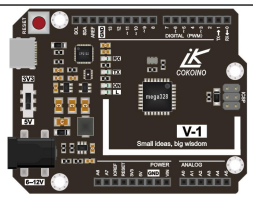
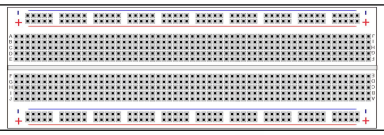
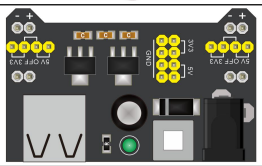
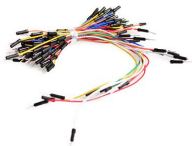
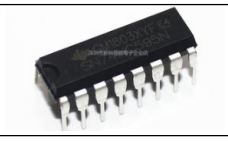
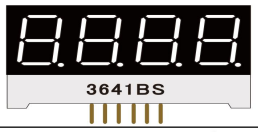
18.Control the value of the display

ABOUT THIS PROJECT：

You will learn：

- Use two buttons to control the value of the four digit 7 segment display

1、Things used in this project:

Hardware components	Picture	Quantity
V-1 board		1 PCS
Breadboard		1 PCS
9V Battery Snap Connector		1 PCS
Breadboard power module		1 PCS
Male to Male DuPont Line		33 PCS
Type C USB Cable		1 PCS
74HC595 Shift Register		2 PCS
Four Digit 7 Segment Display		1 PCS
Button		2 PCS
Red, green, blue, yellow button cap		2 PCS

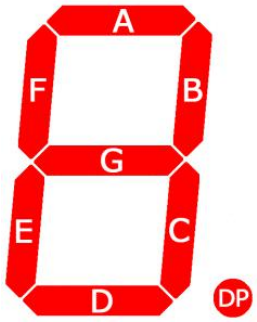
2. Four digit 7 segment display

In this tutorial I'm going to show you how to set up and program single digit and multi-digit seven segment displays on an V-1 board.

Seven segment displays come in a wide variety of sizes and colors. Red, blue, and green are the easiest colors to find. Sizes range from small 0.56 inch displays up to large 4 inch and even 6.5 inch displays. Some displays have a single digit, and others have two or four.

HOW 7-SEGMENT DISPLAYS WORK

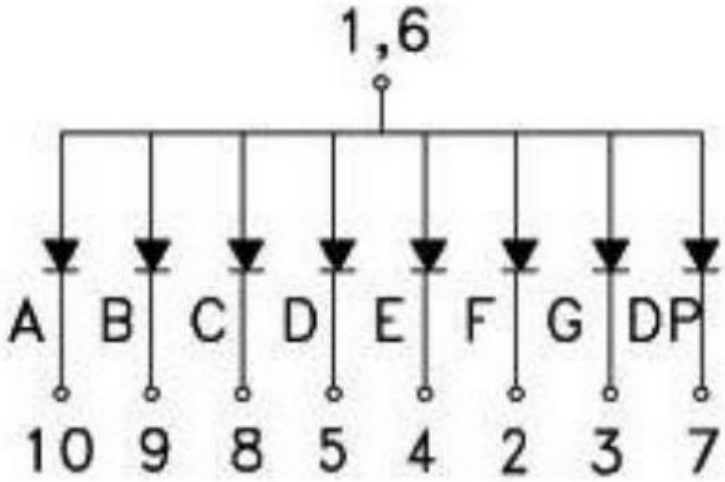
Seven segment displays consist of 7 LEDs, called segments, arranged in the shape of an "8". Most 7-segment displays actually have 8 segments, with a dot on the right side of the digit that serves as a decimal point. Each segment is named with a letter A to G, and DP for the decimal point:

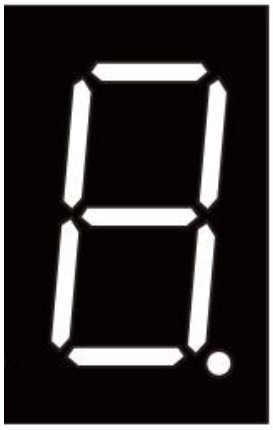


Each segment on the display can be controlled individually, just like a regular LED.
There are two types of 7-segment displays – common cathode and common anode.

COMMON CATHODE DISPLAYS

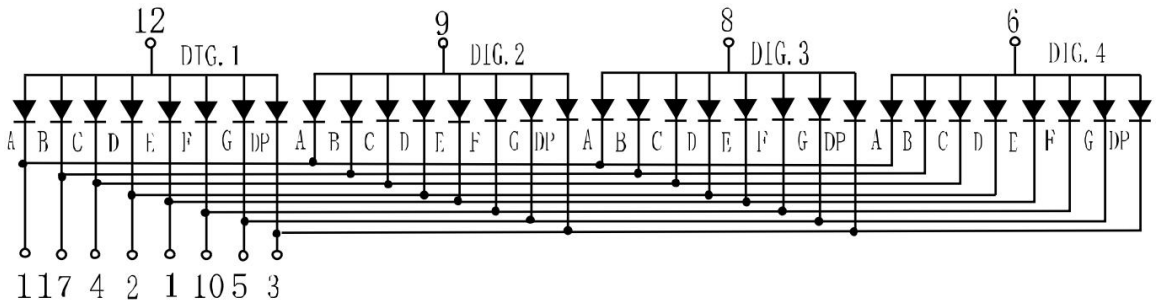
In common anode displays, all of the anode are connected to power and individual segments are turned on and off by switching power to the cathodes:

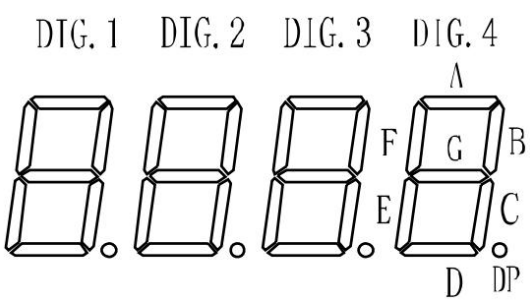


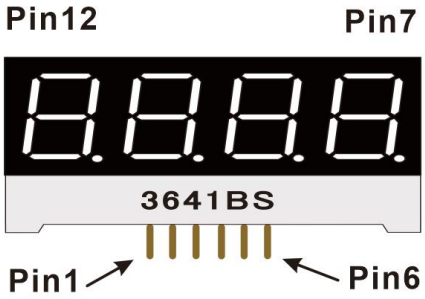


The above is a 1-digit display. Below we explain the 4-digit common anode display used in this experiment. It is composed of four 1-digit common anode displays.

The 4-digit common anode display connects the cathodes of four 1-digit common anode diodes together in the same display segment, and the anodes are separate, as shown below:

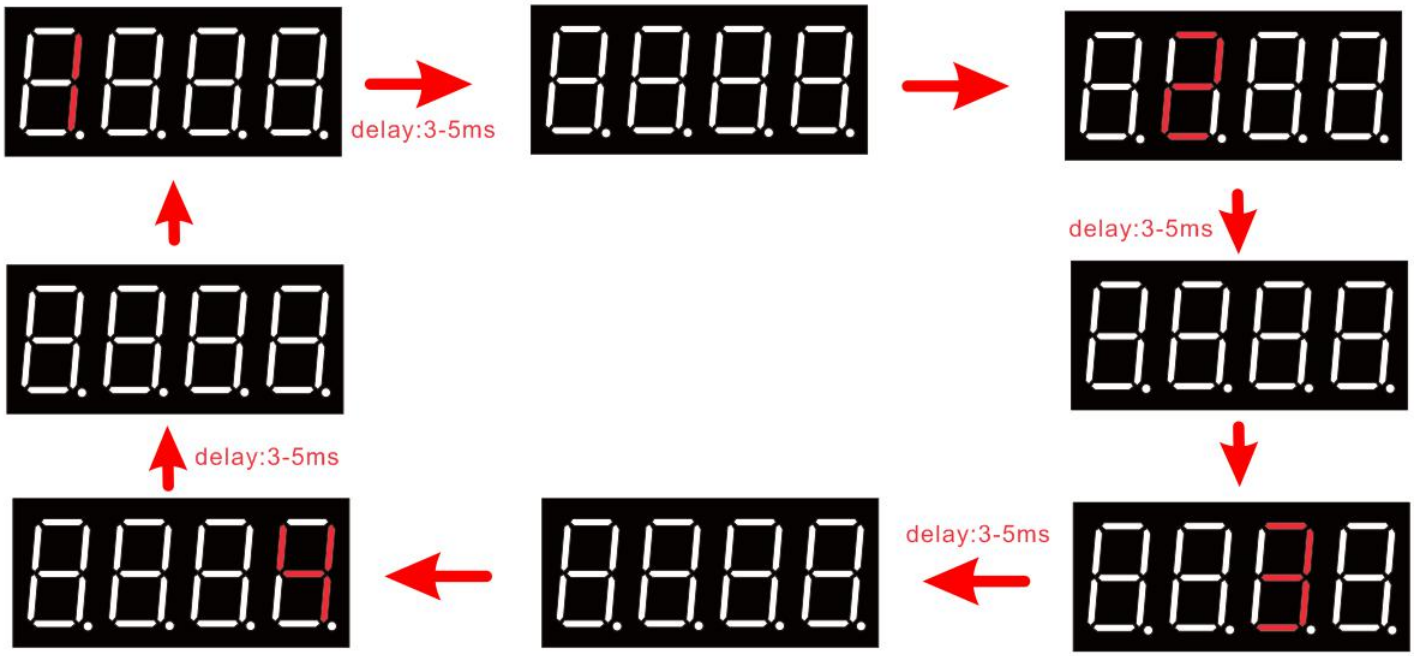






2.1 Dynamic display of the 4-digit display

Here we use 1 to indicate a high level, 0 to indicate a low level, and xxxxxxxx to represent a single display: anode \A\B\C\D\E\F\G\DP. For example, the 4-bit display shows 1234, loop: 1 bit 110011111---> delay 3 milliseconds--->1 bit 011111111--->2 digits 100100101---> delay 3 milliseconds--->2 digits 011111111 --->3 digits 10000111--->delay 3ms--->3 digits 011111111--->4 digits 110011001---> Delay 3ms--->2 digits 011111111, because it has been cycled in sequence Display, and the frequency is greater than 50 Hz, so the human eye can not see the number of the display is jittery, the number seems to be static.



3. Make A Digital display

You will control the dynamic display of the 4-digit display through the digital port of V-1 board. The software part mainly learns the timing interrupt, the millis() acquisition system running function, and the attachInterrupt() external interrupt function.

For details on the millis() function, please refer to:
https://www.arduino.cc/reference/en/language/functions/time/millis/

MsTimer2 library download and timer interrupt details please refer to:
https://playground.arduino.cc/Main/MsTimer2/

For details of external interrupts, please refer to:
https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/

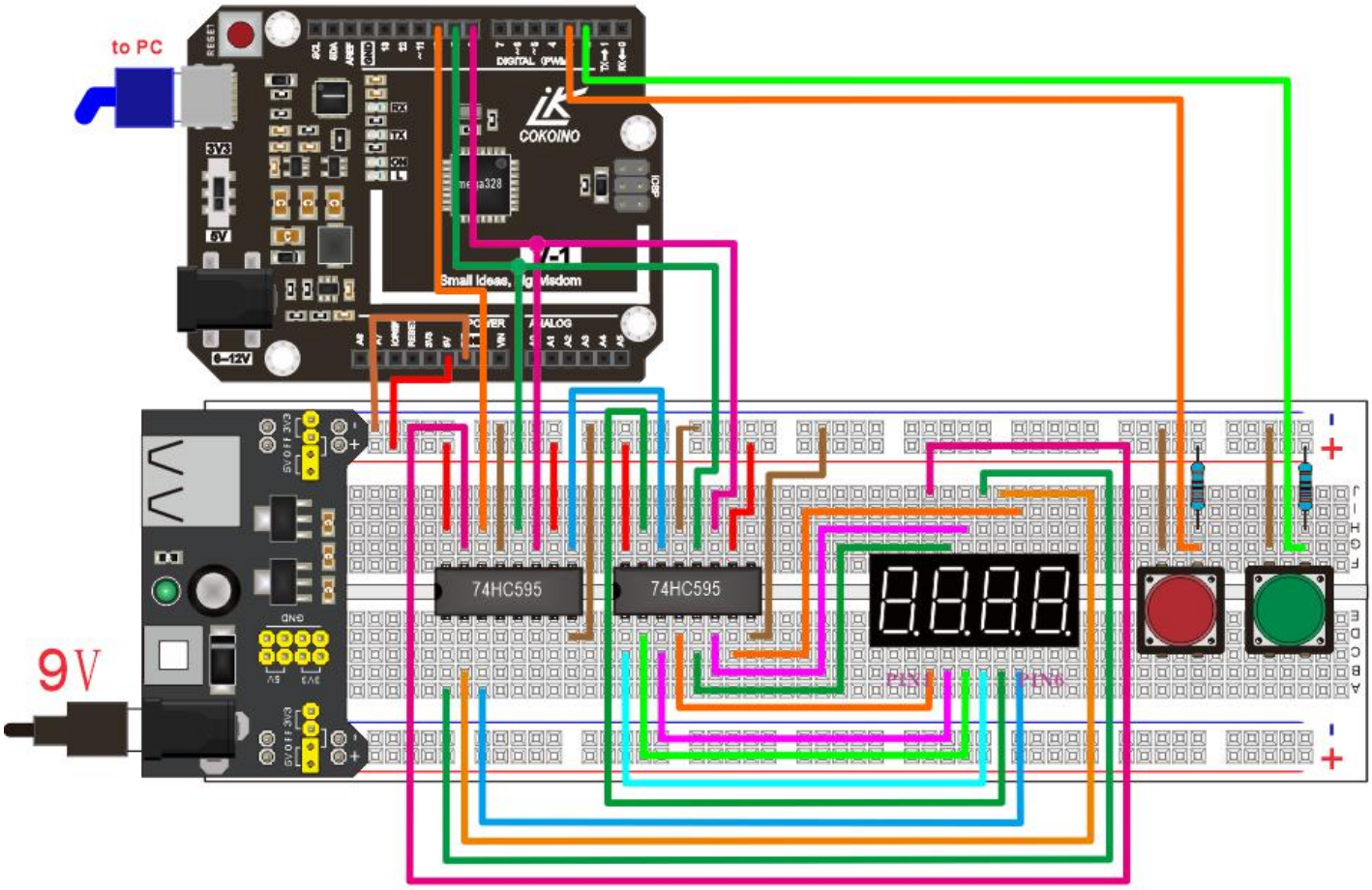
CONNECTING Four DIGIT DISPLAY TO THE ARDUINO

3.1、Sketch

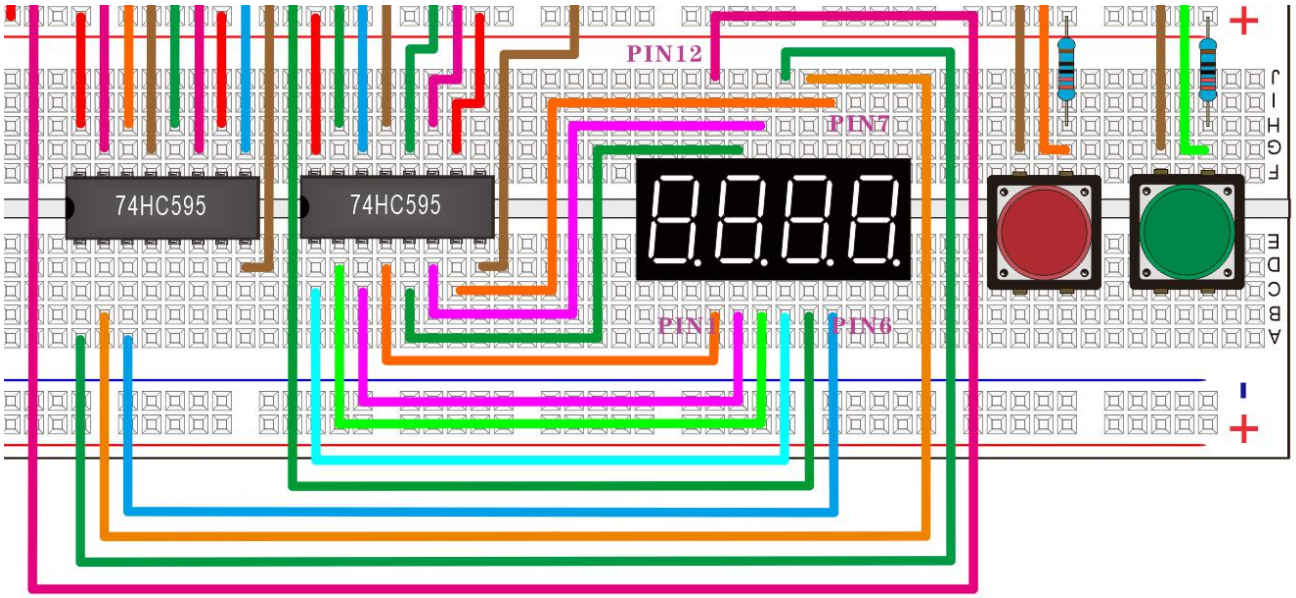
```
#include <MsTimer2.h>
int clockPin = 8;    //SH_CP
int latchPin = 9;    //ST_CP
int dataPin = 10;    //DS
int bit1 = 4;
int bit2 = 5;
int bit3 = 6;
int bit4 = 7;
int key1 = 2;
int key2 = 3;
int count = 0;
int newtime=0;
int oldtime=0;
int dat_bit[4]={0x08,0x04,0x02,0x01};
int dat[10]={0x05,0x7d,0x46,0x54,0x3c,0x94,0x84,0x5d,0x04,0x14};
////////////////////////////////////////////////////////////
void digital_clear(void){                                  //
  digitalWrite(latchPin,LOW);
  shiftOut(dataPin, clockPin, MSBFIRST ,0xff);
  shiftOut(dataPin, clockPin, MSBFIRST ,0x00);
  digitalWrite(latchPin,HIGH);
}
////////////////////////////////////////////////////////////
void shift_bit(int D,int B){
    digitalWrite(latchPin,LOW);
    shiftOut(dataPin, clockPin, MSBFIRST ,dat[D]);
    shiftOut(dataPin, clockPin, MSBFIRST ,dat_bit[B]);
    digitalWrite(latchPin,HIGH);
}
////////////////////////////////////////////////////////////
void digital_display(void){
    digitalWrite(bit1,HIGH);
    shift_bit(count%10,0);
    delay(3);
    digital_clear();
```

```
    digitalWrite(bit1,LOW);

    digitalWrite(bit2,HIGH);
    shift_bit(count%100/10,1);
    delay(3);
    digital_clear();
    digitalWrite(bit2,LOW);

    digitalWrite(bit3,HIGH);
    shift_bit(count%1000/100,2);
    delay(3);
    digital_clear();
    digitalWrite(bit3,LOW);

    digitalWrite(bit4,HIGH);
    shift_bit(count/1000,3);
    delay(3);
    digital_clear();
    digitalWrite(bit4,LOW);
}
//////////////////////////////////////////////////////////////
void count1(void){
  newtime=millis();
  if(newtime-oldtime>100)
       count++;
  oldtime=newtime;
  if(count>=9999)
       count=9999;
}
//////////////////////////////////////////////////////////////
void count2(void){
  newtime=millis();
  if(newtime-oldtime>100)
       count--;
  oldtime=newtime;
  if(count<=0)
       count=0;
}
//////////////////////////////////////////////////////////////
void setup (){
    pinMode(latchPin,OUTPUT);
    pinMode(clockPin,OUTPUT);
    pinMode(dataPin,OUTPUT);
    pinMode(bit1,OUTPUT);
    pinMode(bit2,OUTPUT);
    pinMode(bit3,OUTPUT);
    pinMode(bit4,OUTPUT);
    pinMode(key1,INPUT);
    pinMode(key2,INPUT);
    MsTimer2::set(10, digital_display);
    MsTimer2::start();
    attachInterrupt(0, count1, CHANGE);
    attachInterrupt(1, count2, CHANGE);
    oldtime=millis();
}
//////////////////////////////////////////////////////////////
void loop(){ }
```

3.2、Wiring Diagram

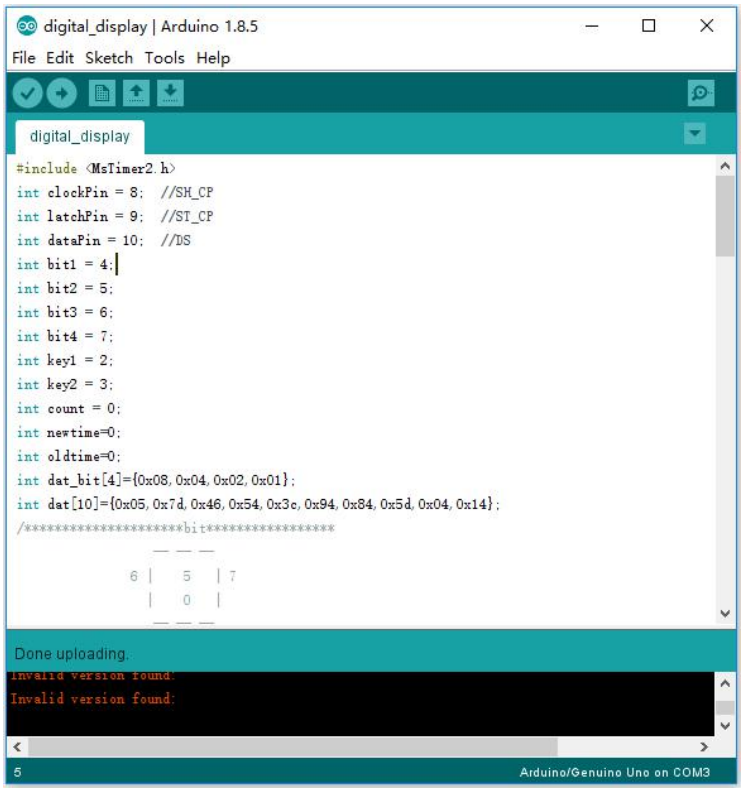


主图

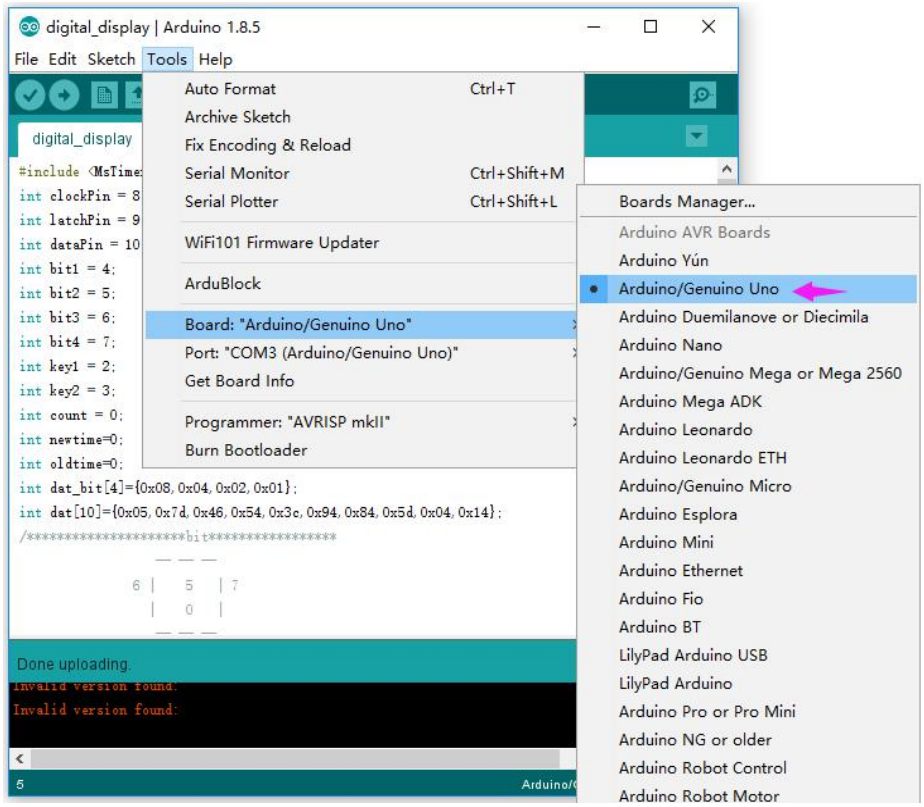


3.3、Steps

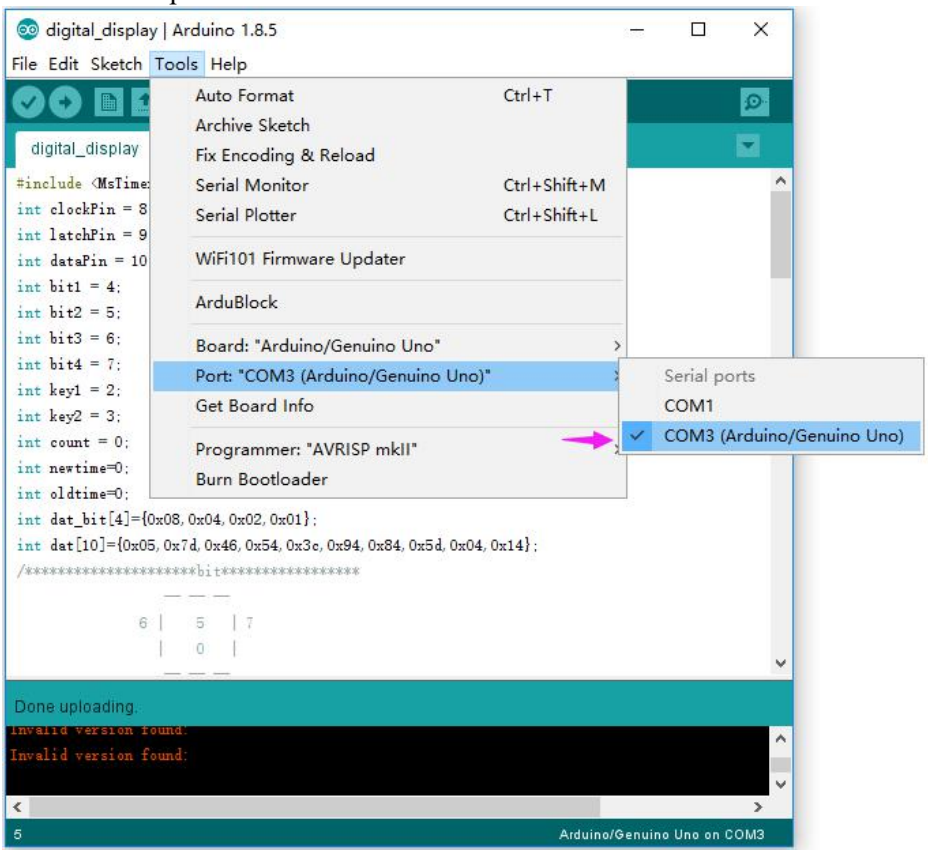
3.3.1、Connect the computer to the V-1 board board with a USB cable and copy the sample code above into the Arduino IDE as shown below:



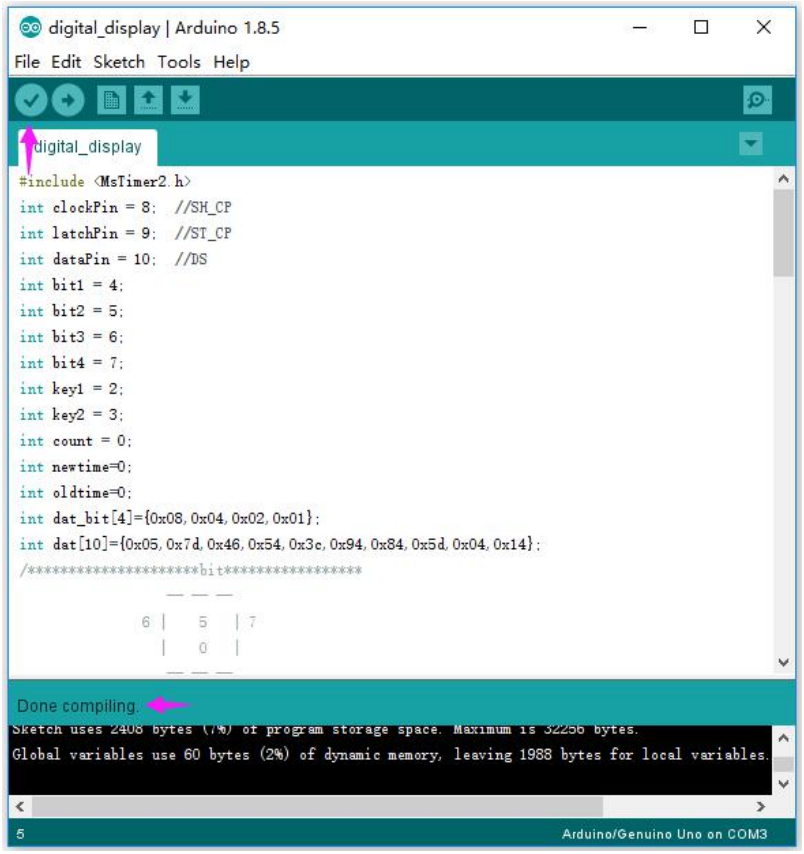
3.3.2、Select the board type



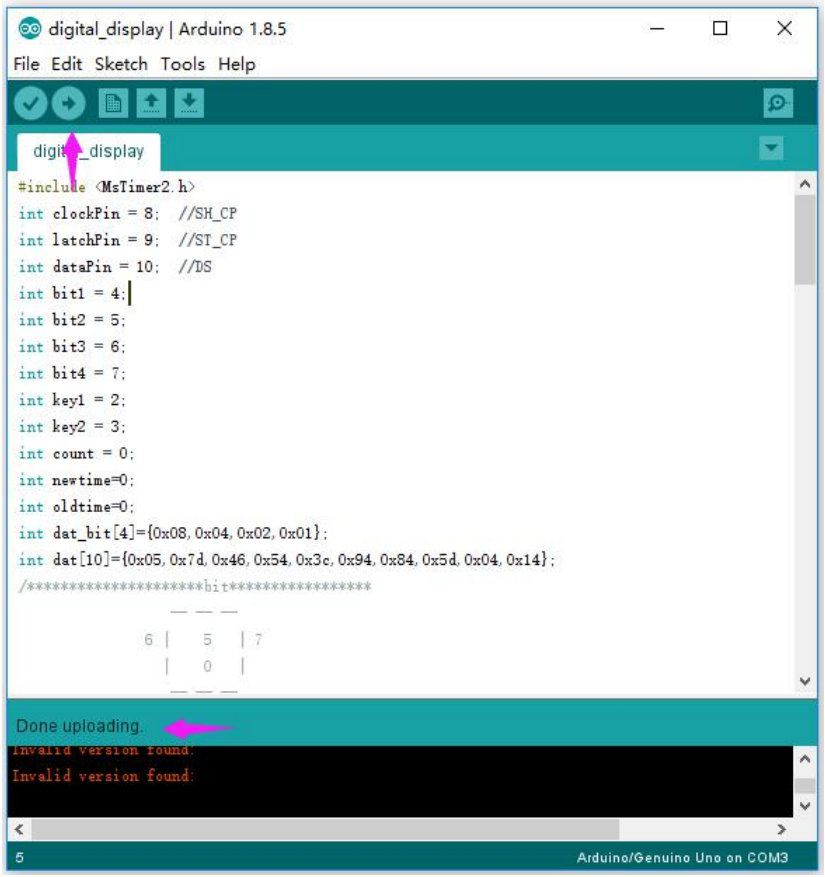
3.3.3、Select port



3.3.4、Compiling



3.3.5、Upload



3.3.6、Unplug the USB cable from the V-1 board, then connect the power module with the external power supply and turn on the power switch. Press the green button d to increase the value of the display. Press the red button to decrease the value.