

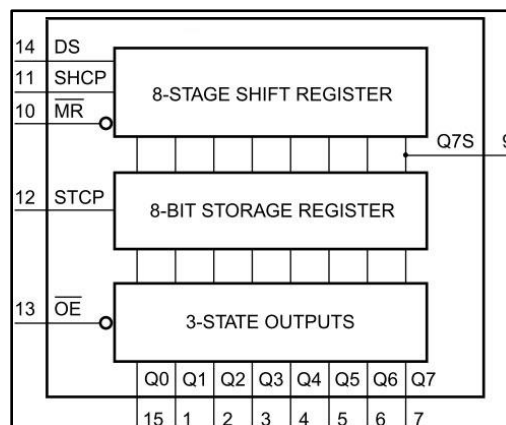
Bài 2: MỞ RỘNG I/O DÙNG THANH GHI DỊCH

2.1. Mục tiêu:

- Sinh viên tìm hiểu hoạt động của IC chuyển dữ liệu từ nối tiếp sang song song 74HC595.
- Ứng dụng thanh ghi dịch (cụ thể là IC74HC595) trong giao tiếp với Led đơn
- Viết chương trình điều khiển nhiều Led đơn, tạo hiệu ứng sáng Led qua giao tiếp với IC 74HC595

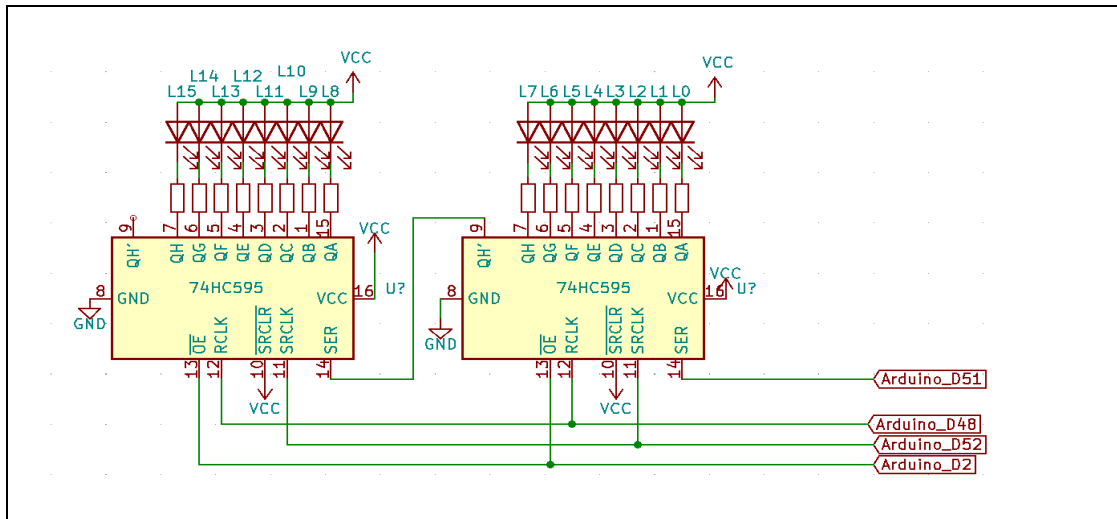
2.2. Điều khiển 16 Led đơn dùng thanh ghi dịch

IC 74HC595 là một chip chuyển dữ liệu từ nối tiếp sang song song, cấu trúc bên trong gồm 3 tầng: thanh ghi dịch, bộ chốt và điều khiển ngõ ra 3 trạng thái. Các chân chức năng được mô tả trong sơ đồ khối bên dưới.



Hình 2.1: Sơ đồ khối IC74HC595

IC 74HC595 được dùng để điều khiển 16 Led đơn, các tín hiệu điều khiển từ Arduino được mô tả như trong hình bên dưới,



Chương trình điều khiển để các Led đơn sáng tắt xen kẽ.

```
#define Latch_595 48 //chot led don
#define clock_595 52 //clock
#define data_595 51 //MOSI
#define OE_595 2
#define onLatch digitalWrite(Latch_595,1)
#define offLatch digitalWrite(Latch_595,0)
void setup()
{
    pinMode(Latch_595,OUTPUT);
    pinMode(clock_595,OUTPUT);
    pinMode(data_595,OUTPUT);
    pinMode(OE_595,OUTPUT);
    digitalWrite(OE_595,0);
}
/*
    Chương trình xuất led
    Data 16 bit
*/
void xuấtLed(int Data){
    for(int i=0;i<=15;i++){
        digitalWrite(data_595,!((Data<<i)&0x8000));
        shiftClock();
    }
    onLatch;
    offLatch;
}
void shiftClock(){
    digitalWrite(clock_595,1);
    digitalWrite(clock_595,0);
}
void loop()
{
    xuấtLed(0x5555);
    delay(500);
}
```

```
xuatLed(0xAAAA);  
delay(500);  
}
```

1. Bạn hãy soạn thảo chương trình trên và nạp vào Kit để xem kết quả. Giải thích hoạt động của chương trình.
2. Với chương trình trên hãy thay thế hàm delay(500) bằng kỹ thuật delay dùng hàm millis() để đảm bảo hoạt động của chương trình không thay đổi.
3. Arduino có hỗ trợ hàm **shiftOut()** để thực hiện ghi dịch từng bit dữ liệu ra một chân port của arduino. Bạn hãy tìm hiểu hàm **shiftOut** theo link:

<https://www.arduino.cc/reference/en/language/functions/advanced-io/shiftout/> và áp dụng hàm này, sửa lại hàm xuatLed(int Data) của trình bên trên để chương trình hoạt động được với hàm shiftOut. Hàm shiftOut trong thư viện Arduino được khai báo trong file :

Table 1: Hàm shiftOut của Arduino

```
void shiftOut(uint8_t dataPin, uint8_t clockPin, uint8_t bitOrder, uint8_t val)  
{  
    uint8_t i;  
    for (i = 0; i < 8; i++) {  
        if (bitOrder == LSBFIRST) {  
            digitalWrite(dataPin, val & 1);  
            val >>= 1;  
        } else {  
            digitalWrite(dataPin, (val & 128) != 0);  
            val <<= 1;  
        }  
        digitalWrite(clockPin, HIGH);  
        digitalWrite(clockPin, LOW);  
    }  
}
```

4. Bạn hãy dựa theo hàm **shiftOut** của arduino để thiết kế lại một hàm mới đặt tên là **shiftOut16**, có chức năng dịch 16 bit dữ liệu vào thanh ghi dịch.

5. Ngoài 2 cách ghi dịch dữ liệu bên trên, khối SPI trong arduino cũng hỗ trợ ghi dịch dữ liệu nhờ vào hoạt động của khối này thông qua thư viện <SPI.h>. Hàm `SPI.begin()` để khởi tạo khối SPI bên trong chip Atmega2560, hàm **`SPI.transfer(data)`** dùng để ghi dịch dữ liệu data ra chân MOSI của khối SPI. Bạn hãy sửa lại phần ghi dịch dữ liệu của chương trình trên bằng cách dùng các hàm trong thư viện **`SPI.h`** để chương trình hoạt động đúng chức năng.
6. Viết chương trình tạo hiệu ứng 16 led sáng và tắt dần, có thể dùng bất kỳ cách nào để ghi dịch dữ liệu.
7. Viết chương trình tạo tối thiểu 3 hiệu ứng ánh sáng trên 16 led đơn và các hiệu ứng được chuyển trạng thái tuần tự theo nút nhấn trên Kit đã được học ở bài thực hành số 1.