pi.

PS: too many wire so you may make a mistake, so I strongly recommend you using a easy multiplex extension board as showing.

You can chose two GPIO pin for the button, In this case, I setup BUTTONA as BCM 27 pin, and BUTTONB as BCM 26 pin.





Setup2. Coding:

You can use two way to drive this 7 segments digital LED display:

First way(we called easy way).

Open a terminal after login into system.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

After upgrading, download wiringPi and update it.

```
git clone git://git.drogon.net/wiringPi
```

```
cd wiringPi
```

```
git pull origin
```

Go back to home directory and get the code that already prepared for you from github.

```
cd
```

```
git clone https://github.com/yoyojacky/RPi-C-CODE.git
```

```
cd RPi-C-CODE/
```

```
gcc -o digitalled 4segment-digitalLED-with-button-controlling.c -lwiringPi
```

```
sudo ./digitalled
```

Then you can push the button and check whether the number are changed.

Hard way:

You can create a file named with .c suffix and compile it, you can just copy my code below in to a file called digitalled.c in a command line:

```
vim.tiny digitalled.c
```

copy following paragraph and paste in your file and save.

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

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

```
/* Define some parameters for GPIO pins */
```

```
#define DIGITAL_A 0
```

```
#define DIGITAL_B 1
```

```
#define DIGITAL_C 2
```

```
#define DIGITAL_D 3
```

```
#define DIGITAL_E 4
```

```
#define DIGITAL_F 5
```

```
#define DIGITAL_G 6
```

```
#define DIGITAL_DP 21
```

```
#define DIGITAL_SEG1 22
```

```
#define DIGITAL_SEG2 23
```

```
#define DIGITAL_SEG3 24
```

```
#define DIGITAL_SEG4 25
```

```
#define BUTTONA 27
```

```
#define BUTTONB 26
```



```
int x;
```

```
int i;
```

```
const char lednumber[10]=
```

```
{  
    0xfc, //0  
    0x60, //1  
    0xda, //2  
    0xf2, //3  
    0x66, //4  
    0xb6, //5  
    0xbe, //6  
    0xe0, //7  
    0xfe, //8  
    0xf6  //9  
};
```

```
void init_digitalled(void)
```

```
{  
    digitalWrite(DIGITAL_A,LOW);  
    digitalWrite(DIGITAL_B,LOW);  
    digitalWrite(DIGITAL_C,LOW);  
    digitalWrite(DIGITAL_D,LOW);  
    digitalWrite(DIGITAL_E,LOW);  
    digitalWrite(DIGITAL_F,LOW);  
    digitalWrite(DIGITAL_G,LOW);  
    digitalWrite(DIGITAL_DP,LOW);  
    digitalWrite(DIGITAL_SEG1,HIGH);  
    digitalWrite(DIGITAL_SEG2,HIGH);  
    digitalWrite(DIGITAL_SEG3,HIGH);  
    digitalWrite(DIGITAL_SEG4,HIGH);  
}
```

```
void light_led(int x)
```

```
{  
    digitalWrite(DIGITAL_A, (lednumber[x] & 0x80));  
    digitalWrite(DIGITAL_B, (lednumber[x] & 0x40));  
    digitalWrite(DIGITAL_C, (lednumber[x] & 0x20));  
    digitalWrite(DIGITAL_D, (lednumber[x] & 0x10));  
    digitalWrite(DIGITAL_E, (lednumber[x] & 0x08));  
    digitalWrite(DIGITAL_F, (lednumber[x] & 0x04));  
    digitalWrite(DIGITAL_G, (lednumber[x] & 0x02));  
    digitalWrite(DIGITAL_DP, (lednumber[x] & 0x01));  
}
```



```
void display_led(int num)
{
    int number = num;
    int led1 = 0;
    int led2 = 0;
    int led3 = 0;
    int led4 = 0;

    led4 = number%10;
    led3 = number/10%10;
    led2 = number/100%10;
    led1 = number/1000%10;

    light_led(led1);
    digitalWrite(DIGITAL_SEG1,LOW);
    delay(1);
    digitalWrite(DIGITAL_SEG1,HIGH);

    light_led(led2);
    digitalWrite(DIGITAL_SEG2,LOW);
    delay(1);
    digitalWrite(DIGITAL_SEG2,HIGH);

    light_led(led3);
    digitalWrite(DIGITAL_SEG3,LOW);
    delay(1);
    digitalWrite(DIGITAL_SEG3,HIGH);

    light_led(led4);
    digitalWrite(DIGITAL_SEG4,LOW);
    delay(1);
    digitalWrite(DIGITAL_SEG4,HIGH);
}

void init_all(void)
{
    wiringPiSetup();
    pinMode(DIGITAL_A,OUTPUT);
    pinMode(DIGITAL_B,OUTPUT);
    pinMode(DIGITAL_C,OUTPUT);
    pinMode(DIGITAL_D,OUTPUT);
    pinMode(DIGITAL_E,OUTPUT);
    pinMode(DIGITAL_F,OUTPUT);
```



```
pinMode(DIGITAL_G,OUTPUT);
pinMode(DIGITAL_DP,OUTPUT);
pinMode(DIGITAL_SEG1,OUTPUT);
pinMode(DIGITAL_SEG2,OUTPUT);
pinMode(DIGITAL_SEG3,OUTPUT);
pinMode(DIGITAL_SEG4,OUTPUT);
pinMode(BUTTONA,INPUT);
pinMode(BUTTONB,INPUT);
}

int main()
{
  init_all();
  init_digitalled();
  for(;;)
  {
    display_led(i);
    if(digitalRead(BUTTONA) == 0 || digitalRead(BUTTONB) == 0)
    {
      for(x=0;x<=20;x++)
      {
        display_led(i);
      };

      if(digitalRead(BUTTONA) == 0)
      {
        printf("ButtonA is pressed! and you press %d times\n", i++);
      };

      if(digitalRead(BUTTONB) == 0)
      {
        if (i>0)
        {
          printf("ButtonB was pressed, and you have been pressed %d times\n", i--);
        };
      };
    };
  };
}
```

Save and quit, compile it as follow in command in your terminal:

```
gcc -o digitalled digitalled.c -lwiringPi
```

Run as:

```
sudo ./digitalled
```

when you running this program you will see following picture, and you can just press the button and increase the number.

```
pi@raspberrypi:~$ ls
Desktop  Downloads  Pictures  python_games  Templates  wiringPi
Documents  Music      Public    RPi-C-CODE    Videos
pi@raspberrypi:~/RPi-C-CODE$ cd RPi-C-CODE/
pi@raspberrypi:~/RPi-C-CODE$ ls
4segment-digitalLED-with-button-controlling.c  digitalLED.c  README.md
digitalled                                     LICENSE.md
pi@raspberrypi:~/RPi-C-CODE$
pi@raspberrypi:~/RPi-C-CODE$ gcc -lwiringPi-CODE$ gcc -o digitalled 4segment-digitalLED-with-button-co
pi@raspberrypi:~/RPi-C-CODE$
pi@raspberrypi:~/RPi-C-CODE$
pi@raspberrypi:~/RPi-C-CODE$ sudo ./digitalled
BottonA is pressed! and you press 0 times
ButtonB was pressed, and you have been pressed 1 times
BottonA is pressed! and you press 0 times
BottonA is pressed! and you press 1 times
BottonA is pressed! and you press 2 times
BottonA is pressed! and you press 3 times
BottonA is pressed! and you press 4 times
BottonA is pressed! and you press 5 times
BottonA is pressed! and you press 6 times
BottonA is pressed! and you press 7 times
ButtonB was pressed, and you have been pressed 8 times
ButtonB was pressed, and you have been pressed 7 times
ButtonB was pressed, and you have been pressed 6 times
ButtonB was pressed, and you have been pressed 5 times
ButtonB was pressed, and you have been pressed 4 times
ButtonB was pressed, and you have been pressed 3 times
ButtonB was pressed, and you have been pressed 2 times
ButtonB was pressed, and you have been pressed 1 times
```

it will display on 4bits 7segments digital LED display.

