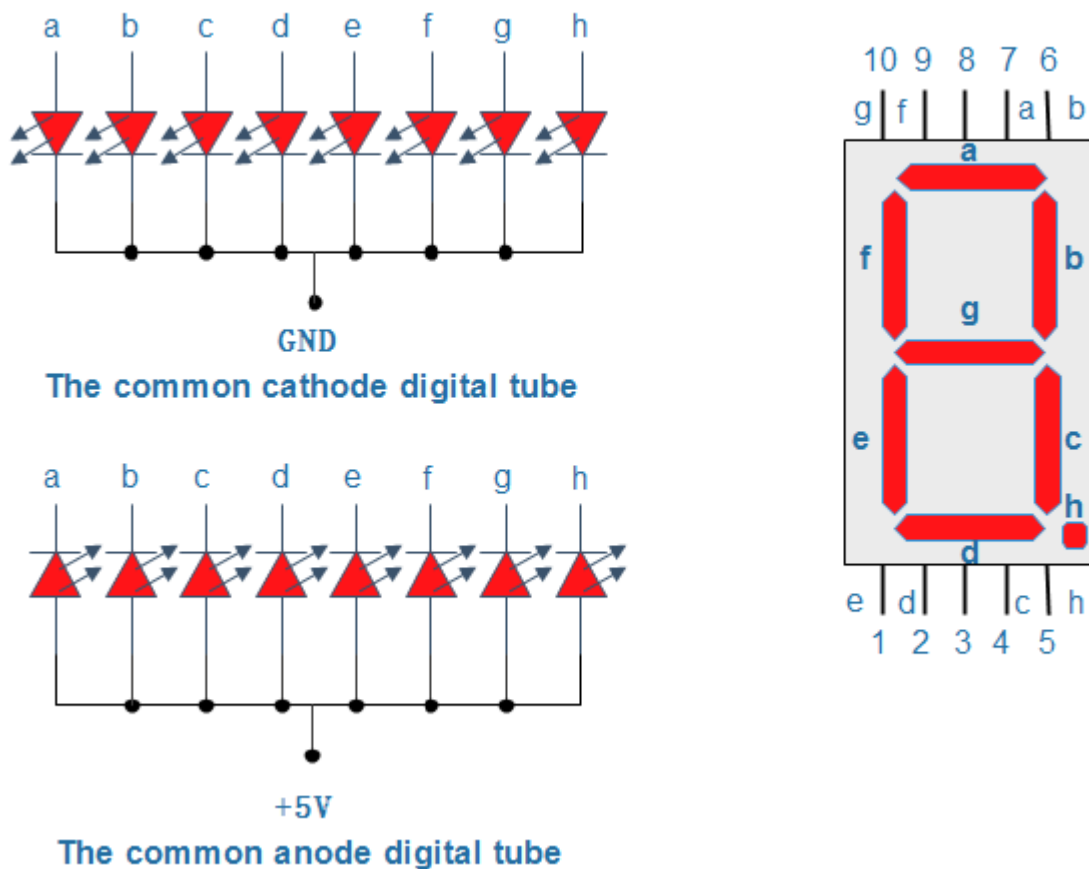


7-Segment Display

Introduction

Nixie tube is a semiconductor light emitting element, and its basic unit is a light-emitting diode. According to the number of segment, Nixie tube is divided into seven-segment tube and eight-segment tube. Eight-segment tube has one more light-emitting diode unit (one more a decimal point) segment than seven-segment tube. We will use eight-segment tube in this experiment. Light-emitting diode unit can be divided into common anode digital tube and common cathode digital tube according to the connection mode. The common anode digital tube refers to all anodes of a light-emitting linking to +5 v. When the cathode of any one segment of light emitting diode is low level, the corresponding segment will light up; when the cathode is high level, the segment stays unlighted. The common cathode digital tube refers to all cathodes of a light-emitting linking to GND. When the anode of any one segment of light emitting diode is high level, the corresponding segment will light up, when the anode is low level, the segment stays unlighted as well.



Each segment of a Nixie tube is composed of light emitting diode, so to display different numbers, the principle is that the corresponding LEDs are lit up. Say we want to display

Number 0, that means “abcdef” is lightened and the other are turned out, so that we only need to see the corresponding the truth table of displayed number.

Light emitting diode of Nixie tube requires to connect current-limiting resistor, or light emitting diode will be destroyed due to excess of electricity. We will use common cathode digital tube in the experiment.

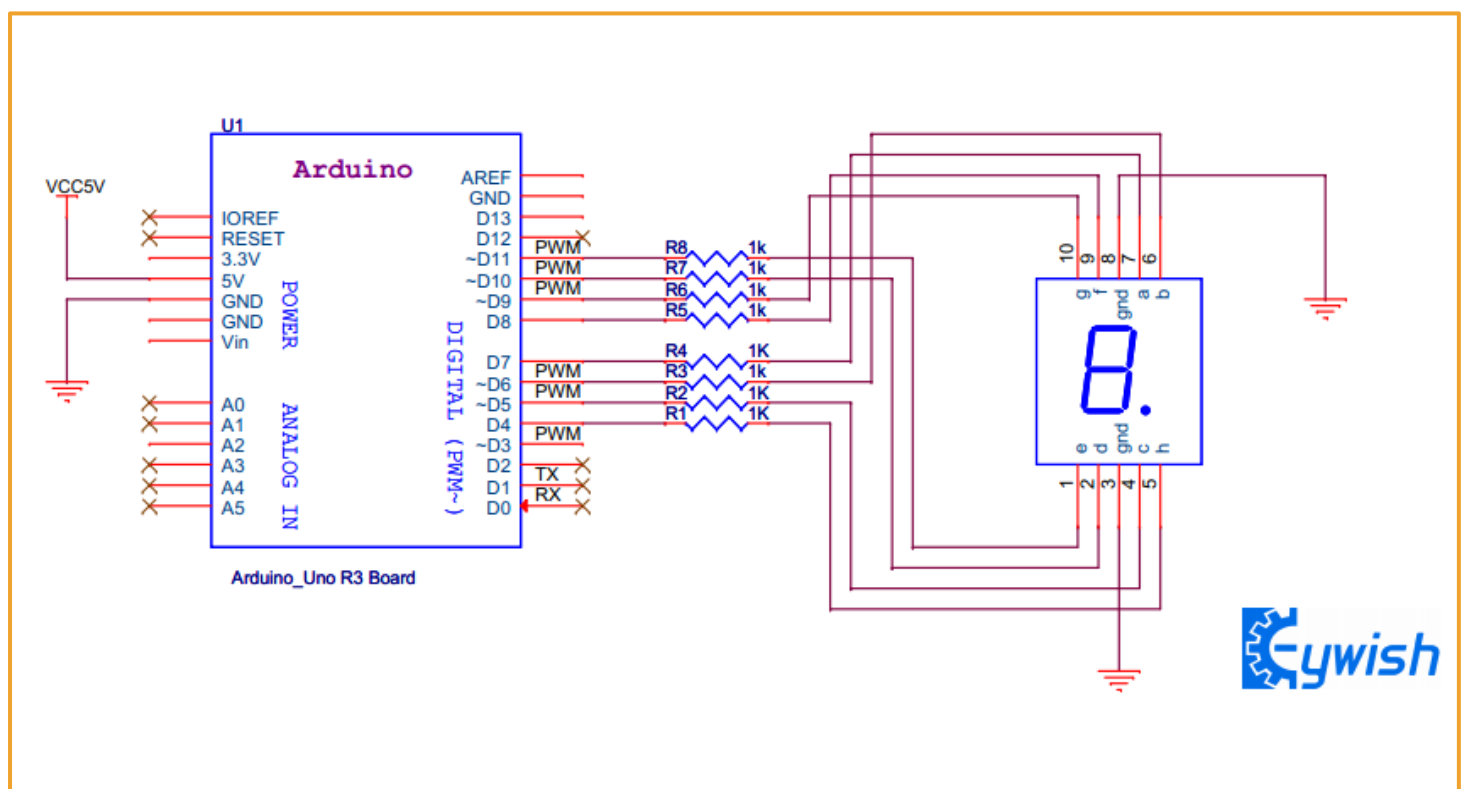
Experiment Purpose

Display Numbers 0-9 on eight-segment tube.

Component List

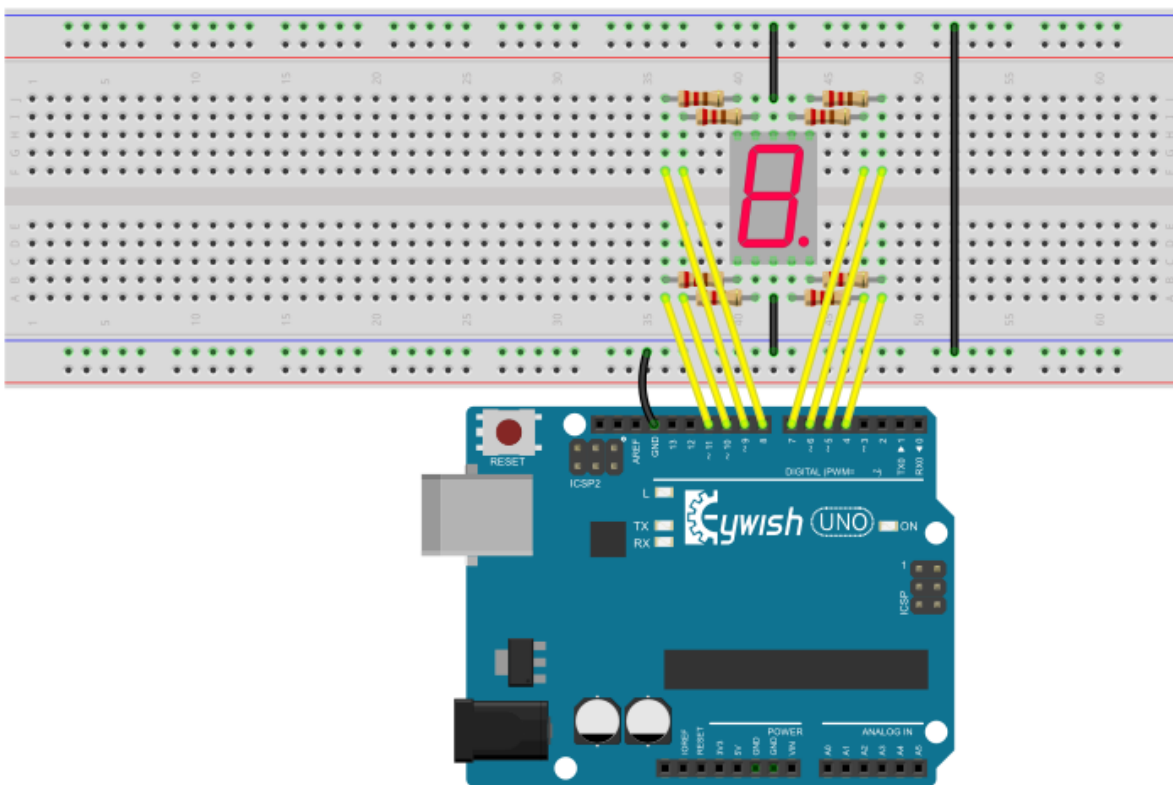
- ◆ Keywish Arduino Uno R3 mainboard
- ◆ Breadboard
- ◆ USB cable
- ◆ 7-segment nixie tube*1
- ◆ 1k Ω Resistor*8
- ◆ Several breadboard jumpers

Schematic Diagram



Wiring of Circuit

Arduino Uno R3	7 Segment display
7	7 (a)
6	6 (b)
5	4 (c)
10	2 (d)
11	1 (e)
8	9 (f)
9	10 (g)
4	5 (h)
GND	8 (gnd)
GND	3 (gnd)



Nixie tube has seven segments for displaying numbers and one for decimal point. When we want to display numbers on the tube, as long as we lighten the corresponding segments. For example, if we want to display Number 1, then “b” and “c” are lightened. We write a subroutine for each number, and make Nixie tube loop 1 ~ 8 every 2 seconds in the main program

Code

```
#define LED_A 7 // define Arduino GPIO7 for led a
#define LED_B 6 // define Arduino GPIO6 for led b
#define LED_C 5 // define Arduino GPIO5 for led c
#define LED_D 10 // define Arduino GPIO11 for led d
#define LED_E 11 // define Arduino GPIO10 for led e
#define LED_F 8 // define Arduino GPIO8 for led f
#define LED_G 9 // define Arduino GPIO9 for led g
#define LED_H 4 // define Arduino GPIO4 for led h
char value , dispaly_char ;
char LED_PIN[8] = { LED_A , LED_B , LED_C , LED_D , LED_E , LED_F ,
LED_G , LED_H } ;
char LED_Display_Value[][2] =
{
    { '0', 0x3F } ,
    { '1', 0x06 } ,
    { '2', 0x5B } ,
    { '3', 0x4F } ,
    { '4', 0x66 } ,
    { '5', 0x6D } ,
    { '6', 0x7D } ,
    { '7', 0x07 } ,
    { '8', 0x7F } ,
    { '9', 0x6F } ,
    { 0 , 0x00 }
};
void DisplayOff(void)
{
    int i ;
    for( i = 0 ; i < 8 ; i++)
        digitalWrite(LED_PIN[i],LOW);
}
char GetDisplayValue(char Array[][2] , char DisplayChar )
{
    int i = 0 ;
    for( i = 0 ; i < 10 ; i++)
    {
        if( Array[i][0] == DisplayChar )
            return Array[i][1] ;
    }
    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 setup()
{
    int i;
    Serial.begin(115200);
    for( i = 0 ; i < 8 ; i++ )
        pinMode( LED_PIN[i] ,OUTPUT ) ;// set all led diplay array pin
output mode
    DisplayOff();
}

void loop()
{
    Serial.println("please input display char \n");
    value = Serial.read() ;
    dispaly_char = GetDisplayValue( LED_Display_Value , value ) ;
    Serial.print(value);
    Serial.print(dispaly_char);
    if ( dispaly_char != 0 )
    {
        DisplayOff();
        LED_Display( dispaly_char );
    }
}
```