

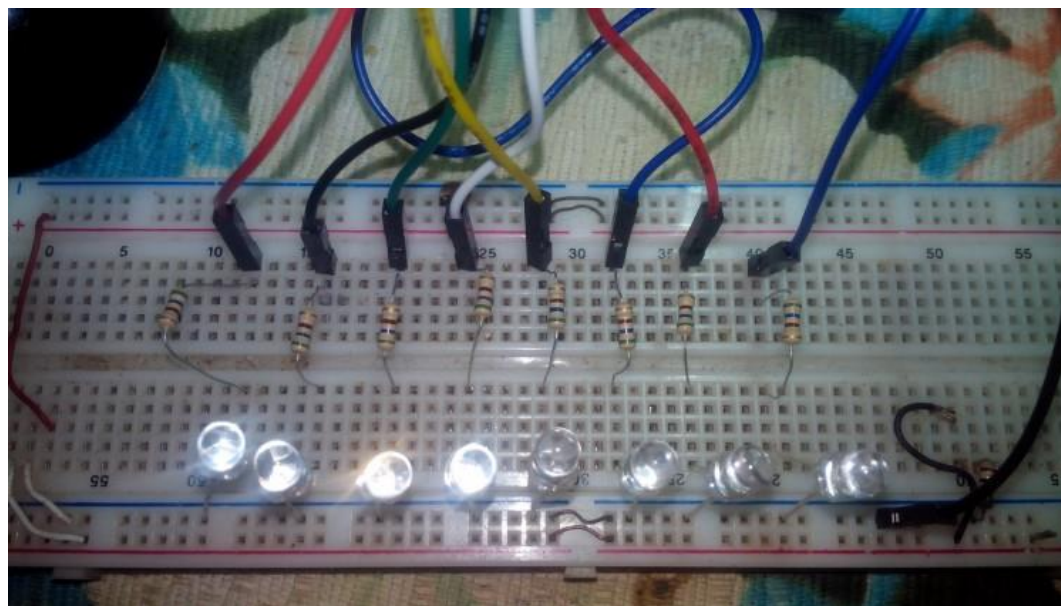
ĐIỀU KHIỂN 8 ĐÈN LED SÁNG THEO Ý MUỐN

Giới thiệu, nội dung cần nắm

Hôm nay, chúng ta sẽ học cách điều khiển 8 đèn LED. Vấn đề này, vừa dễ lại vừa khó, vậy nó dễ chỗ nào, khó chỗ nào, chúng ta cùng nhau tìm hiểu nhé!

Qua bài học này, bạn sẽ hiểu được cách làm thế nào để điều khiển nhiều led bằng cách sử dụng các chân digital, hoặc sử dụng IC HC595!

Bạn sẽ làm được gì trong ngày hôm nay?



Hình ảnh mô tả đoạn mạch đã được lắp đặt và chạy thử

Phần cứng

- [Arduino Uno](#)
- 8 điện trở [560 Ohm](#) (hoặc [220 Ohm](#) hoặc [1kOhm](#))
- [Breadboard](#)
- [Dây cắm breadboard](#)
- 8 đèn [LED siêu sáng](#) (tất cả các loại màu đều được cả, bạn không cần chuẩn bị 8 LED sáng cùng màu như tôi, hãy thử mua nhiều LED khác màu và lắp nhấp nháy theo ý mình nhé).

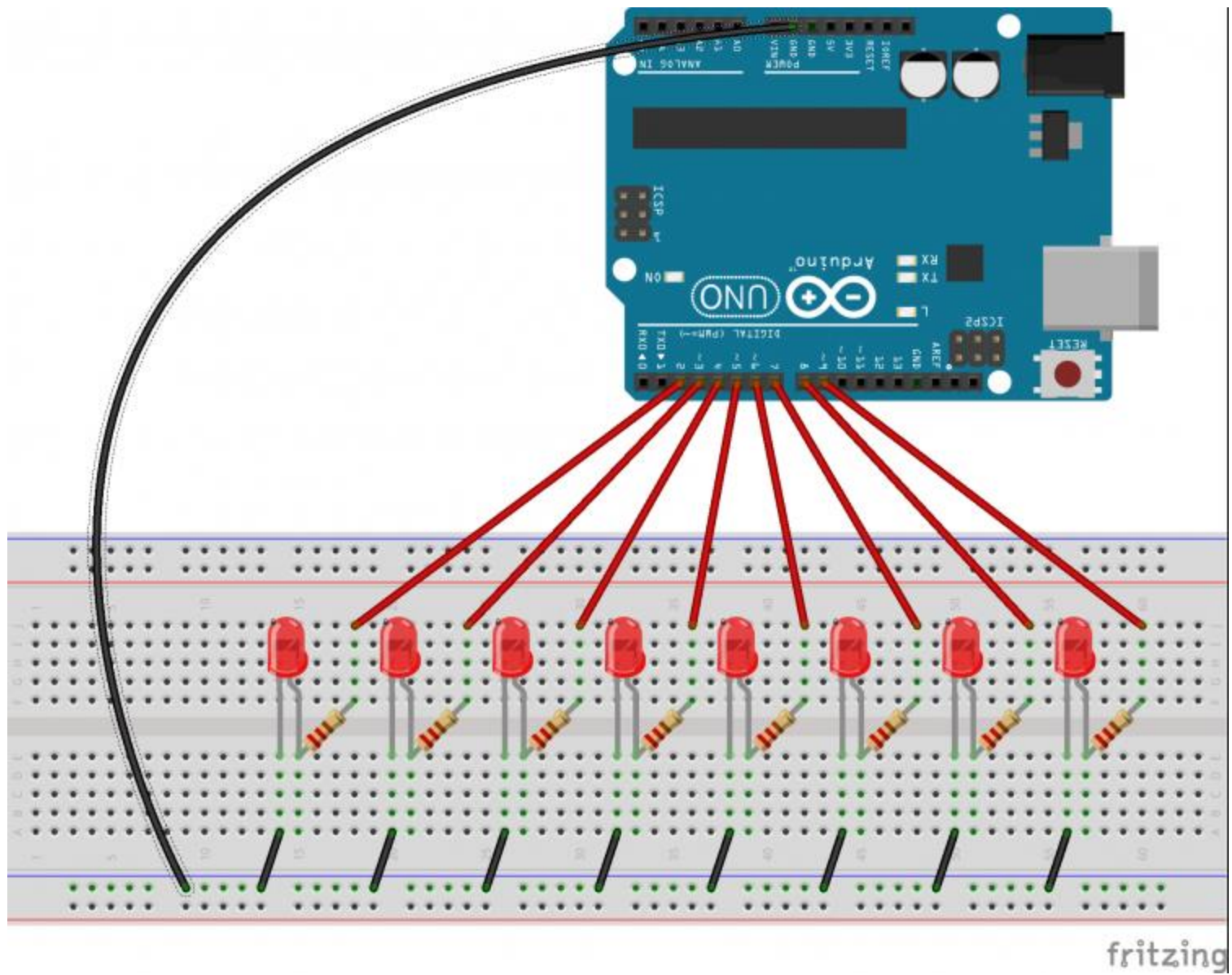
Thời gian

1. Nếu bạn chỉ tham khảo bài viết và làm theo hướng dẫn để tạo mạch LED 8 đèn thì bạn cần 20 phút.
2. Còn nếu bạn muốn tham khảo cách làm với nhiều LED hơn, thì bạn cần nhiều thời gian (1 tiếng hoặc hơn) và cần nhiều phần cứng hơn:
 - [IC 74HC595](#) (trên thị trường nhất nhiều loại, bạn chỉ cần tìm IC nào có chữ này là được). Nếu bạn muốn điều khiển nhiều LED hơn thì bạn hãy mua nhiều hơn 1 con (cứ 1 con điều khiển được 8 LED)!

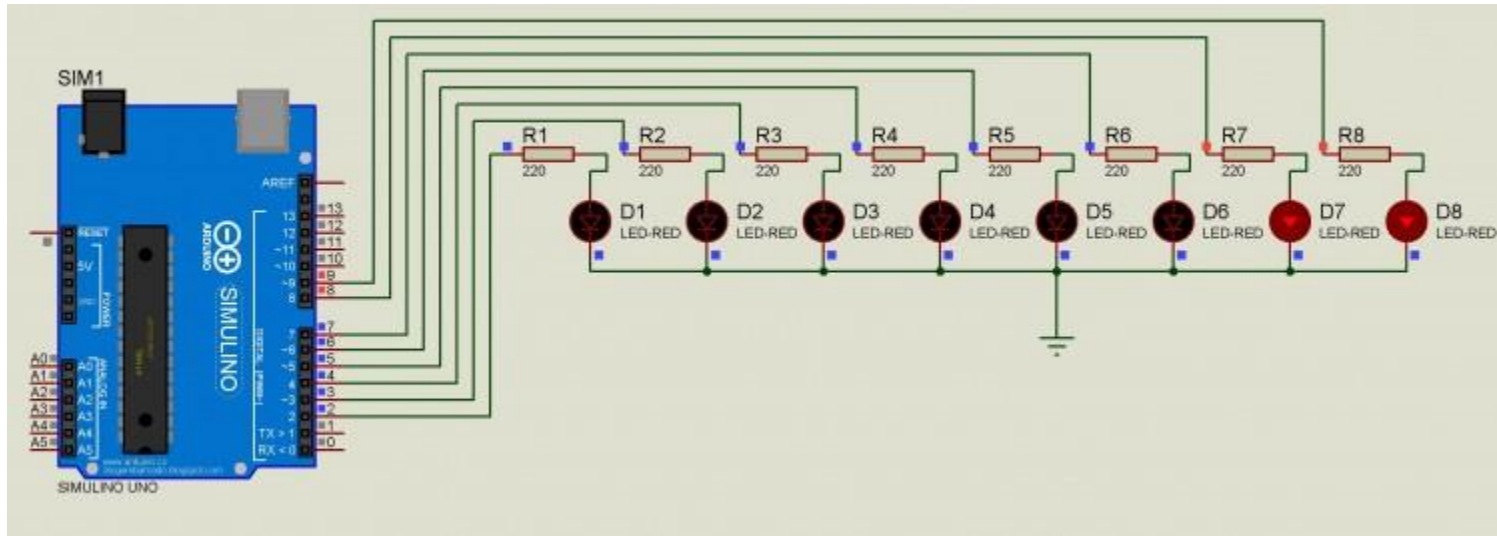
Bây giờ, không làm mất thời gian của bạn nữa, tiến hành thôi!

Thứ nhất, điều khiển 8 LED bằng 8 chân Digital của Arduino!

Bạn hãy lắp mạch như sau, kích vào hình để xem ảnh "bự" nhé.



bạn có thể thích cách mắc như thế này



Sau đây là đoạn code lập trình, tôi đã chú thích trong đoạn code khá rõ ràng, bạn chỉ cần copy vào dán vào Arduino IDE thôi, một số kiến thức bạn cần tham khảo nếu chưa biết là:

1. pinMode
2. digitalWrite
3. array
4. for
5. delay
6. int
7. byte
8. sizeof

```
byte ledPin[] = {2,3,4,5,6,7,8,9}; // Mảng lưu vị trí các chân Digital mà các đèn LED sử dụng theo thứ tự từ 1->8. Bạn có thể thêm các LED bằng cách thêm các chân digital vào mảng này
byte pinCount; // Khai báo biến pinCount dùng cho việc lưu tổng số chân LED
```

```

void setup() {
    pinCount = sizeof(ledPin); //Xem thêm thông tin về hàm sizeof tại
http://arduino.vn/reference/sizeof
    for (int i=0;i<pinCount;i++) {
        pinMode(ledPin[i],OUTPUT); //Các chân LED là OUTPUT
        digitalWrite(ledPin[i],LOW); //Mặc định các đèn LED sẽ tắt
    }
}

void loop() {
    /*
        Bật tuần tự các đèn LED
    */
    for (int i=0; i < pinCount; i++) {
        digitalWrite(ledPin[i],HIGH); //Bật đèn
        delay(500); // Dừng để các đèn LED sáng dần
    }

    /*
        Tắt tuần tự các đèn LED
    */
    for (int i = 0;i < pinCount; i += 1) {
        digitalWrite(ledPin[i],LOW); // Tắt đèn
        delay(500); // Dừng để các đèn LED tắt dần
    }
}

```

Và bạn đã hoàn thành bài hướng dẫn ngày hôm nay rồi đấy ?? Vẫn chưa đến phần hay mà, nhưng nếu muốn đi tiếp bạn phải dành thời gian và mua IC HC595 đây. Ngoài ra, bạn có thể tải về bản mô phỏng Proteus

Thứ hai, phân tích những ưu điểm, nhược điểm ở mạch điều khiển 8 LED bằng 8 chân digital

Ưu điểm:

1. Dễ lắp đặt

2. Dễ lập trình
3. Dễ hiểu
4. Đồ điện sẵn có từ các ví dụ khác.

Nhược điểm không thể khắc phục:

1. Bạn chỉ có thể điều khiển tối đa được 20 đèn LED với cách này mà thôi (dù bạn có pro đến mức nào đi nữa)
2. Nếu mang tặng người yêu mà chỉ 20 đèn LED thì quả thật sẽ rất khó làm được nhiều ứng dụng hay, đúng không nào ?

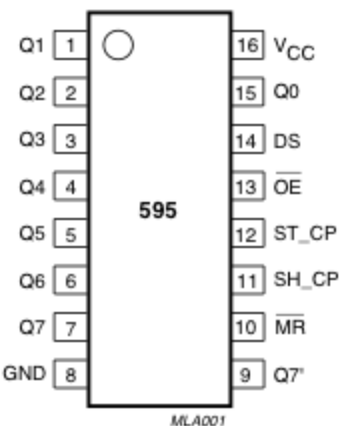
Vậy, có cách nào khác để khắc phục mà lại cực kỳ dễ dàng với người mới làm không? Vâng, CÓ!

Thứ ba, điều khiển nhiều đèn LED (bao nhiêu cũng được, phụ thuộc vào thời gian của bạn mà thôi) chỉ với 3 chân Digital 

Bạn có tin được hay không, chỉ với 3 chân digital mà thôi, bạn có thể điều khiển được rất nhiều đèn LED, mà bạn chỉ cần một mạch Arduino UNO R3 và vài con IC HC 595. Không nói nhiều nữa, tiến hành ngay thôi!

Kiến thức bạn cần biết thêm cũng không nhiều đâu, tôi sẽ hướng dẫn một cách dễ hiểu nhất, hãy kiên trì bạn nhé.

	PINS 1-7, 15	Q0 đến Q7	các chân xuất tín hiệu, giống như các chân Digital được cài đặt là OUTPUT
	PIN 8	GND	Ground, Cực âm
	PIN 9	Q7"	Chân xuất ra tín hiệu Serial
	PIN 10	MR	Master Reclear, nối cực dương để bật IC hoạt động
	PIN 11	SH_CP	Shift register clock pin
	PIN 12	ST_CP	Storage register clock pin (latch pin)

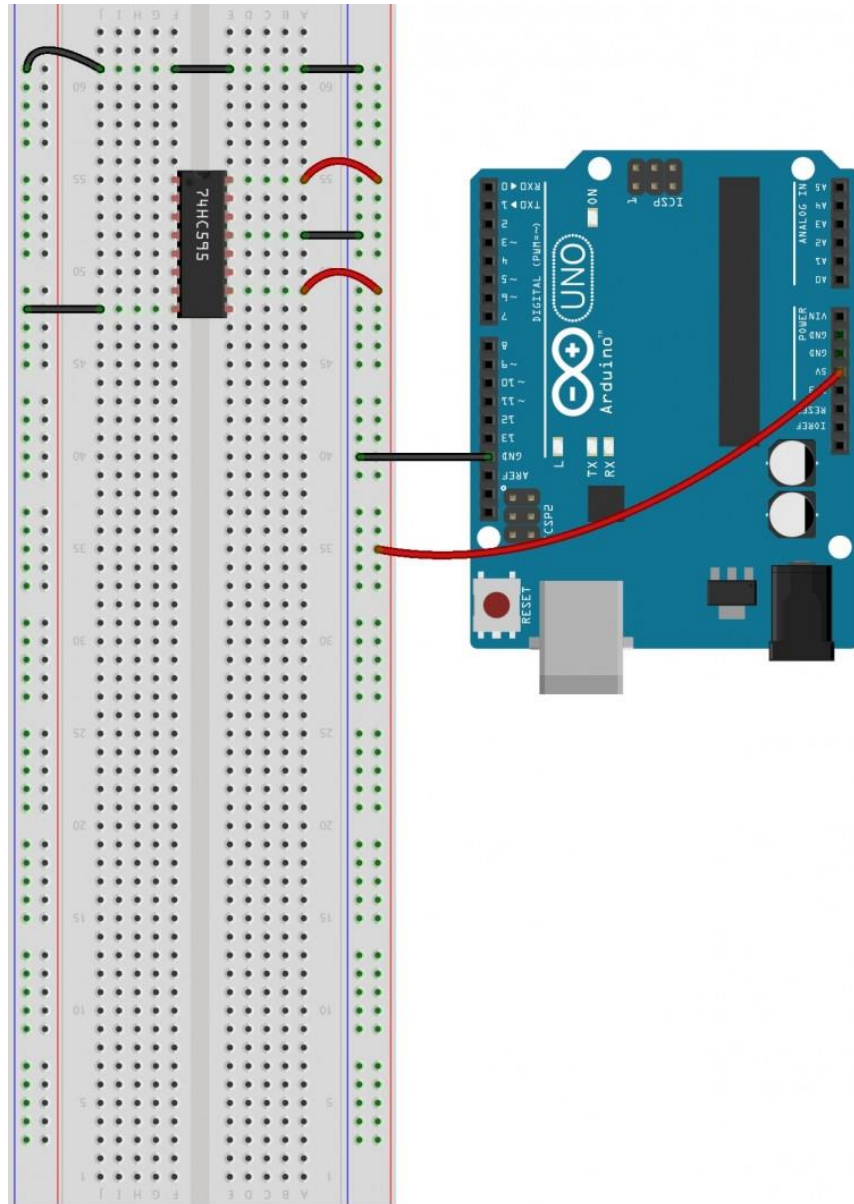
	PIN 13	OE	Output enable, nối cực âm để các đèn LED có thể sáng được.
	PIN 14	DS	Serial data input
	PIN 16	Vcc	Cấp nguồn cho IC và LED.

Thứ tư, điều khiển 8 LED với 1 IC 595

Lưu ý nho nhỏ với các bạn là các bạn hãy giữ nguyên các chân cắm của mạch 8 LED nhấp nháy chúng ta vừa làm ở trên nhé, chỉ tháo các khớp nối giữa các dây breadboard với Arduino thôi nha.

Đầu tiên, chúng ta cần bật IC 595 lên. Bạn hãy nối mạch như sau:

- GND (pin 8) nối đến cực âm
- Vcc (pin 16) nối đến chân 5V
- OE (pin 13) nối đến cực âm
- MR (pin 10) nối đến chân 5V



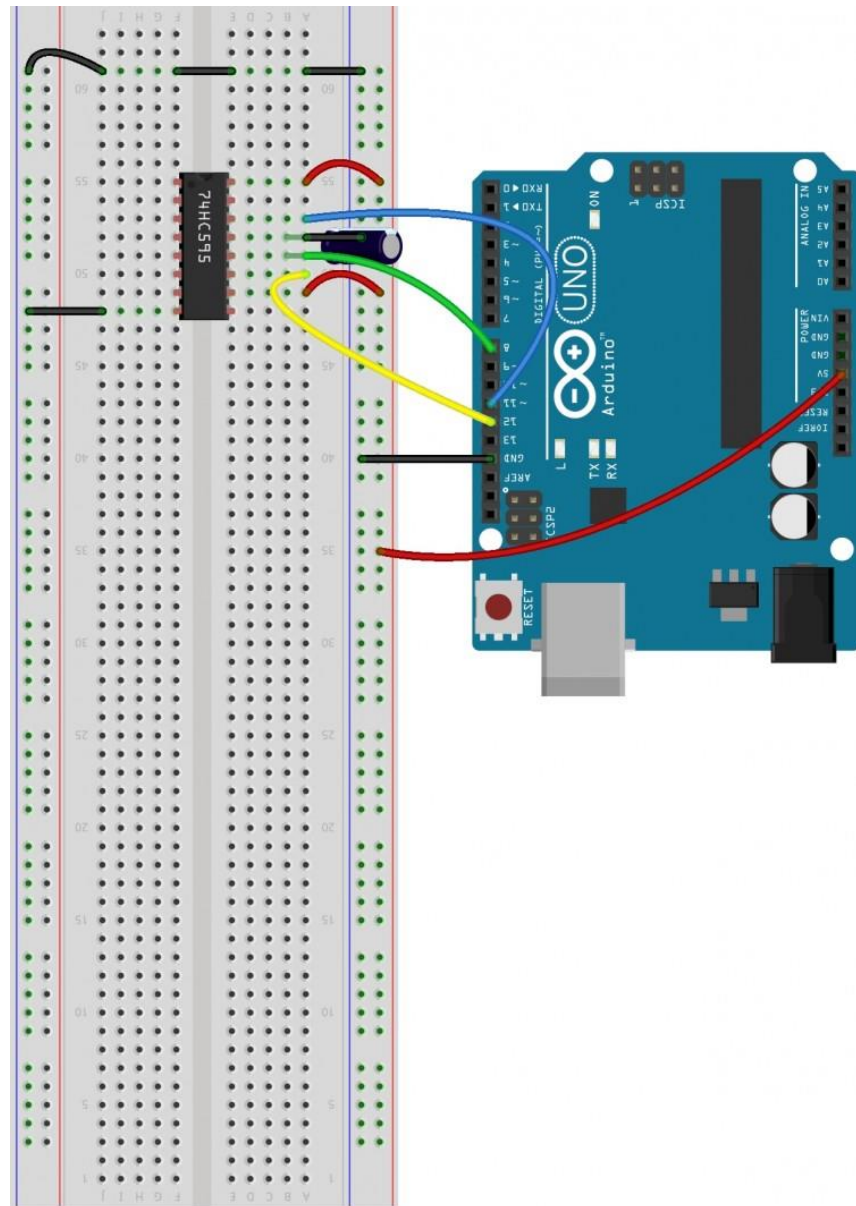
fritzing

Tại sao lại mắc như vậy? Đó là bởi vì trong datasheet (tài liệu thông tin vi mạch) của 595 yêu cầu như vậy. Ngoài ra, bạn cũng có thể mắc các chân OE, MR bằng các chân digital của Arduino (để bật tắt các IC 595) mà thôi tôi nghĩ điều này không cần thiết đâu, với lại như vậy lại tốn thêm nhiều chân digital 😊.

Tiếp theo, chúng ta sẽ nối Arduino với IC HC595.

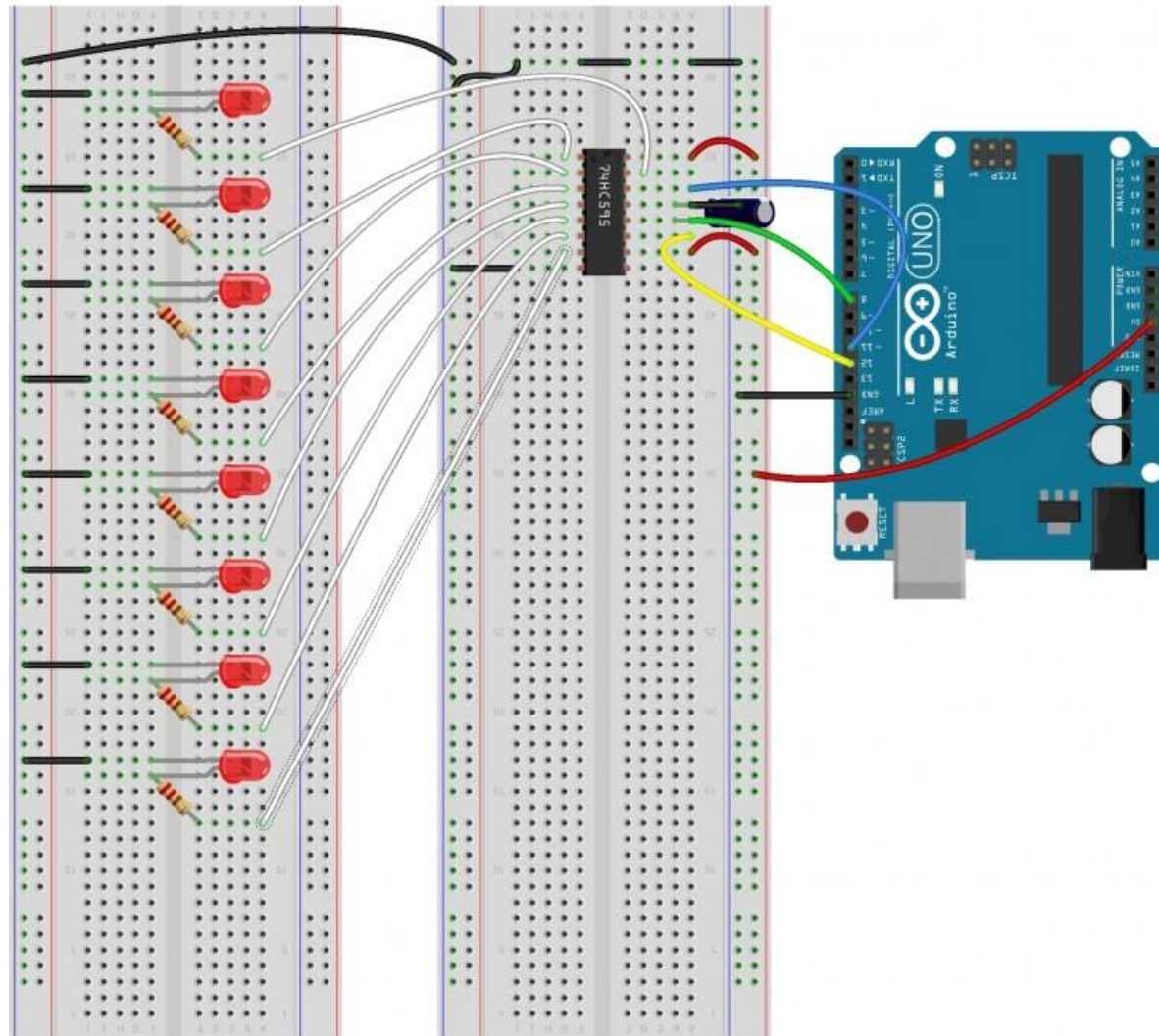
- DS (pin 14) đến Arduino DigitalPin 11 (dây xanh nước biển)
- SH_CP (pin 11) đến Arduino DigitalPin 12 (dây màu vàng)
- ST_CP (pin 12) đến Arduino DigitalPin 8 (dây màu xanh lá).

Lưu ý, vì một số IC HC 595 bị nhiễu trong quá trình shiftout. Nếu gặp tình trạng như vậy thì bạn đừng lo, chúng ta đã có cách, bạn chỉ mắc một tụ điện khoảng 0,1 uF (micro pha ra) (cực dương của tụ gắn với chân digital HC 595 và cực âm vào cực âm của nguồn).



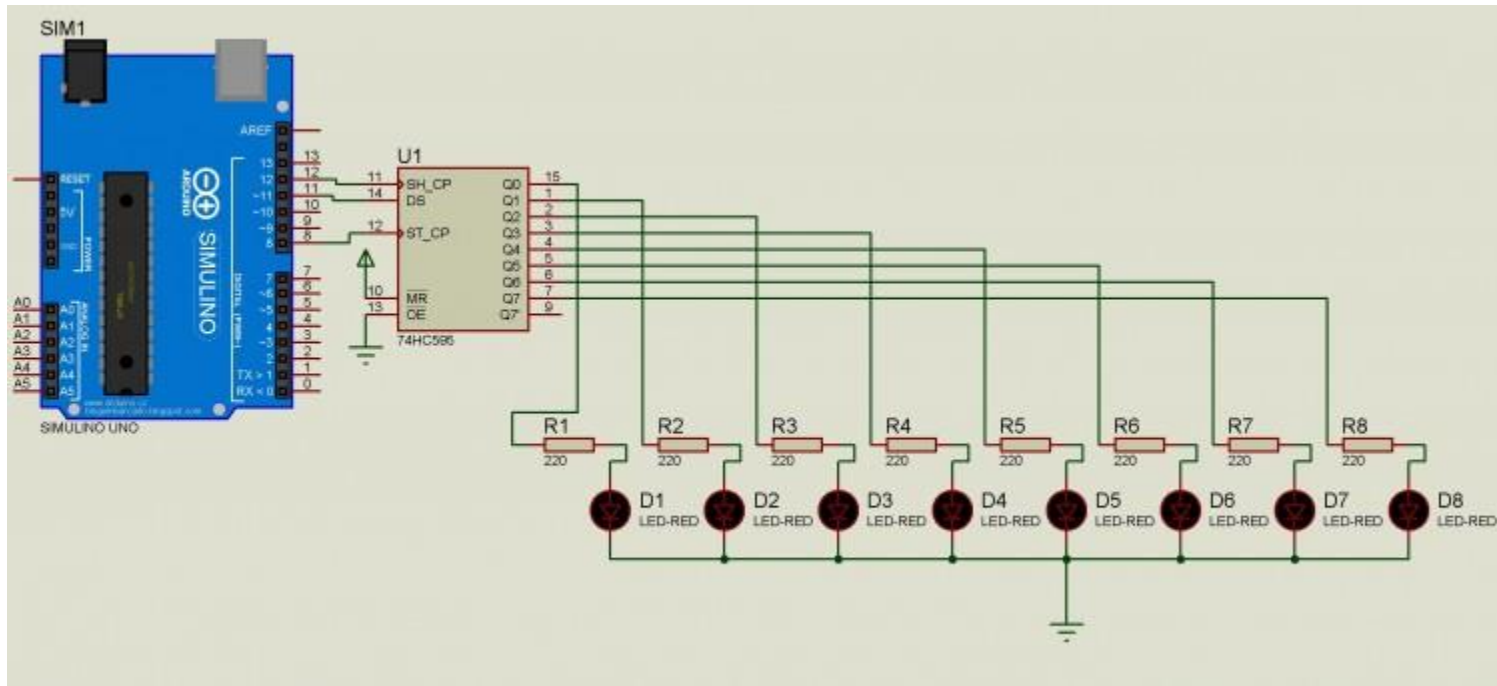
fritzing

Sau đó, bạn sử dụng 8 đèn LED của chúng ta đã mắc ban đầu và mắc theo như sơ đồ dưới đây!



fritzing

bạn có thể thích cách mặc này hơn



Cuối cùng, phần quan trọng nhất của buổi học hôm nay của chúng ta, đó là lập trình để điều khiển những con LED này!

Để điều khiển được LED qua IC HC 595, chúng ta phải làm quen với một kỹ thuật, được gọi là shiftOut. Nói vậy thôi chứ không có khó đâu, bạn chỉ cần hiểu đơn giản, shiftOut là việc gửi tín hiệu cho 1 IC có hỗ trợ shiftOut (ví dụ HC 595 này), cứ mỗi lần gửi nó gửi 1 **byte** (không hơn không kém), mỗi 1 bit (có tổng cộng 8 bit trong 1 byte) sẽ quản lý giá trị điện tại chân tín hiệu của HC 595 (các chân có tên là Q0-Q7).

Bây giờ dễ để hình dung, chúng ta sẽ tưởng tượng như thế này:

1. Các LED sẽ sáng nếu được cấp điện thế dương (5V chẳng hạn) vào cực dương và cực âm của LED được gắn vào cực âm của nguồn (hiển nhiên rồi), vì vậy chúng ta sẽ gắn cực âm của LED vào nguồn và nếu muốn điều

khởi cho đèn LED sáng thì chỉ quản lý điện thế gắn vào cực dương của LED, nếu là âm thì đèn tắt, nếu là dương thì đèn sáng.

2. Còn IC 595 sẽ nhận lệnh shiftOut của Arduino và sẽ quản lý 8 chân tín hiệu của nó (các chân có tên Q0-Q7) bằng 1 byte \Leftrightarrow 8 bit (từ bit 0 đến bit 7). Nếu tại bit 0 có giá trị là 1 \Rightarrow Q0 được nối với nguồn 5V, nếu bit 0 có giá trị là 0 thì Q0 sẽ được nối tới cực âm, tương tự với các 7 bit còn lại.
3. Bây giờ, chúng ta sẽ dùng shiftOut để quản lý hiệu điện thế các chân từ Q0 đến Q7 của IC HC595, từ đó, điều khiển việc bật / tắt đèn.

Lợi ích của việc này, đó là bạn chỉ cần 3 chân digital như đã hướng dẫn mắc ở trên là có thể điều khiển được bao nhiêu LED tùy ý (cứ mỗi con HC 595 điều khiển được tới 8 LED). Để điều khiển nhiều LED, chúng ta chỉ việc mắc thêm các bé 595 vào thôi, mắc như thế nào chúng ta sẽ biết ngay thôi 😊.

Bạn copy đoạn code sau và dán vào bên trong Arduino IDE, trong này, tôi có hướng dẫn bạn shiftOut để thực hiện lại quá trình đèn LED như lúc chúng ta sử dụng 8 chân digital.

```
/*
shiftOut với 8 LED bằng 1 IC HC595
*/
//chân ST_CP của 74HC595
int latchPin = 8;
//chân SH_CP của 74HC595
int clockPin = 12;
//Chân DS của 74HC595
int dataPin = 11;

//Trạng thái của LED, hay chính là byte mà ta sẽ gửi qua shiftOut
byte ledStatus;
void setup() {
  //Bạn BUỘC PHẢI pinMode các chân này là OUTPUT
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}
```

```

void loop() {
  /*
    Trong tin học, ngoài các phép +, -, *, / hay % mà bạn đã biết trên hệ cơ số 10.
    Thì còn có nhiều phép tính khác nữa. Và một trong số đó là Bit Math (toán bit) trên hệ cơ
    số 2.
    Để hiểu những gì tôi viết tiếp theo sau, bạn cần có kiến thức về Bit Math.
    Để tìm hiểu về Bit Math, bạn vào mục Tài liệu tham khảo ở bảng chọn nằm phía trên cùng
    trang web và chạy xuống khi bạn kéo chuột trên trang Arduino.VN

  */
  //Sáng tuần tự
  ledStatus = 0; //mặc định là không có đèn nào sáng hết (0 = 0b00000000)
  for (int i = 0; i < 8; i++) {
    ledStatus = (ledStatus << 1) | 1; //Đẩy toàn bộ các bit qua trái 1 bit và cộng bit có giá
    trị là 1 ở bit 0

    /**
     * Bắt buộc phải có để shiftOut
     */

    digitalWrite(latchPin, LOW); //các đèn LED sẽ không sáng khi bạn digital LOW

    //ShiftOut ra IC
    shiftOut(dataPin, clockPin, MSBFIRST, ledStatus);

    digitalWrite(latchPin, HIGH); //các đèn LED sẽ sáng với trạng thái vừa được cập nhập

    /**
     * Kết thúc bắt buộc phải có
     */

    delay(500); // Dừng chương trình khoảng 500 mili giây để thấy các hiệu ứng của đèn LED
  }
}

```

```
//Tắt tuần tự
for (int i = 0;i<8;i++) {
    ledStatus <= 1; //Đẩy tất cả các bit qua bên trái 1 bit
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, ledStatus);
    digitalWrite(latchPin, HIGH);
    delay(500);
}
}
```

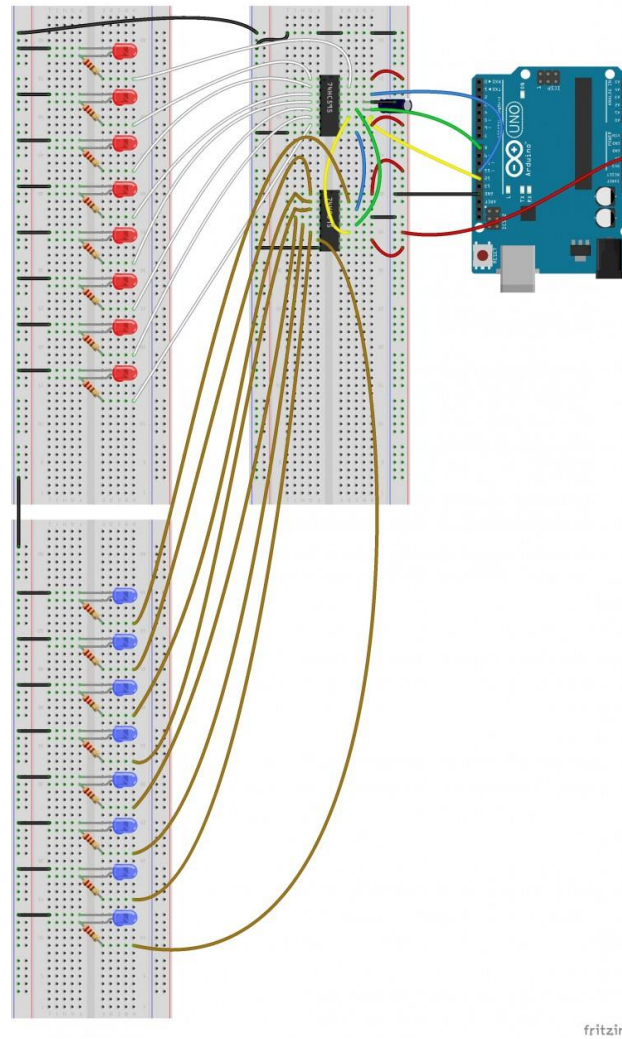
Bạn thấy đấy, đoạn bắt buộc phải có để shiftOut chúng ta có thể thay thế bằng một hàm tự tạo `void`. Ngoài ra, bạn có thể tải về bản mô phỏng Proteus.

Thứ năm, điều khiển nhiều đèn LED (số lượng LED chỉ phụ thuộc vào mức độ chịu khó của bạn) bằng Arduino với nhiều IC HC 595

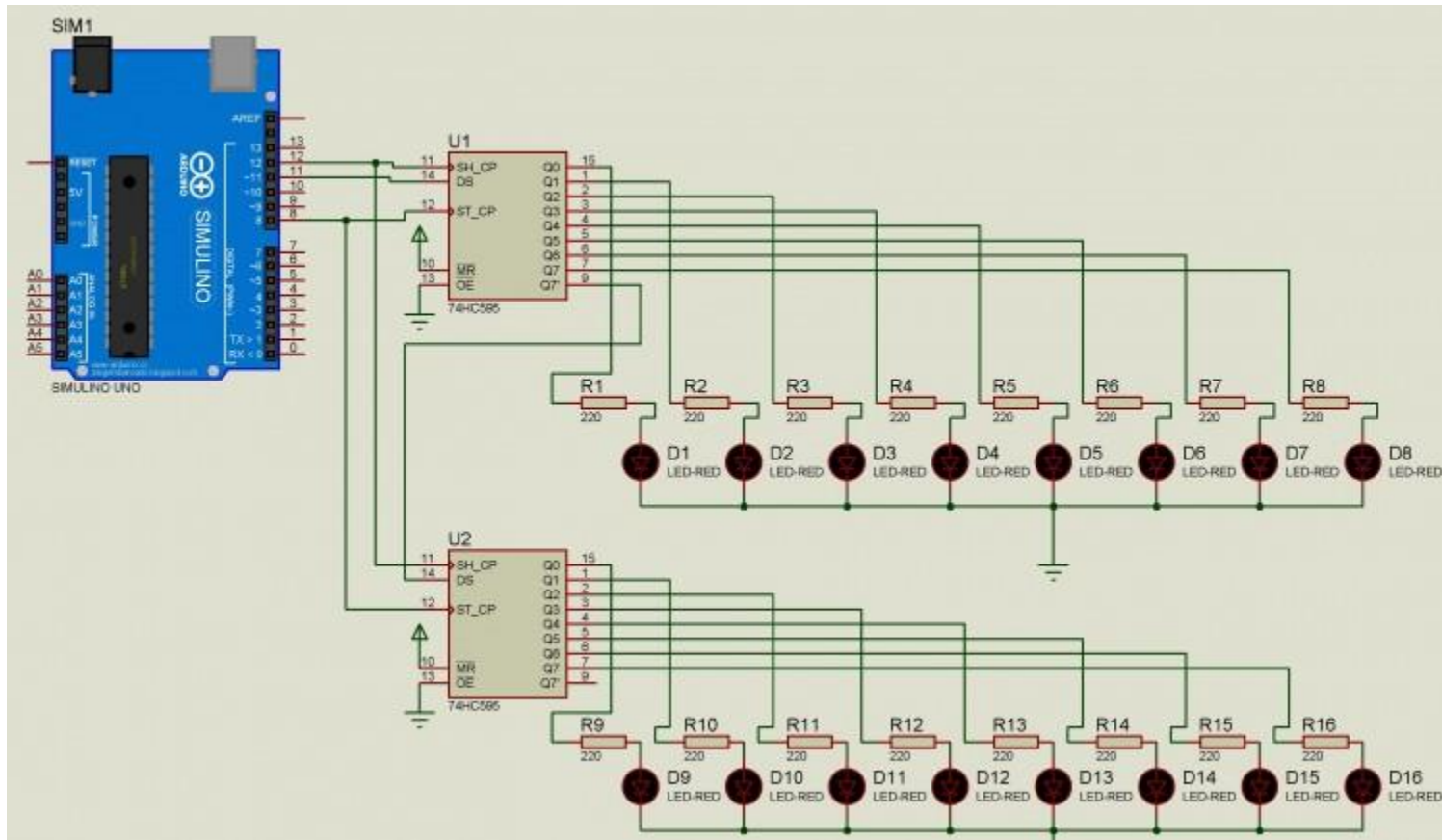
Như bạn đã biết, cứ mỗi một con HC 595 thì chúng ta sẽ điều khiển được tối đa 8 LED (mà tôi khuyên bạn nên thiết kế với số LED chia hết cho 8). Vậy theo lý thuyết và thực tế là như vậy, chỉ cần nhiều 595 là chúng ta làm được nhiều LED hơn. Cái khó, mà không, cũng không khó, chỉ cần hiểu sơ sơ là bạn sẽ làm được LED như mấy tấm biển quảng cáo rồi! Trước hết, bạn cần đọc nguyên lý mà tôi đã soạn để bạn - một người mới nghiên cứu có thể hiểu và làm được.

1. Chỉ IC HC595 đầu tiên mắc như những gì nói ở [trên](#), kể từ IC HC595 thứ hai trở đi, các bạn mắc như những gì tôi nói ở đây và theo trình tự như sau: IC 2 nối với IC 1, IC 3 nối với IC 2, IC 4 nối với IC 3, và cứ như thế IC mắc sau cùng sẽ mắc vào IC vừa được mắc trước đó.
2. Cứ mỗi lần shiftOut, thì byte đầu tiên sẽ đến IC HC 595 cuối cùng, byte thứ hai sẽ đến IC HC595 thêm trước IC HC595 đó và cứ như thế nhé.
3. Cứ mỗi lần shiftOut ra trạng thái các LED, bạn buộc phải shiftOut hết cho đến shiftOut đầu tiên (vì shiftOut lần 1 sẽ tới cái IC cuối cùng). Nếu chưa rành, bạn hãy sử dụng hàm `void shiftOutHC595` để shiftOut mỗi khi bạn muốn cập nhật trạng thái của mình.
4. Một đừng nản, hãy kiên trì!

Nào, chúng ta cùng lắp mạch thôi, những IC HC 595 thêm mới chúng ta vẫn lắp như mục [Thứ 4](#) ở trên. Nhưng có một thay đổi nhỏ xíu, đó là thay vì nối chân số 14 của HC 595 với chân digital 11 của Arduino mà chúng ta phải nối vào chân số 9 của IC 595 trước đó. Bạn hãy tham khảo mạch sau để rõ hơn.



hoặc bạn có thể thích cách mắc này hơn



Việc cuối cùng bạn cần làm là dán đoạn code này vào Arduino IDE, sau đó đọc từng dòng tôi comment trong này nhé 😈 !

```
1. /*
2. shiftOut với 8 LED bằng 1 IC HC595
3. */
4. //chân ST_CP của 74HC595
5. int latchPin = 8;
```

```

6. //chân SH_CP của 74HC595
7. int clockPin = 12;
8. //Chân DS của 74HC595
9. int dataPin = 11;
10.
11. //Trạng thái của LED, hay chính là byte mà ta sẽ gửi qua shiftOut
12. const int HC595_COUNT = 2; //Nếu bạn dùng nhiều hơn thì thay bằng một số lớn hơn 2.
13. byte ledStatus[HC595_COUNT] = {0};
14. void setup() {
15.     //Bạn BUỘC PHẢI pinMode các chân này là OUTPUT
16.     pinMode(latchPin, OUTPUT);
17.     pinMode(clockPin, OUTPUT);
18.     pinMode(dataPin, OUTPUT);
19. }
20.
21. void fillValueToArray(byte value) {
22.     for (int i = 0; i < HC595_COUNT; i += 1) {
23.         ledStatus[i] = value;
24.     }
25. }
26.
27. void shiftOutHC595(int dataPin, int clockPin, byte ledStatus[]) {
28.     digitalWrite(latchPin, LOW);
29.
30.     for (int i = 0; i < HC595_COUNT; i++) {
31.         shiftOut(dataPin, clockPin, LSBFIRST, ledStatus[i]); // Chúng ta dùng LSBFIRST thay cho
        MSBFIRST là có lý do của nó, cái này tôi không biết trình như thế nào để bạn hiểu rõ nhất
32.         // Tốt nhất bạn hãy thay LSBFIRST thành MSBFIRST để rõ ràng những gì tôi muốn nói.
33.     }
34.
35.     digitalWrite(latchPin, HIGH);
36. }
37.
38. void loop() {
39.     /*
40.         Trong tin học, ngoài các phép +, -, *, / hay % mà bạn đã biết trên hệ cơ số 10.
41.         Thì còn có nhiều phép tính khác nữa. Và một trong số đó là Bit Math (toán bit) trên hệ cơ
        số 2.
42.         Để hiểu những gì tôi viết tiếp theo sau, bạn cần có kiến thức về Bit Math.

```

```

43.     Để tìm hiểu về Bit Math, bạn vào mục Tài liệu tham khảo ở bảng chọn nằm phía trên cùng
        trang web và chạy xuống khi bạn kéo chuột trên trang Arduino.VN
44.     */
45.     //Sáng tuần tự
46.
47.     //vì ledStatus là một mảng vì vậy để mặc định tắt cả đèn tắt thì chúng ta phải for đến từng
        giá trị của mảng rồi đặt giá trị là 0.
48.     fillValueToArray(0);
49.     //Bật tuần tự
50.     for (int i = 0; i < HC595_COUNT; i++) {
51.         for (byte j=0;j<8;j++) {
52.             ledStatus[i] = (ledStatus[i] << 1) | 1;
53.             shiftOutHC595(dataPin,clockPin,ledStatus);
54.             delay(100); // Dừng chương trình khoảng 500 mili giây để thấy các hiệu ứng của đèn LED
55.         }
56.     }
57.
58.     //Tắt tuần tự
59.     for (int i = 0; i < HC595_COUNT; i++) {
60.         for (byte j=0;j<8;j++) {
61.             ledStatus[i] = (ledStatus[i] << 1);
62.             shiftOutHC595(dataPin,clockPin,ledStatus);
63.             delay(100); // Dừng chương trình khoảng 500 mili giây để thấy các hiệu ứng của đèn LED
64.         }
65.     }
66.
67.     /*
68.         Một số thuật toán nhấp nháy khác, tôi chia sẻ với các bạn đây, hãy làm một cái gì đấy tặng
        người thân yêu mình nhé.
69.         Cộng đồng của chúng ta không chỉ hướng tới Arduino, mà còn hướng tới cuộc sống tinh thần
        của các bạn nữa :)
70.     */
71.
72.     //Nhấp nháy nhanh
73.     for (byte k = 0; k < 20; k++) {
74.         fillValueToArray(0b10101010);
75.         shiftOutHC595(dataPin,clockPin,ledStatus);
76.         delay(50);
77.         fillValueToArray(0b01010101);

```

```

78.     shiftOutHC595(dataPin,clockPin,ledStatus);
79.     delay(50);
80. }
81.
82. //sáng 1 đèn rồi cách 1 đèn ko sáng rồi lại sáng những đèn chưa bật
83. fillValueToArray(0);
84. for (int i = 0; i < HC595_COUNT; i++) {
85.     for (byte j = 0;j<8;j += 2) {
86.         ledStatus[i] |= 1 << j;
87.         shiftOutHC595(dataPin,clockPin,ledStatus);
88.         delay(200);
89.     }
90. }
91. for (int i = 0; i < HC595_COUNT; i++) {
92.     for (byte j = 1;j<8;j += 2) {
93.         ledStatus[i] |= 1 << j;
94.         shiftOutHC595(dataPin,clockPin,ledStatus);
95.         delay(200);
96.     }
97. }
98.
99. //Tắt dần theo thứ tự trên
100. for (int i = HC595_COUNT - 1; i >= 0; i--) {
101.     for (int j = 7;j >= 0;j -= 2) {
102.         ledStatus[i] &= ~(1 << j);
103.         shiftOutHC595(dataPin,clockPin,ledStatus);
104.         delay(200);
105.     }
106. }
107. for (int i = HC595_COUNT - 1; i >= 0; i--) {
108.     for (int j = 6;j >= 0;j -= 2) {
109.         ledStatus[i] &= ~(1 << j);
110.         shiftOutHC595(dataPin,clockPin,ledStatus);
111.         delay(200);
112.     }
113. }
114.
115. /// Hãy khám phá thế giới lập trình này nhé :)
116. }

```

Lời kết

Thật là mệt mỏi đúng không nào, chạy ra ngoài và tự thưởng cho mình một lon nước ngọt nhé!

Tôi cũng như bạn, đều bắt đầu với một chiếc ly rỗng, dần dần tôi tự đổ nước vào chiếc ly khi cộng đồng Arduino Việt Nam là một cái gì đó rất mơ hồ. Còn bạn, bạn đã có sự hỗ trợ của cộng đồng Arduino Việt Nam thành hình ngày nay, vì vậy, đừng nản lòng, hãy chiến đấu với bản thân mình!