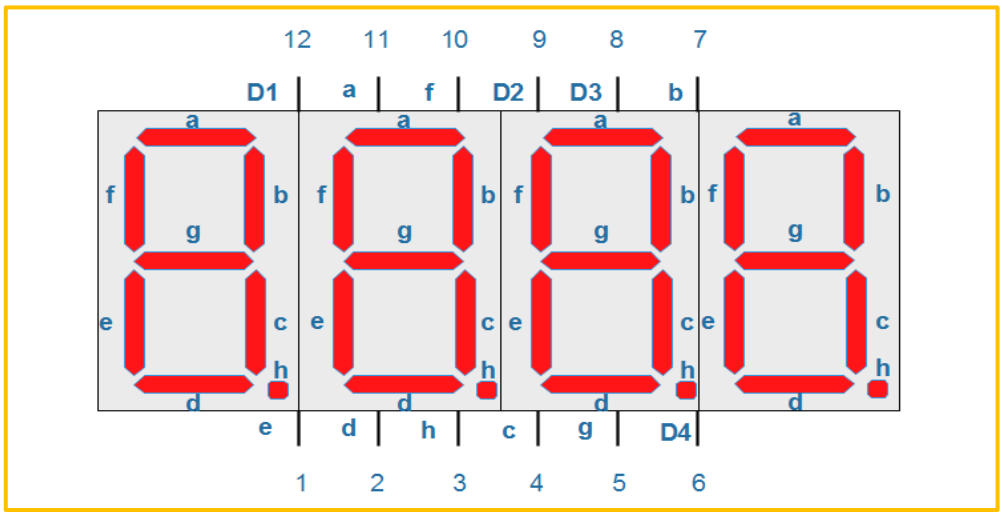
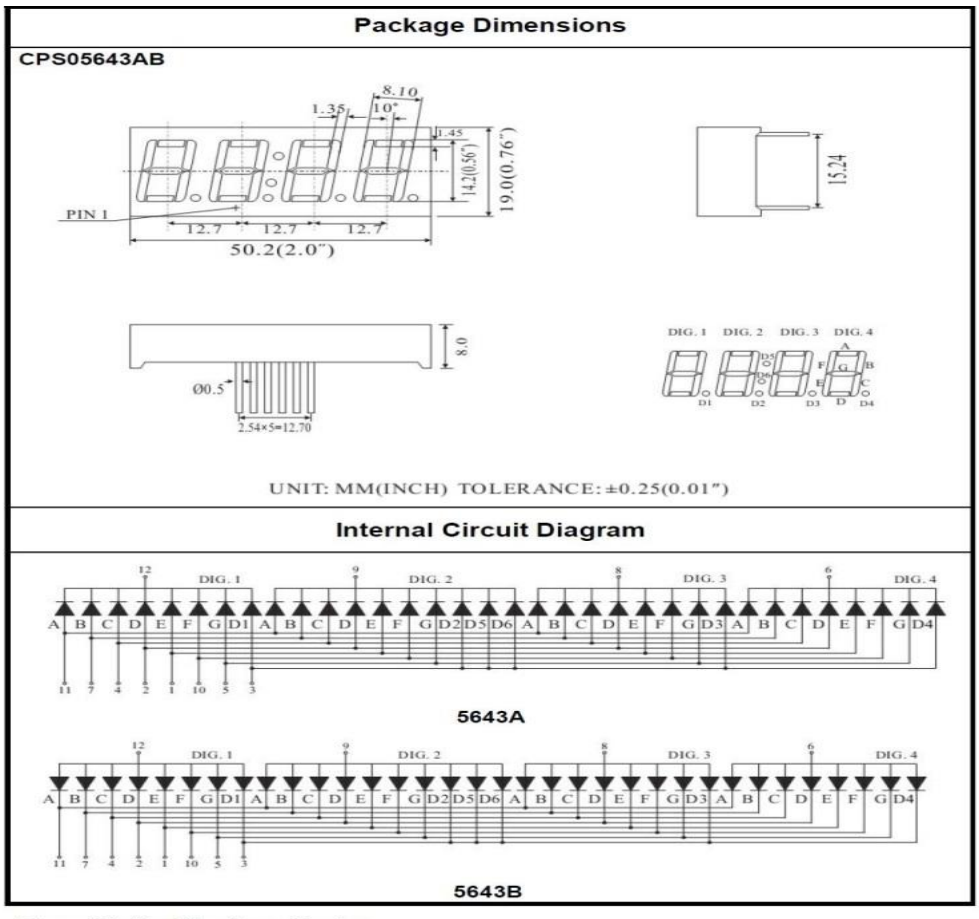
4 bit 7 segment digital tube display experiment

Introduction of 4 bit 7 segment digital tube

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 4 digital tube.





Four Digits Displays Series

4 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 the program is how to dynamically scan the four digital tubes. In fact, through the previous experiment with single digital display, the display of four digital display tube is quite easy. Since it is attributed to the common anode tube, first, we set D1, D2, D3, D4 to low level and all the leds turned out, then we output the truth table of "adbcdefg" to the corresponding gpio port, select the corresponding bit pin and scan continuously. How to achieve 1 minute countdown? We delay by about 1s, subtract 1 from the countdown time, and keep refreshing the display time.

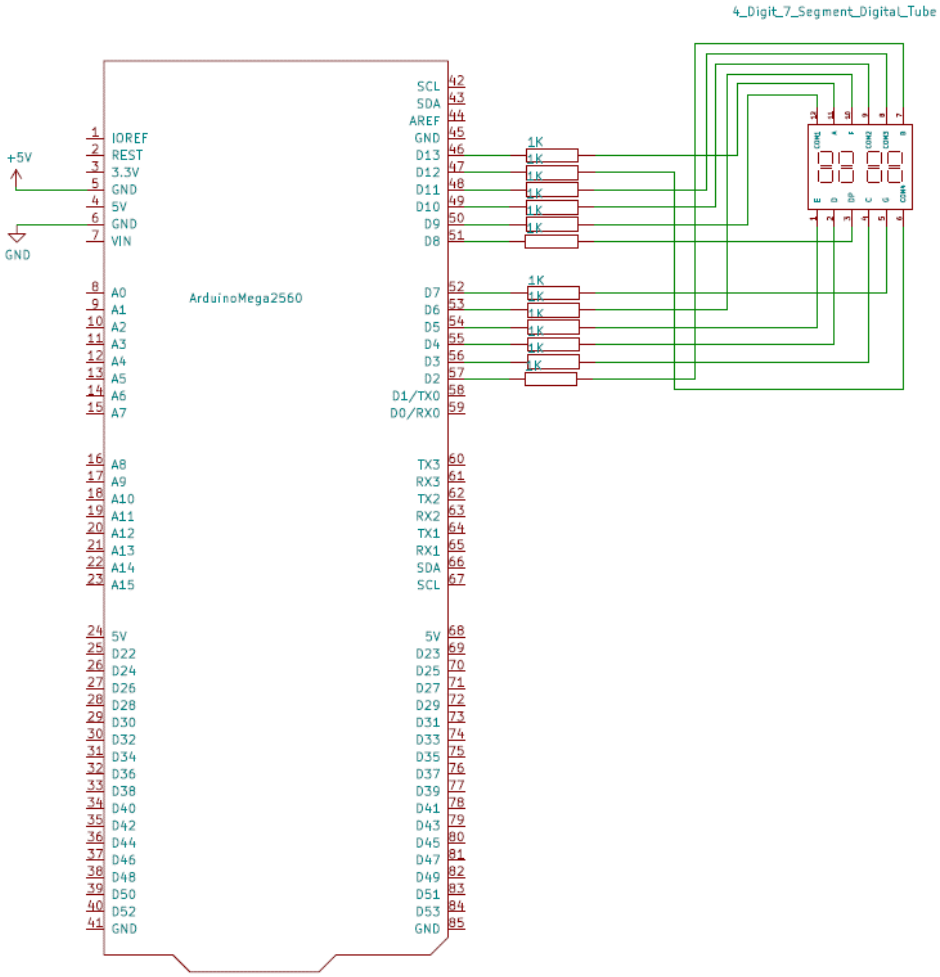
Experiment Purpose

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

Component List

- Keywish Arduino Mega 2560 Mainboard
- Breadboard
- USB cable
- 4-Digit 7-Segment Display * 1
- 1k Resistor * 14
- Several jumper wires

Schematic Diagram



Wiring of Circuit

arduino Mega 2560 控制板	数码管
6	(f)
2	(b)
3	(c)
4	(d)
5	(e)
13	(a)
7	(g)
8	(h)
9	(d1)
10	(d2)
11	(d3)
12	(d4)

Code

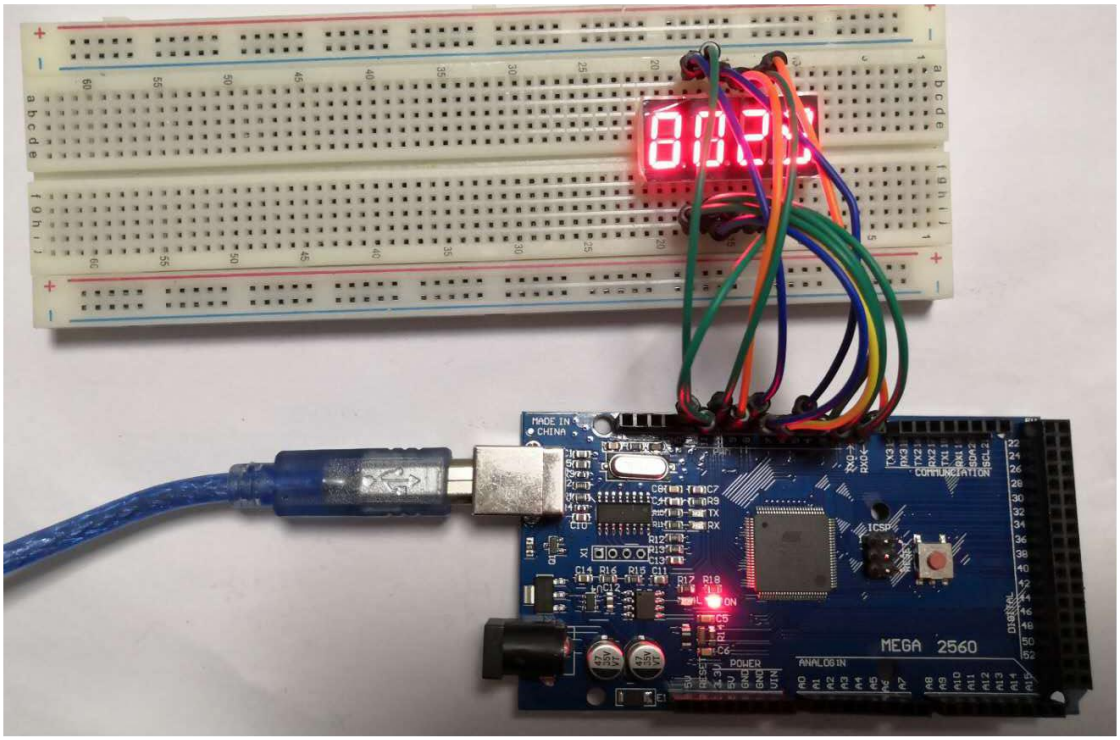
```
#include "SegmentDisplay.h"
#define  LED_A   13      // define Arduino GPIO1 for led a
#define  LED_B   2      // define Arduino GPIO2 for led b
#define  LED_C   3      // define Arduino GPIO3 for led c
#define  LED_D   4      // define Arduino GPIO4 for led d
#define  LED_E   5      // define Arduino GPIO5 for led e
#define  LED_F   6      // define Arduino GPIO6 for led f
#define  LED_G   7      // define Arduino GPIO7 for led g
#define  LED_H   8      // define Arduino GPIO8 for led h
#define  LED_D1  9
#define  LED_D2  10
#define  LED_D3  11
#define  LED_D4  12
SegmentDisplay _4Bit_7SegmentDisplay(LED_A, LED_B, LED_C, LED_D, LED_E, LED_F, LED_G,
LED_H, LED_D1, LED_D2, LED_D3, LED_D4);
  int ShowTime = 60, count = 0;
void setup()
{
    Serial.begin(9600);
    _4Bit_7SegmentDisplay.TurnOffAllLed();
}

void loop()
{
    if (count++ > 50 )
    {
        ShowTime-- ;
        count = 0 ;
        Serial.println(ShowTime);
    }
    _4Bit_7SegmentDisplay.DisplayChar((int)ShowTime);
    delay(5);
    if (ShowTime == 0) {
        _4Bit_7SegmentDisplay.TurnOffAllLed();
        while(1);
    }
}
```

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

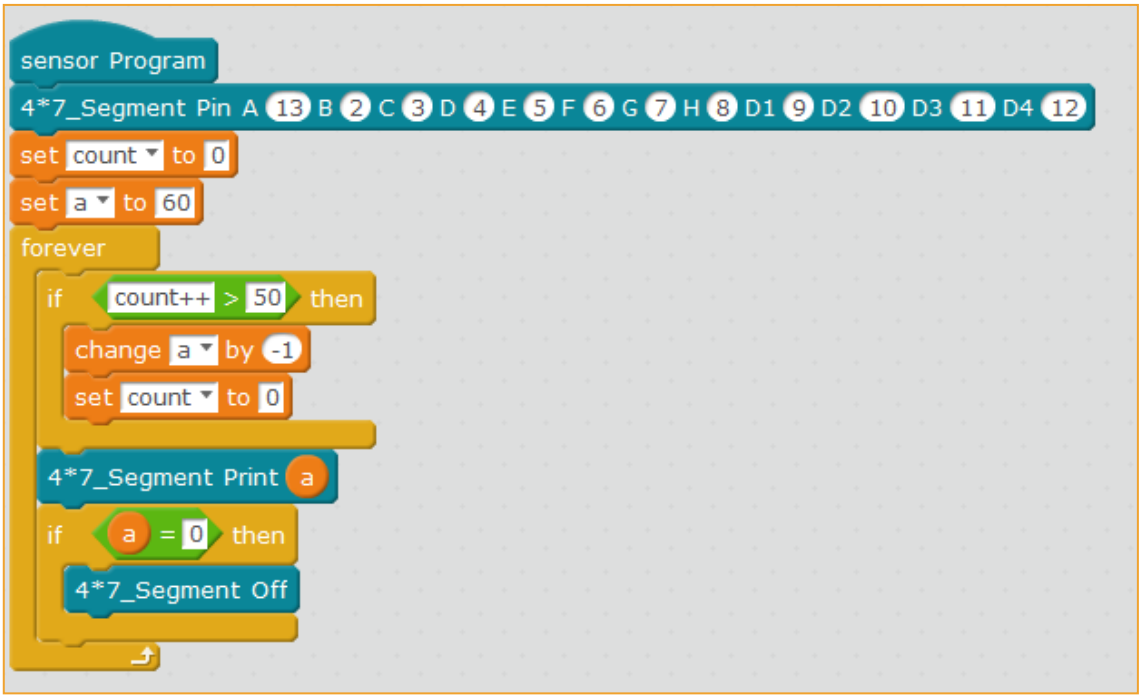
Loop	numble	bit_base	disp
1	1234	1000	1
2	234	100	2
3	34	10	3
4	4	1	4

Experiment Result

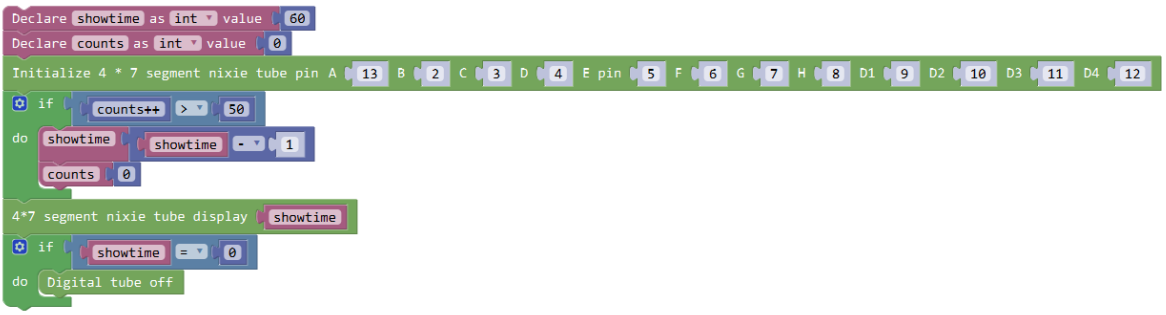


Mblock programming program

Mblock writtes 4-Digit 7-Segment Display program as shown in the figure below:



Mixly programming program



MagicBlcok programming program