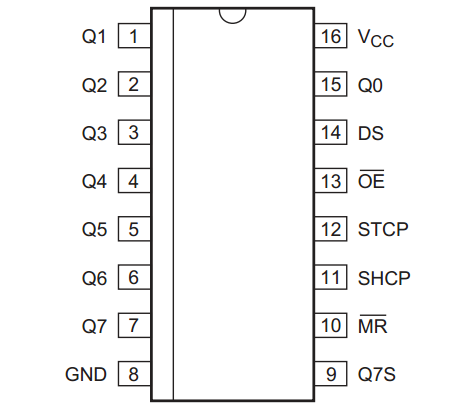
# **SN74HC595 drives 4-digit digital tube Experiment**

## 74HC595 Introduction

The 74HC595 is a CMOS shift register containing 8-bit serial input and parallel open-drain output that provides data to a register with three-state output. Shift register and storage register, respectively, have an independent clock respectively, the shift register 74 hc595 are needed with the highest priority (SRCLR) directly in the end, serial input (DS) used to cascade of serial output at the next higher level, when the output enable (OE) is a high end, 74 hc595 are needed in parallel in a high impedance state, output for the low level is enabled for parallel output.

Both the shift register clock SHCP and the storage register clock STCP are raised edge triggers.

Pinning information



## Pin description

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Pin** | **Description** |
| Q0 | 15 | parallel data output 0 |
| Q1 | 1 | parallel data output 1 |
| Q2 | 2 | parallel data output 2 |
| Q3 | 3 | parallel data output 3 |
| Q4 | 4 | parallel data output 4 |
| Q5 | 5 | parallel data output 5 |
| Q6 | 6 | parallel data output 6 |
| Q7 | 7 | parallel data output 7 |
| GND | 8 | ground(0V) |
| Q7S | 9 | serial data output |
| MR | 10 | master reset(active low) |
| SHCP | 11 | shift register clock input |
| STCP | 12 | storage register clock input |
| OE | 13 | output enable input(active low) |
| DS | 14 | serial data input |
| Vcc | 16 | supply voltage |

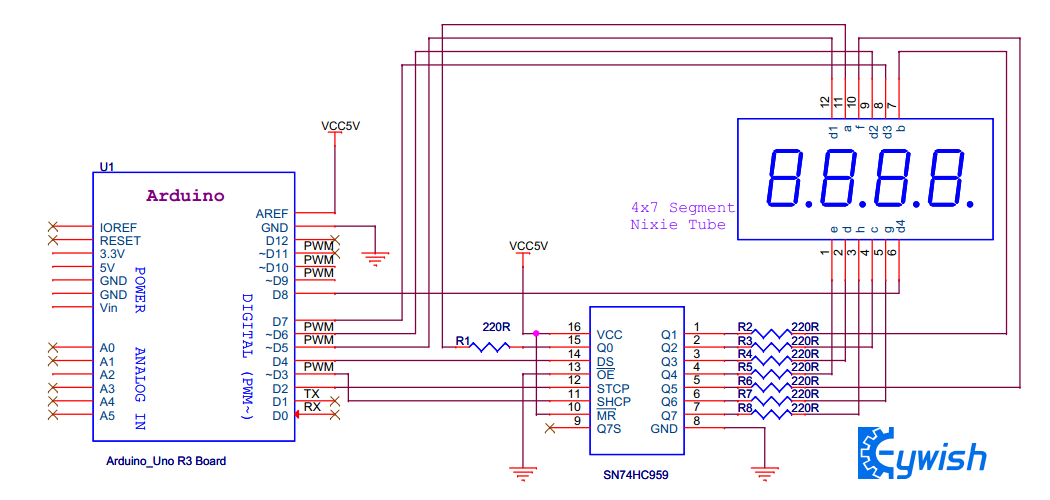
## Experiment Purpose

In this experiment, we used Arduino to drive the serial output to 74HC595, and then the parallel port of 74HC595 to drive a common cathode four-digit digital tube. Then driven directly by the Arduino digital tube of a foot, this experiment using eight 220 Ω resistance limit current role, let the dynamic display of digital tube digital has been reduced to 0 from 9.

## Component List

* Keywish Arduino UNO R3 mainboard
* 4-7Segment cathode tube \* 1
* SN74HC595 \* 1
* 220 Ω resistor\* 8
* Several jumper wires

## Schematic Diagram

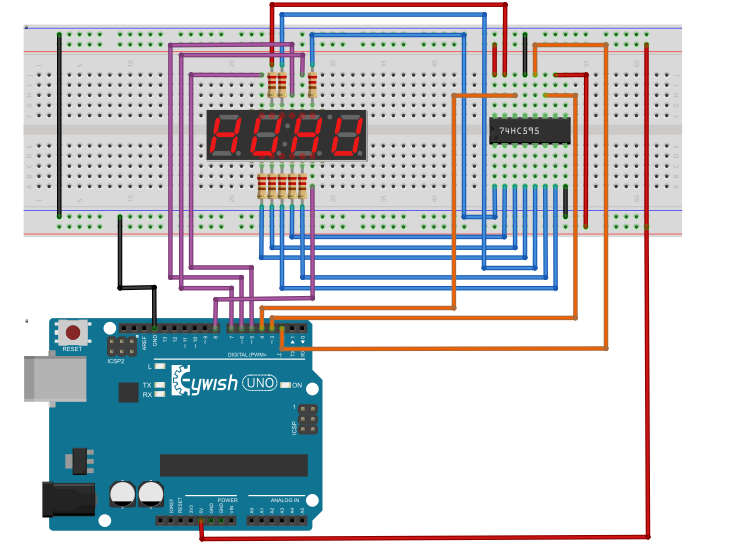


## Wiring of Circuit

|  |  |
| --- | --- |
| arduino Uno | SN74HC595 |
| 2 | 12(STCP) |
| 3 | 11(SHCP) |
| 4 | 14(DS) |

|  |  |
| --- | --- |
| arduino Uno | 7 Segment nixie tube |
| 5 | 12 |
| 6 | 9 |
| 7 | 8 |
| 8 | 6 |

|  |  |
| --- | --- |
| SN74HC595 | 7 Segment nixie tube |
| 15 | 11 |
| 1 | 7 |
| 2 | 5 |
| 3 | 2 |
| 4 | 1 |
| 5 | 10 |
| 6 | 5 |
| 7 | 3 |



## Code

#define BIT\_CHOICE\_1 5

#define BIT\_CHOICE\_2 6

#define BIT\_CHOICE\_3 7

#define BIT\_CHOICE\_4 8

#define STCP\_PIN 2

#define SHCP\_PIN 3

#define DATA\_PIN 4 //define stcp shcp ds pin

int BIT\_CHOICE**[**4**]=** **{**BIT\_CHOICE\_1 **,** BIT\_CHOICE\_2 **,**BIT\_CHOICE\_3**,**BIT\_CHOICE\_4**};**// 4x8bit

unsigned char

DisplayNumble**[**10**]={**0x00**,**0x6F**,**0x7F**,**0x07**,**0x7D**,**0x6D**,**0x66**,**0x4F**,**0x5B**,**0X06**};**

void setup**()**

**{**

pinMode**(**STCP\_PIN**,**OUTPUT**);**

pinMode**(**SHCP\_PIN**,**OUTPUT**);**

pinMode**(**DATA\_PIN**,**OUTPUT**);** //set stcp shcp ds pin putput mode

**for(**int i**=**0**;**i**<**4**;**i**++)**

**{**

pinMode**(**BIT\_CHOICE**[**i**],**OUTPUT**);**

digitalWrite**(**BIT\_CHOICE**[**i**],**HIGH**);**

**}**

**}**

void nixie\_pin\_chioce\_set**(**int value**)**

**{**

**for(**int i**=**0**;**i**<**4**;**i**++)**

**{**

digitalWrite**(**BIT\_CHOICE**[**i**],**value**);**

**}**

**}**

void loop**()**

**{**

int i**=**0**;**

**for(**i **=** 9**;** i**>=**0 **;**i**--** **)** // numble 9 - > 0 down

**{**

nixie\_pin\_chioce\_set**(**HIGH**);**

digitalWrite**(**STCP\_PIN**,**LOW**);**

shiftOut**(**DATA\_PIN**,**SHCP\_PIN**,**MSBFIRST**,**DisplayNumble**[**i**]);** //serial shift out put display numble

digitalWrite**(**STCP\_PIN**,**HIGH**);**

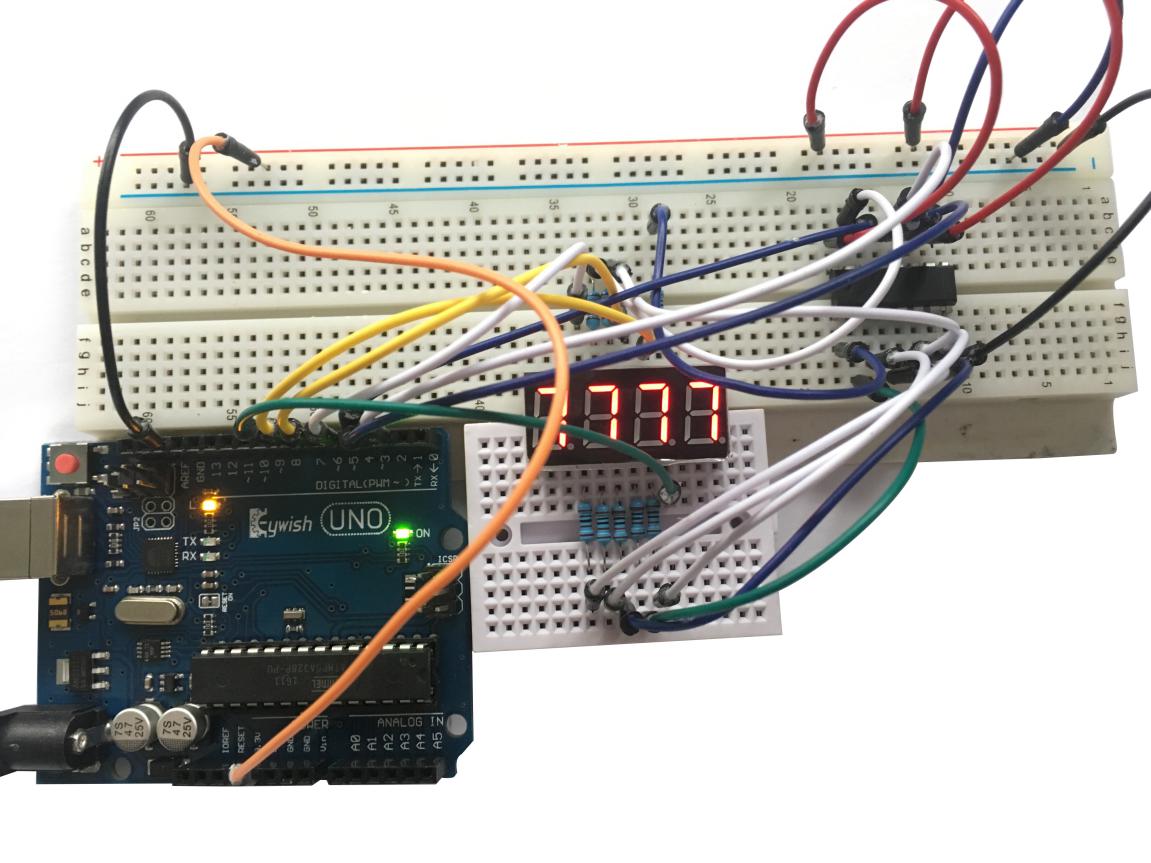
nixie\_pin\_chioce\_set**(**LOW**);**

delay**(**1000**);**

**}**

**}**

## Experiment Result



## Mixly programming program

