

# Lesson 8 74HC595 And Segment Display

## Introduction

In this lesson, you will learn how to use the 74HC595 shift register to control the segment display show number from 0-13.

## Hardware Required

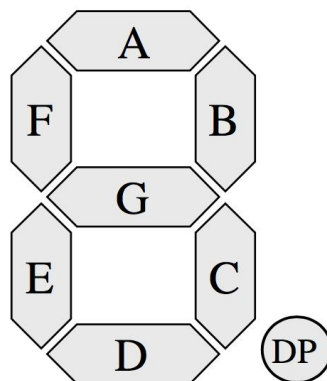
- ✓ 1 \* RexQualis Mega 2560
- ✓ 1 \* Breadboard
- ✓ 8 \* 220 ohm Resistors
- ✓ 1 \* 74hc595 IC
- ✓ 1 \* 1 Digit 7-Segment Display
- ✓ 26 \* M-M Jumper Wires



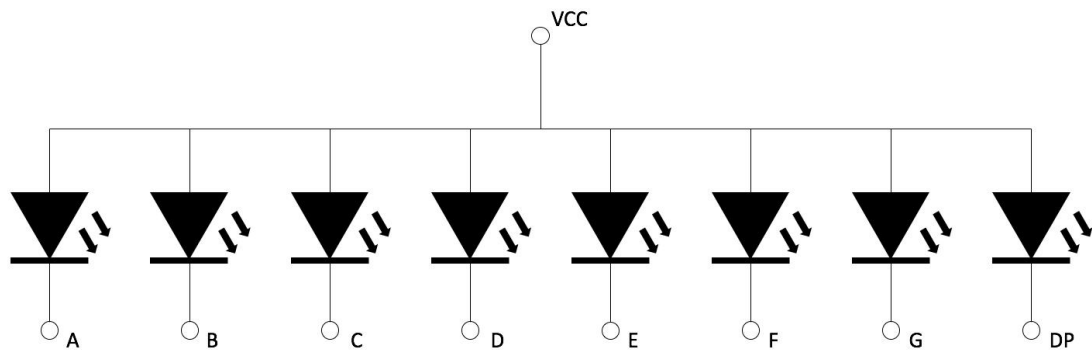
## Principle

### Seven Segment Display

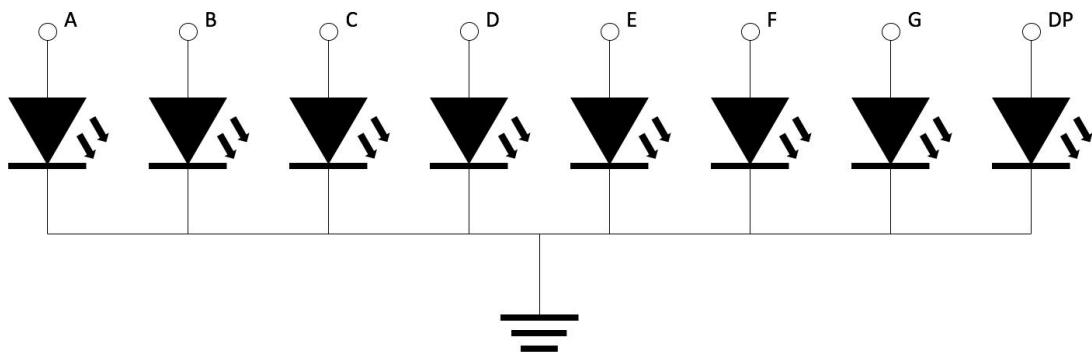
A seven-segment display is a LED module composed of 8 LEDs. 7 of the LEDs are for segments of one digit (shown as A to G below) and the other LED is for the decimal point (shown as DP below).



A common-anode seven-segment display



A common-cathode seven-segment display



For saving the pin number for controlling a seven-segment display, a shift register is used as a serial-to-parallel converter to send signals to the display. That is, we serially send 8 bits of data, which represents the way we want to turn on the display, by one signal pin into the shift register and the register can output the corresponding data pattern to its 8 output pins at once (parallel).

Connect the 7-Segment display and 74HC595 shift register to Arduino hardware:

Connect Vcc pin on 74HC595 to 5V pin on Arduino hardware.

Connect GND and OE pins on 74HC595 to GND pin on Arduino hardware.

Connect DS or SER pin on 74HC595 to digital pin 8 on Arduino hardware.

Connect SHCP or SRCLK pin on 74HC595 to digital pin 10 on Arduino hardware.

Connect STCP or RCLK pin on 74HC595 to digital pin 9 on Arduino hardware.

Connect Q0-Q6 or QA-QG pin on 74HC595 to pin A-G on 7-segment display.

Connect Q7 or QH pin on 74HC595 to pin DP on 7-segment display.

Connect pin 3 and 8 on 7-segment display to GND pin on Arduino hardware. (This example uses the common cathode, if you use the common anode, please connect the 3, 8 pin to Mega 2560 board + 5V)

## Code interpretation

**// define the LED digit patterns, from 0 - 13**

**// 1 = LED on, 0 = LED off, in this order:**

**//74HC595 pin      Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7**

byte seven\_seg\_digits[14] = {

B01111010,    **// = D**

B10011100,    **// = C**

B00111110,    **// = B**

B11101110,    **// = A**

B11100110,    **// = 9**

B11111110,    **// = 8**

B11100000,    **// = 7**

B10111110,    **// = 6**

B10110110,    **// = 5**

B01100110,    **// = 4**

B11110010,    **// = 3**

B11011010,    **// = 2**

```

    B01100000,  // = 1

    B11111100,  // = 0

};

// connect to the ST_CP of 74HC595 (pin 9,latch pin)

int latchPin = 9;

// connect to the SH_CP of 74HC595 (pin 10, clock pin)

int clockPin = 10;

// connect to the DS of 74HC595 (pin 8)

int dataPin = 8;

void setup() {

// Set latchPin, clockPin, dataPin as output

    pinMode(latchPin, OUTPUT);

    pinMode(clockPin, OUTPUT);

    pinMode(dataPin, OUTPUT);

}

// display a number on the digital segment display

void sevenSegWrite(byte digit) {

// set the latchPin to low potential, before sending data

    digitalWrite(latchPin, LOW);

// the original data (bit pattern)

    shiftOut(dataPin, clockPin, LSBFIRST, seven_seg_digits[digit]);

// set the latchPin to high potential, after sending data

```

```

digitalWrite(latchPin, HIGH);

}

void loop() {

// count from 14 to 0

    for (byte digit = 14; digit > 0; --digit) {

        delay(1000);

        sevenSegWrite(digit - 1);

    }

// suspend 4 seconds

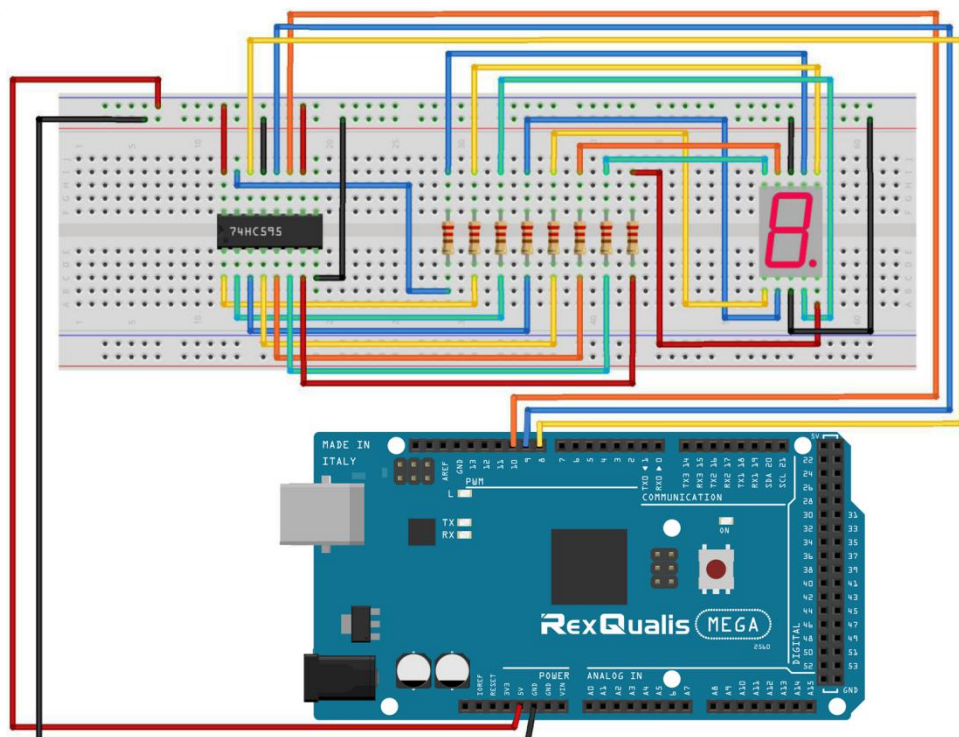
    delay(5000);

}

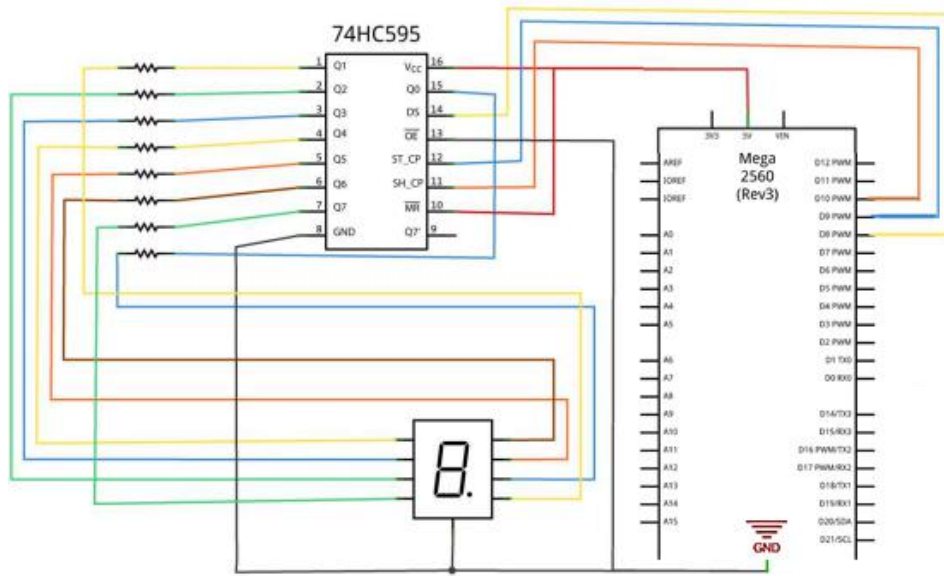
```

## Experimental Procedures

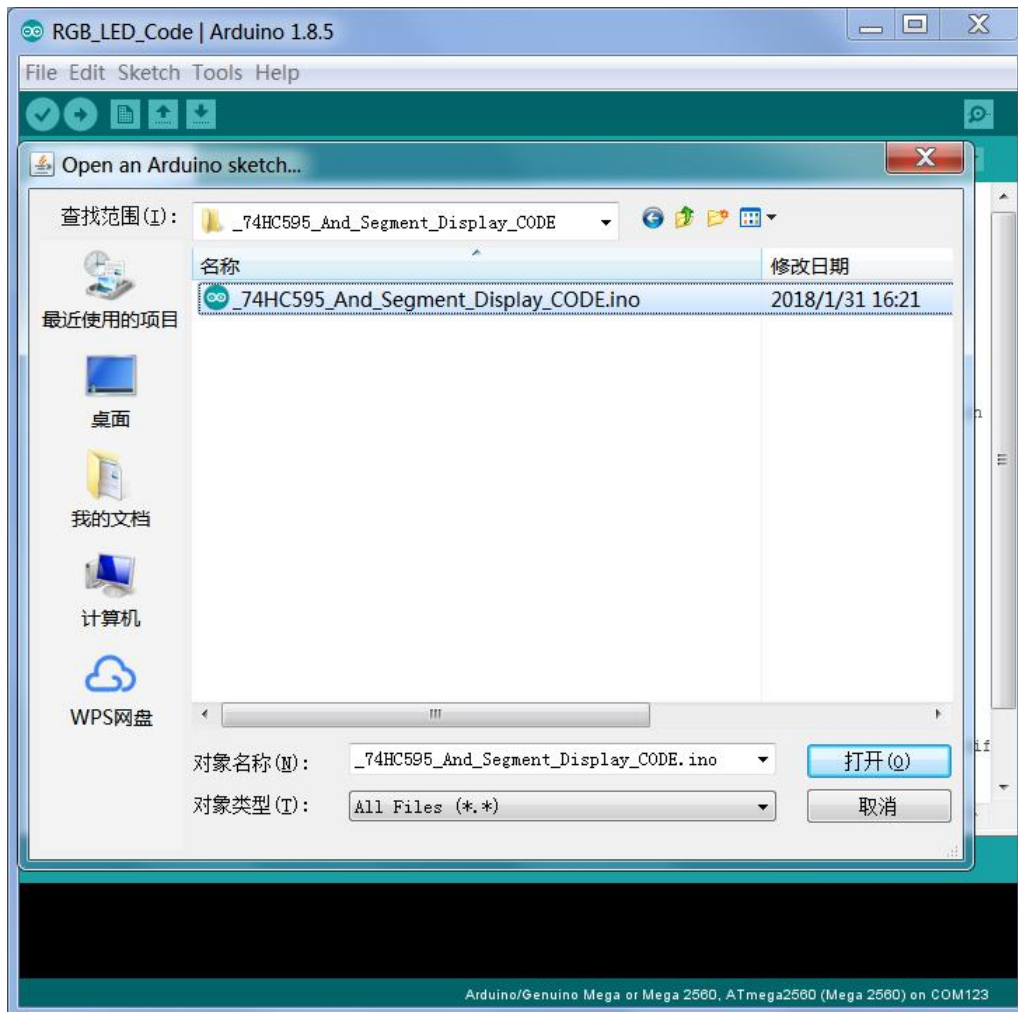
### Step 1: Build the circuit



## Schematic Diagram



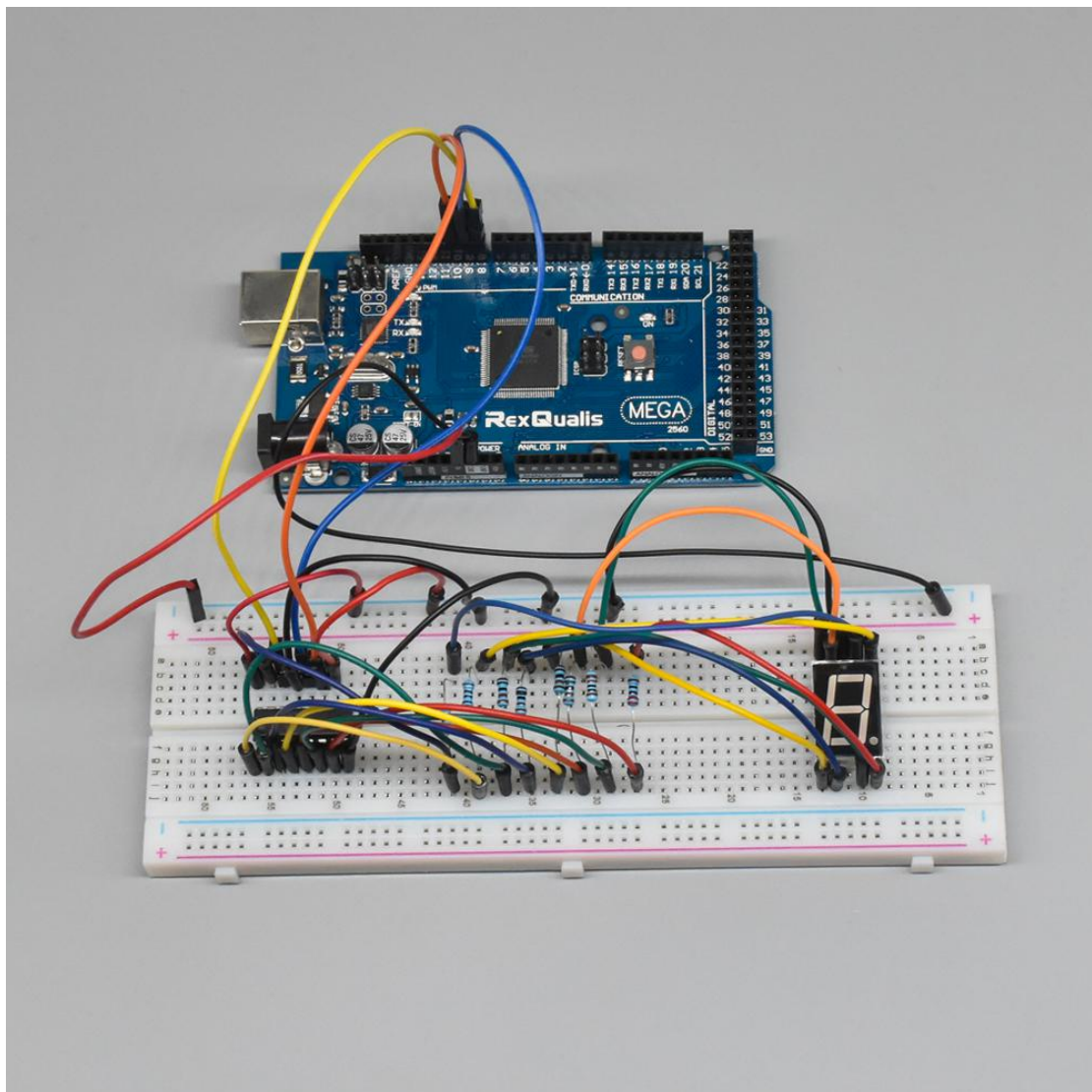
### Step 2:Open the code: 74HC595\_And\_Segment\_Display\_CODE



**Step 3:** Attach Arduino Mega 2560 board to your computer via USB cable and check that the '**Board Type**' and '**Serial Port**' are set correctly.

**Step 4:** Upload the code to the RexQualis Mega 2560 board.

Then, You can see the segment display show the number from 0-13(10=A,11=b,12=C,13=d).



You can see the video of the experiment results on YouTube:

<https://youtu.be/AUxntF4EBjE>

**If it isn' t working, make sure you have assembled the circuit correctly, verified and uploaded the code to your board. For how to upload the code and install the library, check Lesson 0 Preface.**