

74HC595

Overview

This course will use the 74595 chip to make a water light

Experimental Materials:

Raspberry Pi *1

T-type expansion board *1

5mm red LED light *2

5mm green LED light *2

5mm yellow LED light *2

5mm white LED light *2

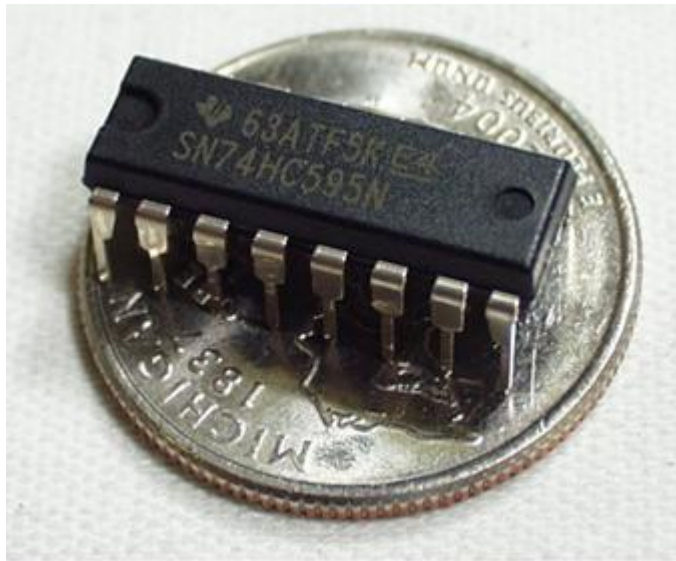
220 ohm resistor *8

Breadboard*1

Some DuPont lines

74HC595 *1

Product description:

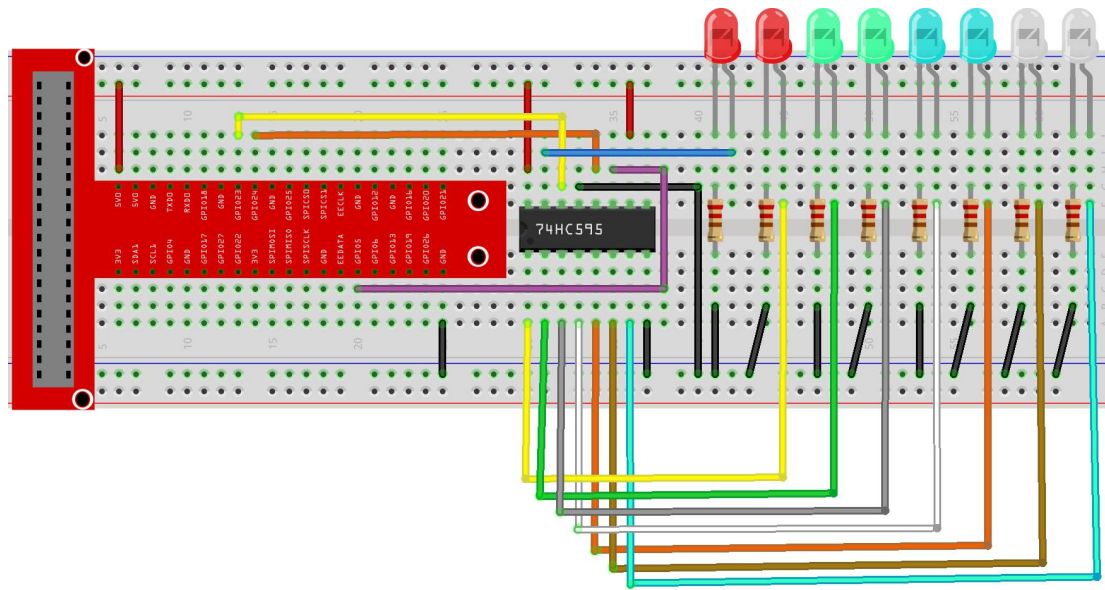


- Function: The 74HC595 is a shift register. You only need three pins on the microcontroller to control eight outputs at the same time.
- Application: more digital interface control occasions

Technical Parameters:

- ◆ Package Type: DIP
- ◆ 8-bit serial input, parallel output shift
- ◆ Operating voltage range: 2.0 to 6.0 V
- ◆ High-intensity current tri-state output can drive up to 15 TTL loads
- ◆ 5 V drive current is ± 6 mA
- ◆ Operating temperature range: -40°C to $+85^{\circ}\text{C}$

Wiring diagram:



C code:

```
#include <wiringPi.h>
#include <stdio.h>

#define SDI 4 //serial data input
#define RCLK 5 //memory clock input(STCP)
#define SRCLK 21 //shift register clock input(SHCP)

unsigned char LED[8] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80};

void pulse(int pin) {
    digitalWrite(pin, 0);
    digitalWrite(pin, 1);
}

void SI(unsigned char byte) {
    int i;
```

```

    for(i=0;i<8;i++){
        digitalWrite(SDI, ((byte & (0x80 >> i)) > 0));
        pulse(SRCLK);
    }
}

void init(void){
    pinMode(SDI, OUTPUT); //make P23 output
    pinMode(RCLK, OUTPUT); //make P24 output
    pinMode(SRCLK, OUTPUT); //make P5 output

    digitalWrite(SDI, 0);
    digitalWrite(RCLK, 0);
    digitalWrite(SRCLK, 0);
}

int main(void){
    int i;

    if(wiringPiSetup() == -1){ //when initialize wiring failed, print message to
screen
        printf("setup wiringPi failed !");
        return 1;
    }

    init();

    printf("\n");
    printf("\n");
    printf("=====\n");
    printf("|          LEDs with 74HC595          |\n");
    printf("|          Control LEDs with 74HC595      |\n");
    printf("|                                          |\n");
    printf("=====\n");
    printf("\n");
    printf("\n");

    while(1){
        for(i=0;i<8;i++){
            SI(LED[i]);
            pulse(RCLK);
            delay(150);

```

```

        //printf("i = %d\n", i);
    }
    delay(500);

    for(i=0;i<3;i++){
        SI(0xff);
        pulse(RCLK);
        delay(100);
        SI(0x00);
        pulse(RCLK);
        delay(100);
    }
    delay(500);
//    digitalWrite(RCLK, 0);

    for(i=0;i<8;i++){
        SI(LED[8-i-1]);
        pulse(RCLK);
        delay(150);
    }
    delay(500);

    for(i=0;i<3;i++){
        SI(0xff);
        pulse(RCLK);
        delay(100);
        SI(0x00);
        pulse(RCLK);
        delay(100);
    }
    delay(500);
}

return 0;
}

```

Python Code:

```
#!/usr/bin/env python
```

```

#=====
#
#

```

```

# Extend use of 8 LED with 74HC595.
# Change the WhichLeds and sleeptime value under
# loop() function to change LED mode and speed.
#
#=====

import RPi.GPIO as GPIO
import time

SDI    = 23
RCLK   = 24
SRCLK  = 5

LED0 = [0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80]    #original mode
BLINK = [0xff, 0x00, 0xff, 0x00, 0xff, 0x00]                #blink
LED1 = [0x01, 0x03, 0x07, 0x0f, 0x1f, 0x3f, 0x7f, 0xff]    #blink mode 1
LED2 = [0x01, 0x05, 0x15, 0x55, 0xb5, 0xf5, 0xfb, 0xff]    #blink mode 2
LED3 = [0x02, 0x03, 0x0b, 0x0f, 0x2f, 0x3f, 0xbf, 0xff]    #blink mode 3
#=====

def print_message():
    print ("=====")
    print ("|                LEDs with 74HC595                |")
    print ("|                |")
    print ("|          Control LEDs with 74HC595          |")
    print ("|                |")
    print ("=====\\n")
    print 'Program is running...'
    print 'Please press Ctrl+C to end the program...'
    raw_input ("Press Enter to begin\\n")

def setup():
    GPIO.setmode(GPIO.BCM)    # Number GPIOs by its BCM location
    GPIO.setup(SDI, GPIO.OUT, initial=GPIO.LOW)
    GPIO.setup(RCLK, GPIO.OUT, initial=GPIO.LOW)
    GPIO.setup(SRCLK, GPIO.OUT, initial=GPIO.LOW)

# Shift the data to 74HC595
def hc595_shift(dat):
    for bit in range(0, 8):
        GPIO.output(SDI, 0x80 & (dat << bit))
        GPIO.output(SRCLK, GPIO.HIGH)
        time.sleep(0.001)

```

```

        GPIO.output(SRCLK, GPIO.LOW)
GPIO.output(RCLK, GPIO.HIGH)
time.sleep(0.001)
GPIO.output(RCLK, GPIO.LOW)

def main():
    print_message()
    mode = LED0 # Change Mode, modes from LED0 to LED3
    sleeptime = 0.15 # Change speed, lower value, faster speed
    blink_sleeptime = 0.3
    leds = ['-', '-', '-', '-', '-', '-', '-', '-']
    while True:
        # Change LED status from mode
        print "    mode"
        for m in mode:
            hc595_shift(m)
            leds[mode.index(m)] = 1 # Show which led is on
            print leds
            time.sleep(sleeptime)
            leds[mode.index(m)] = '-' # Show the led is off

        print "    blink"
        for m in BLINK:
            hc595_shift(m)
            if (m == 0x00):
                leds = ['-'] * 8
            elif (m == 0xff):
                leds = [1] * 8
            print leds
            time.sleep(blink_sleeptime)

    # Change LED status from mode reverse
    print "    reversed mode"
    for m in reversed(mode):
        hc595_shift(m)
        leds[mode.index(m)] = 1 # Show which led is on
        print leds
        time.sleep(sleeptime)
        leds[mode.index(m)] = '-' # Show the led is off

    print "    blink"
    for m in BLINK:
        hc595_shift(m)
        if (m == 0x00):

```

```

        leds = ['-' ] * 8
    elif (m == 0xff):
        leds = [1] * 8
    print leds
    time.sleep(blink sleeptime)

def destroy():
    GPIO.cleanup()

if __name__ == '__main__':
    setup()
    try:
        main()
    except KeyboardInterrupt:
        destroy()

```

Experimental results:

In the directory where the code file is located, execute the following command

C:

```
gcc -Wall -o 74HC595 74HC595.c -lwiringPi
sudo ./74HC595
```

Python:

```
python 74HC595.py
```

After the instruction is executed, led lights up in accordance with certain rules

