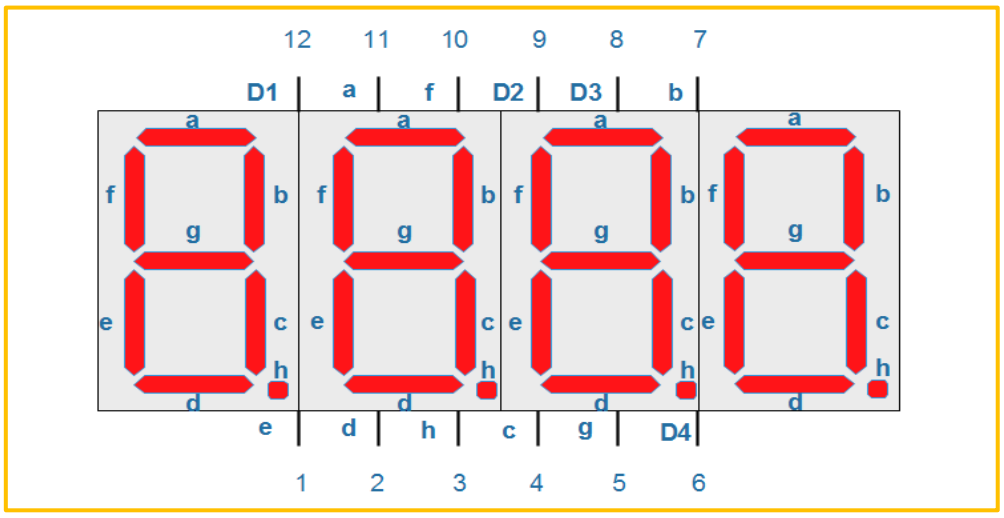
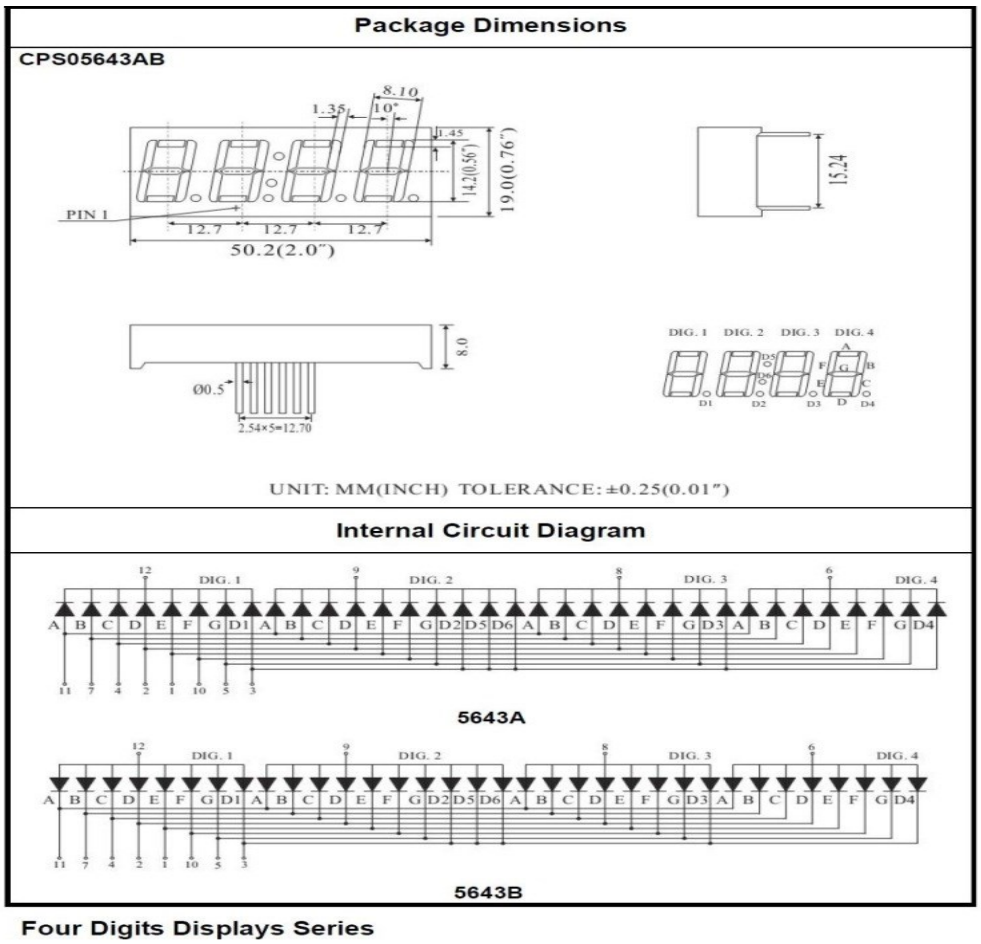
4-Digit 7-Segment Display

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

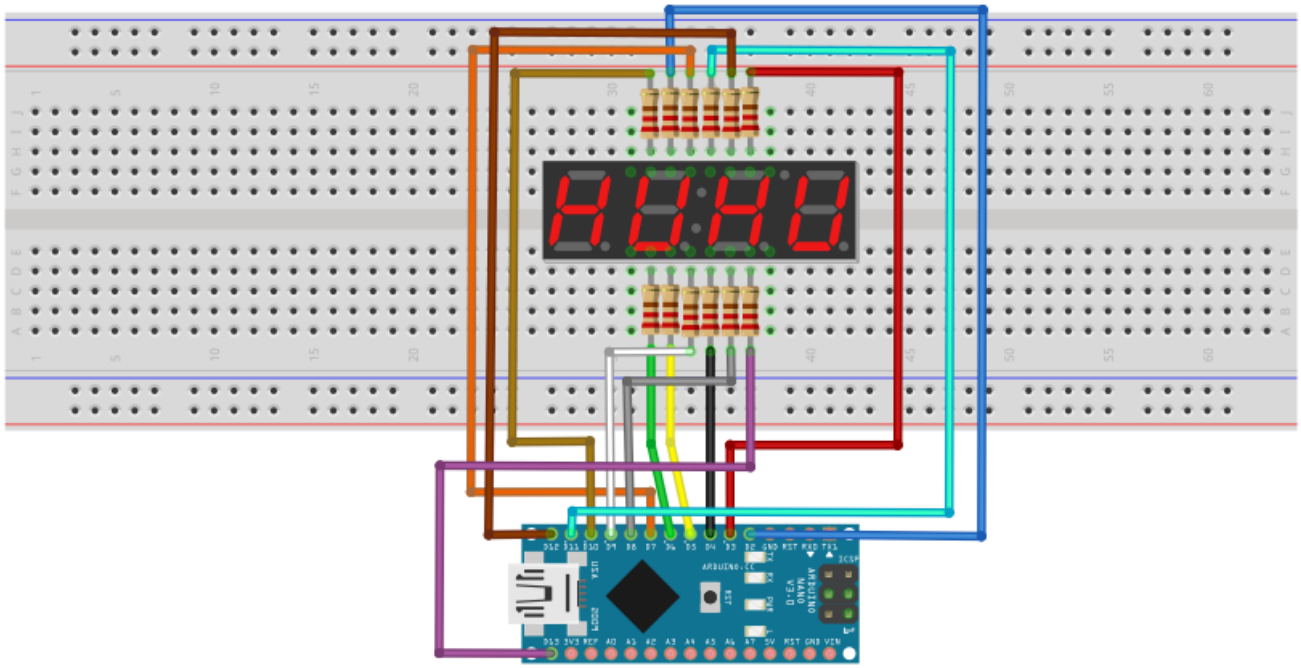
The aim is to display “1234” four characters via dynamically scanning 4-Digit 7-Segment Display.

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  LED A   2       // define Arduino GPIO1 for led a
#define  LED B   3       // define Arduino GPIO2 for led b
#define  LED C   4       // define Arduino GPIO3 for led c
#define  LED D   5       // define Arduino GPIO4 for led d
#define  LED E   6       // define Arduino GPIO5 for led e
#define  LED F   7       // define Arduino GPIO6 for led f
#define  LED G   8       // define Arduino GPIO7 for led g
#define  LED H   9       // define Arduino GPIO8 for led h
#define  LED D1  10
#define  LED D2  11
#define  LED D3  12
#define  LED_D4  13

#define  COM PORT 2  /1:common negative port2: common positive port
char LED PIN[8] = { LED_A , LED_B , LED_C , LED_D , LED_E , LED_F ,
LED G , LED H } ;
char LED_SELECT[4] = {LED_D1 ,LED_D2 , LED_D3 , LED_D4 } ;

char disp[4] = { 0 , 0 , 0 , 0 } ;          // display array value
int display_dat = 59, count = 0 ;

char LED_Display_Value[][3]  =
{ // vaul negative positive
    { '0', 0x3F , 0xC0 } ,
    { '1', 0x06 , 0xF9 } ,
    { '2', 0x5B , 0xA4 } ,
    { '3', 0x4F , 0xB0 } ,
    { '4', 0x66 , 0x99 } ,
    { '5', 0x6D , 0x92 } ,
    { '6', 0x7D , 0x82 } ,
    { '7', 0x07 , 0xF8 } ,
    { '8', 0x7F , 0x80 } ,
    { '9', 0x6F , 0x90 } ,
    {  0 , 0x00 , 0xFF }
};
void  DisplayOff(void)
{
    int i ;
    for( i = 0 ; i < 8 ; i++)
    digitalWrite(LED_PIN[i],LOW);
    for( i = 0 ; i < 4 ; i++)
    digitalWrite(LED_SELECT[i],LOW);
}
```

```
char  GetDisplayValue(char Array[][3] , char DisplayChar )
{
    int i = 0 ;
    for( i = 0 ; i < 10 ; i++)
    {
        if( Array[i][0] == DisplayChar )
            return Array[i][COM_PORT] ;    //return the common
positive port value
    }
    return 0 ;
}

void LED_Display ( char ch)
{
    int i ;
    for( i = 0 ; i < 8 ; i++)
    {
        if(  ch & ( 1 << i ) )
        {
            digitalWrite(LED_PIN[i] , HIGH);
        }else{
            digitalWrite(LED_PIN[i],LOW);
       }
    }
}

void numble2dis( int numble )
{
    int numble bit = 0 ;
    int bit base = 1000 ;
    for( numble_bit = 0 ; numble_bit < 4 ; numble_bit++ )
    {
        if( numble/bit_base != 0)
        {
            disp[numble_bit] = numble/bit_base + '0'  ;// int date
convert to ASCII
            numble = numble%bit_base ;
        }else
        {
            disp[numble_bit] = '0';                  // led display
none
        }
        bit_base = bit_base / 10 ;
    }
}
```

```
void setup()
{
    int i;
    for( i = 0 ; i < 8 ; i++ )
    pinMode( LED PIN[i] ,OUTPUT ) ;          // set all led diplay
array pin output mode
    for( i = 0 ; i < 4 ; i++ )
    pinMode( LED_SELECT[i] ,OUTPUT ) ;       // set led select output
mode
    DisplayOff();
    number2dis(59);
}
void loop()
{
    int i ;
    if( count++ > 50 )
    {
        display_dat-- ;
        number2dis(display_dat);    // integer convert to ASCII value
        count = 0 ;
    }
    for( i = 0  ; i < 4 ; i++ )
    {
        LED_Display( GetDisplayValue(LED_Display_Value,disp[i]) ) ;
        digitalWrite(LED_SELECT[i] ,HIGH );
        delay(5);
        digitalWrite(LED_SELECT[i] ,LOW );
    }
    if(display_dat == 0 )
    while(1);
}
```

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