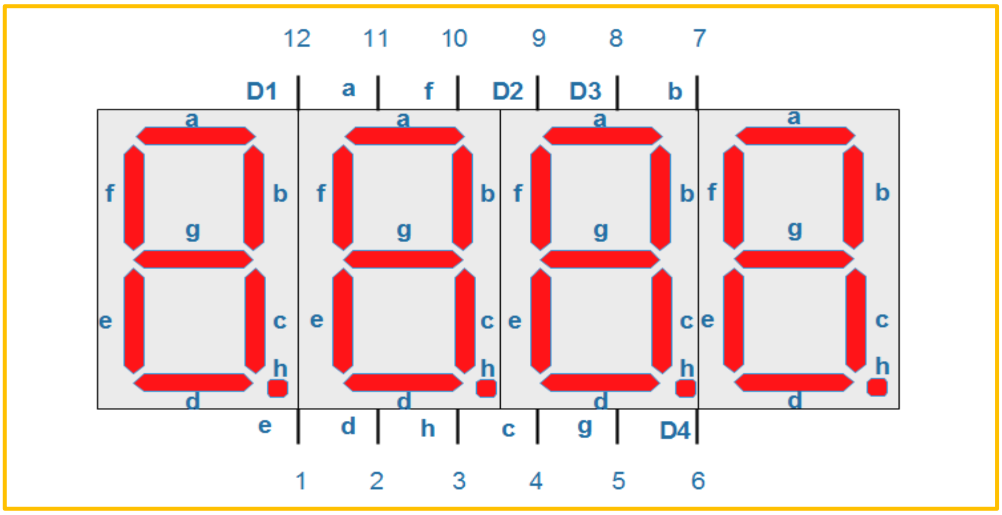
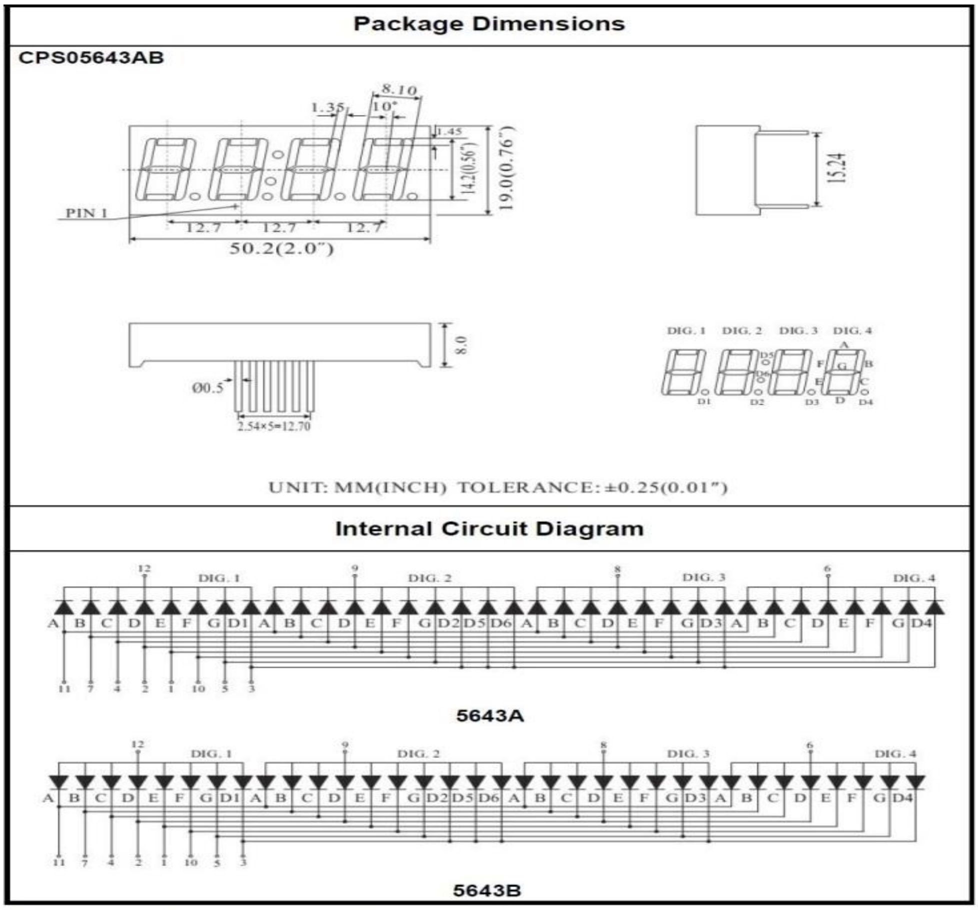
4-Digit 7-Segment Display

Introduction

We used a 7-segment tube before. When we want to display more than one number, then multidigit tube is required. Here we introduce four digital tube, actually each individual 7-segment tube is almost the same as the tube used above. In this experiment, we will use the Arduino to drive a common anode four digital tube.





Four Digits Displays Series

Four digital tube has 12 pins. The upper left is the biggest number 12 pin. Besides the 8-segment we used to display "adbcdefg", there are another 4 pins D1, D2, D3, D4 to be used as the "bit" pins. When the "bit" pins of common anode four digital tube is high level, the corresponding tubes light up. The display principle of four digital tube is that constantly scanning D1, D2, D3, D4, and then the corresponding eight-segment tubes will light up in turn. Due to the residual effect of human eye, so it looks like the four digital tube display at the same time.

With the principle introduced above, we now make a simulated countdown time bomb like the movies do. The bomb will exploded in one minute.

Experiment Principle

The most important purpose of this program is how to scan the four digital tube dynamically. In fact, with the single digital tube display experiment before, the display of four digital tube is quite easy. Due to it is attributed to a common anode tube, first of all, we are going to set D1, D2, D3, D4 to low level, all LED turn out, then we output the truth table of "adbcdefg" to the corresponding gpio port, select the corresponding bit pins and scan constantly. How to implement the 1 minute countdown? In the program, we will continuously get the current time through millis () function and determine whether it is greater than 1000ms. If so compared to the time before, the countdown time minus 1, then it is translated into character string that the Nixie tube displays.

Experiment Purpose

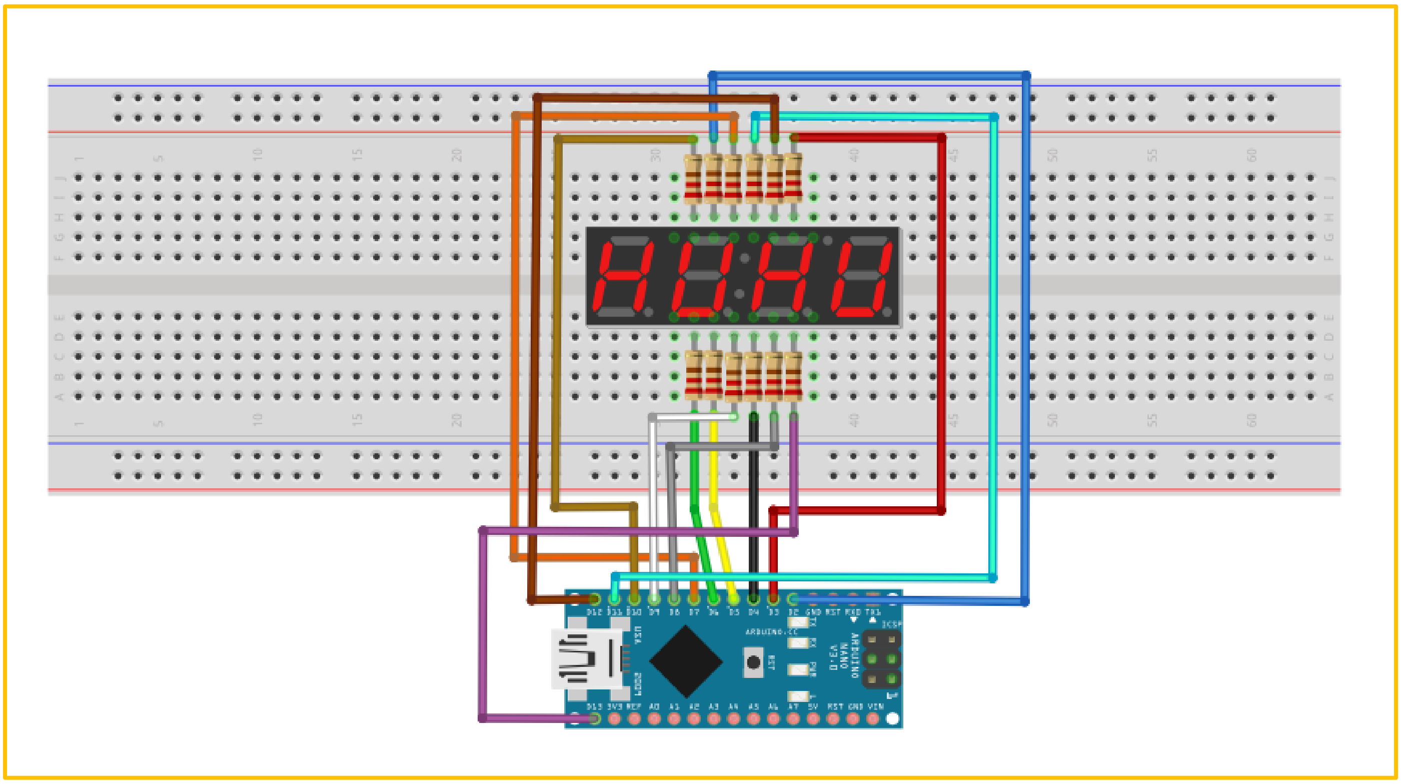
The aim is to display “1234” four characters via dynamically scanning 4-Digit 7-Segment Display.

Component List

- Arduino Nano Mainboard
- Breadboard
- USB cable
- 4-Digit 7-Segment Display
- 1k Resistor * 12
- Several wires

Wiring of Circuit

Arduino Nano	4-Digit 7-Segment Display
2	11(a)
3	7(b)
4	4(c)
5	2(d)
6	1(e)
7	10(f)
8	5(g)
9	3(h)
10	12(D1)
11	9(D2)
12	8(D3)
13	6(D4)



Code

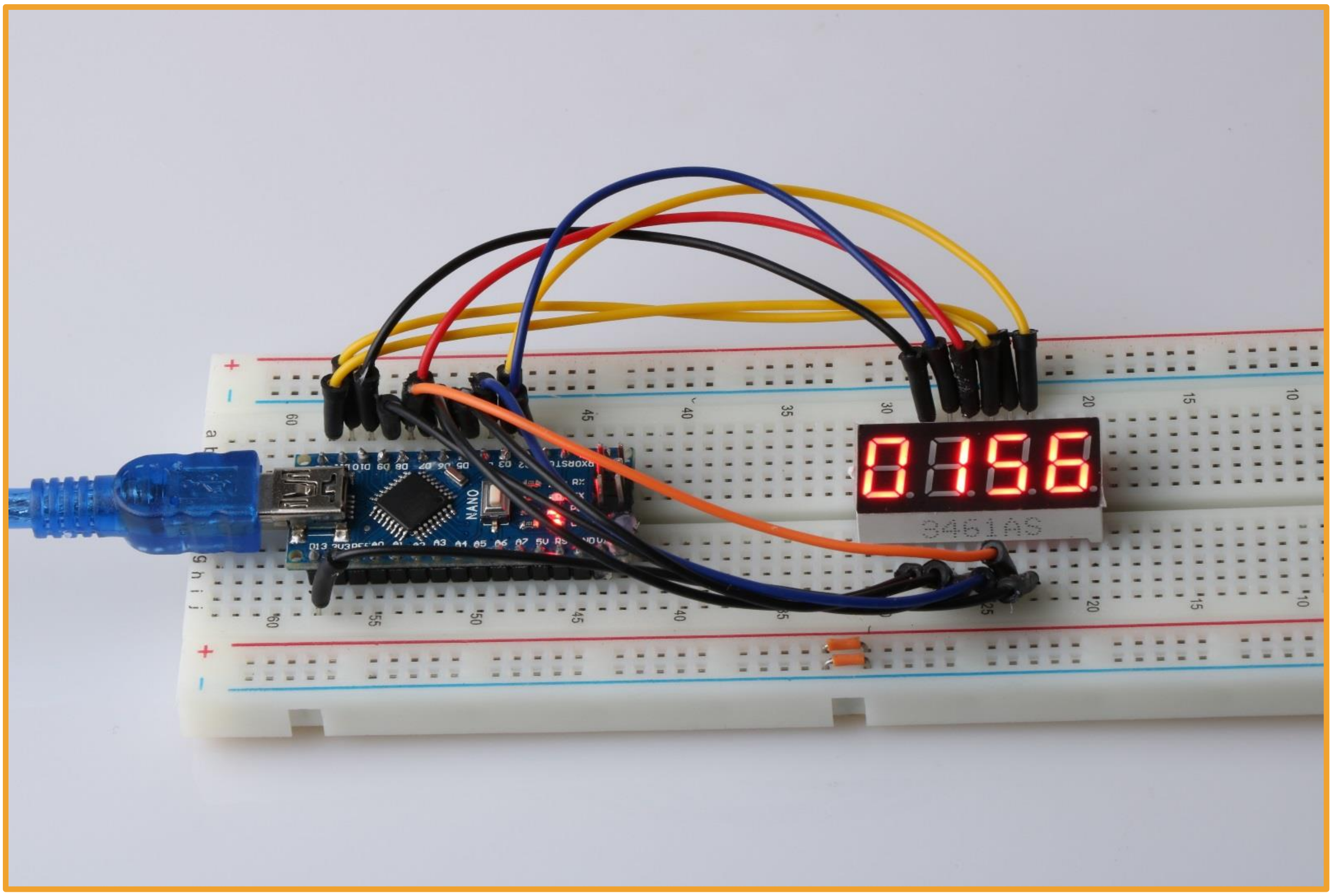
```
#define SEG_A 2
#define SEG_B 3
#define SEG_C 4
#define SEG_D 5
```

```
#define SEG_E 6
#define SEG_F 7
#define SEG_G 8
#define SEG_H 9

#define COM1 10
#define COM2 11
#define COM3 12
#define COM4 13


unsigned char table[10][8] =
{
   {0,   0, 1, 1, 1, 1, 1, 1},        //0
   {0,   0, 0, 0, 0, 1, 1, 0},        //1
   {0,   1, 0, 1, 1, 0, 1, 1},        //2
   {0,   1, 0, 0, 1, 1, 1, 1},        //3
   {0,   1, 1, 0, 0, 1, 1, 0},        //4
   {0,   1, 1, 0, 1, 1, 0, 1},        //5
   {0,   1, 1, 1, 1, 1, 0, 1},        //6
   {0,   0, 0, 0, 0, 1, 1, 1},        //7
   {0,   1, 1, 1, 1, 1, 1, 1},        //8
   {0,   1, 1, 0, 1, 1, 1, 1}      //9
};

void setup()
{
   pinMode(SEG_A,OUTPUT);    //设置为输出引脚
   pinMode(SEG_B,OUTPUT);
   pinMode(SEG_C,OUTPUT);
   pinMode(SEG_D,OUTPUT);
   pinMode(SEG_E,OUTPUT);
   pinMode(SEG_F,OUTPUT);
   pinMode(SEG_G,OUTPUT);
   pinMode(SEG_H,OUTPUT);

   pinMode(COM1,OUTPUT);
   pinMode(COM2,OUTPUT);
   pinMode(COM3,OUTPUT);
   pinMode(COM4,OUTPUT);
}
```

```
void loop()
{
   Display(1,1);        //第1位显示1
   delay(500);
   Display(2,2);        //第2位显示2
   delay(500);
   Display(3,3);        //第3位显示3
   delay(500);
   Display(4,4);        //第4位显示4
   delay(500);
}

void Display(unsigned char com,unsigned char num)
{
   digitalWrite(SEG_A,LOW);        //去除余晖
   digitalWrite(SEG_B,LOW);
   digitalWrite(SEG_C,LOW);
   digitalWrite(SEG_D,LOW);
   digitalWrite(SEG_E,LOW);
   digitalWrite(SEG_F,LOW);
   digitalWrite(SEG_G,LOW);
   digitalWrite(SEG_H,LOW);

   switch(com)                 //选通位选
   {
      case 1:
         digitalWrite(COM1,LOW);    //选择位1
         digitalWrite(COM2,HIGH);
         digitalWrite(COM3,HIGH);
         digitalWrite(COM4,HIGH);
         break;
      case 2:
         digitalWrite(COM1,HIGH);
         digitalWrite(COM2,LOW);    //选择位2
         digitalWrite(COM3,HIGH);
         digitalWrite(COM4,HIGH);
         break;
      case 3:
         digitalWrite(COM1,HIGH);
         digitalWrite(COM2,HIGH);
         digitalWrite(COM3,LOW);    //选择位3
         digitalWrite(COM4,HIGH);
```

```
            break;
        case 4:
            digitalWrite(COM1,HIGH);
            digitalWrite(COM2,HIGH);
            digitalWrite(COM3,HIGH);
            digitalWrite(COM4,LOW);    //选择位4
            break;
        default:break;
    }

    digitalWrite(SEG_A,table[num][7]);         //a查询码值表
    digitalWrite(SEG_B,table[num][6]);
    digitalWrite(SEG_C,table[num][5]);
    digitalWrite(SEG_D,table[num][4]);
    digitalWrite(SEG_E,table[num][3]);
    digitalWrite(SEG_F,table[num][2]);
    digitalWrite(SEG_G,table[num][1]);
    digitalWrite(SEG_H,table[num][0]);
}
```

Experiment Result



Notice : The 4 numeric digits are converted into the value of AscII by number2dis, say, we are going to convert “1234”, this should be as follows

Loop	numble	bit_base	disp
1	1234	1000	1
2	234	100	2
3	34	10	3
4	4	1	4