**Sử dụng IC MAX7219 để điều khiển LED Ma trận 8x8**

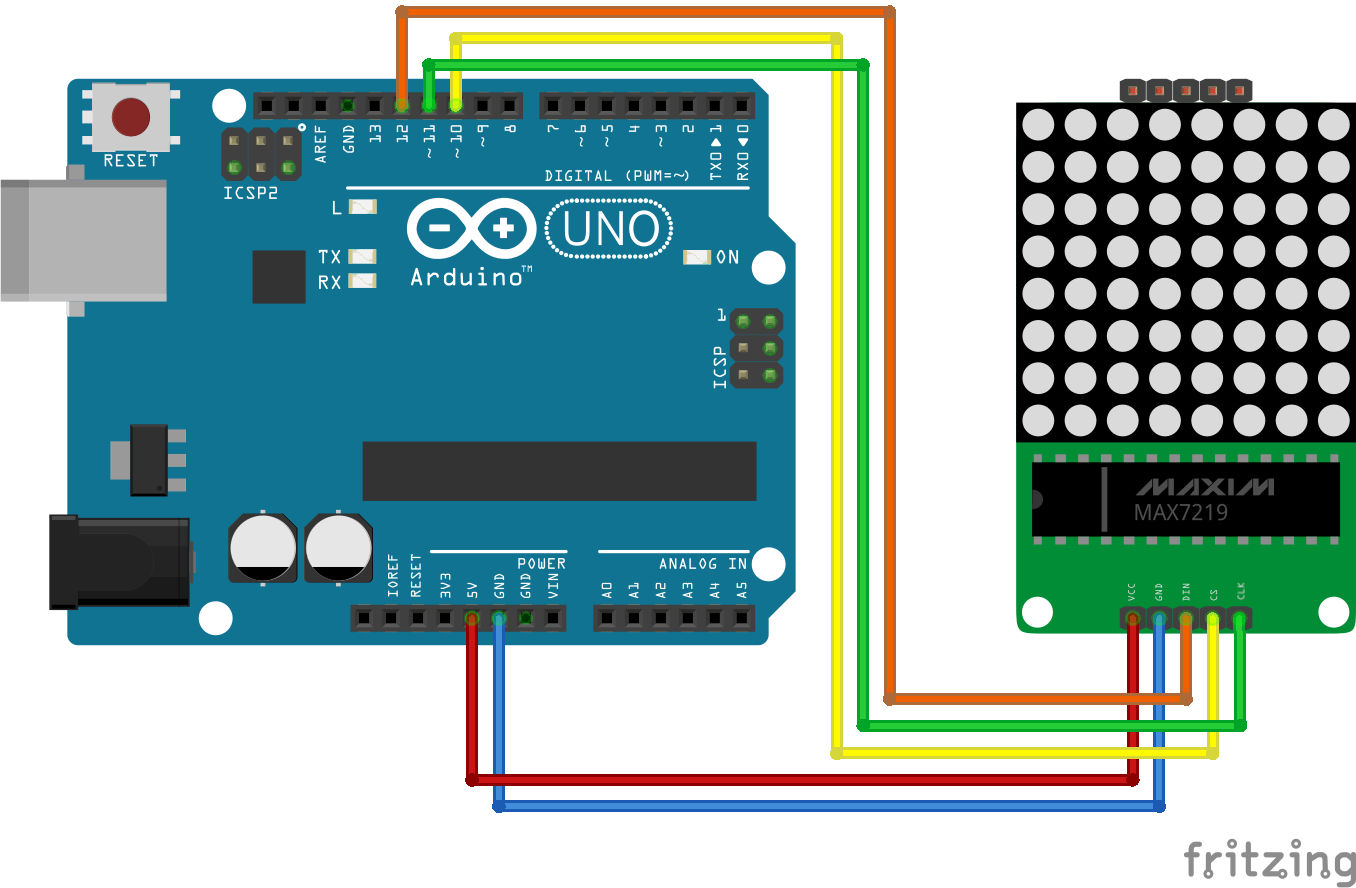
**Giới thiệu**

Hiển thị LED 7 thanh và LED ma trận là những bài học rất cơ bản cho những người tìm hiểu về Arduino. Trên công đồng arduino.vn cũng đã có những bài viết về vấn đề này. Trong quá trình tìm hiểu các tài liệu nước ngoài, mình thấy người ta thường sử dụng IC MAX7219 để điều khiển cho cả LED ma trận và LED 7 thanh. Thấy trên cộng đồng chưa có bài viết nào về MAX7219 nên mình thực hiện bài viết này, với hy vọng có thể đơn giản hơn việc lập trình hiển thị LED ma trận. Việc hiển thị LED 7 thanh sẽ được trình bày trong các bài viết sau[.](http://arduino.vn/bai-viet/941-su-dung-ic-max7219-de-dieu-khien-led-ma-tran-8x8-shiftout-16-bit)

**Phần cứng**

* Arduino UNO
* Module điều khiển LED MATRIX- Max7219
* Dây cắm bus đực - cái

**Nối mạch**

[](http://k3.arduino.vn/img/2016/07/03/0/2608_81210923-1467504340-0-12.jpg)

**Sơ đồ nối dây**

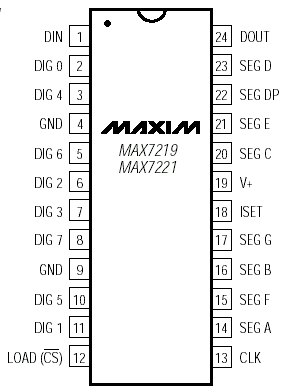
|  |  |
| --- | --- |
| **Module LED MATRIX 7219** | **ARDUINO UNO R3** |
| VCC | 5V |
| GND | GND |
| DIN | D12 |
| CS | D10 |
| CLK | D11 |

**IC MAX7219**

IC MAX7219 được hãng MAXIM thiết kế và sản xuất, thuận tiện sử dụng để điều khiển LED ma trận và LED 7 thanh (1 chip có thể điều khiển LED ma trận 8x8 hoặc LED 7 thanh 8 chữ số) và chỉ cần 1 điện trở để hạn dòng cho tất cả các led (so với các IC khác hầu như phải có 1 điện trở cho mỗi led). Trên mỗi chip đã được tích hợp bộ giải mã BCD, mạch quét dồn kênh, thanh ghi dịch, ...

Giống như với 74HC595, để điều khiển LED ma trận ta sử dụng MAX7219 hoạt động giống như một thanh ghi dịch khi bạn cần nhập dữ liệu vào nối tiếp theo từng bit. Tổng cộng có 16 bit được nhập vào tại một thời điểm. MAX7219 chỉ cần 3 chân từ Arduino để kết nối với các chân DIN, LOAD và CLK của IC.

Sơ đồ chân của MAX7219 được mô tả như hình dưới.

[](http://k3.arduino.vn/img/2016/07/03/0/2607_81210923-1467488817-0-max7219.gif)

Chân LOAD được kéo xuống LOW và bit đầu tiên của dữ liệu được đưa vào chân DIN, trên sườn lên của xung CLK, các bit tại chân DIN được đưa vào thanh ghi bên trong chip. Sau đó xung CLK xuống LOW và các bit tiếp theo được thiết lập tại chân DIN trước khi lặp lại quá trình. Sau khi 16 bit được đưa vào thanh ghi, chân LOAD được đưa lên HIGH để chốt dữ liệu vào. Chân DOUT (24) được sử dụng nếu có từ 2 chip MAX7219 trở lên. DOUT của chip thứ nhất kết nối DIN của chip thứ hai và cứ như vậy. Dữ liệu được đưa ra khỏi chân DOUT trên sườn xuống của xung CLK.

Các bạn có thể tham khảo thêm datasheet của MAX7219 để hiểu rõ hơn về nó.

**Lập trình**

MAX7219 rất thuận tiện cho việc hiển thị LED ma trận nên người ta đã tạo sẵn thư viện. Các bạn có thể download tại [đây](https://github.com/wayoda/LedControl/archive/master.zip).

Trong thư viện có một số hàm thông dụng như: setLed (led sáng tại điểm); setRow (led sáng tại hàng), setColumn (led sáng tại cột); clearDisplay (tất cả led đều bật/tắt). Do đó ta có thể điều khiển LED ma trận theo ý muốn một các khá dễ dàng.

Trong bài viết này, mình sẽ lập trình hiển thị ký tự và chạy chuỗi ký tự để các bạn có thể tham khảo.

1. #include "LedControl.h"// thêm thư viện
2. LedControl matrix = LedControl(2, 4, 3, 1);
3. // Chân 12 nối với chân DataIn
4. // Chân 11 nối với chân CLK
5. // Chân 10 nối với chân LOAD
6. // Sử dụng 1 IC MAX7219
7. void setup() {
8. matrix.shutdown(0, false); // Bật hiển thị
9. matrix.setIntensity(0, 15); // Đặt độ sáng lớn nhất
10. matrix.clearDisplay(0); // Tắt tất cả led
11. }
12. // Thiết lập mã cho các ký tự
13. byte A[56] = {
14. 0x00,0x3F,0x7F,0xA4,0xA4,0x7F,0x3F,0x00, // A
15. 0x00,0xFF,0xFF,0x98,0x94,0x92,0x61,0x00, // R
16. 0x00,0xFF,0xFF,0x81,0x81,0x7E,0x3C,0x00, // D
17. 0x00,0xFE,0xFF,0x01,0x01,0xFF,0xFE,0x00, // U
18. 0x00,0x81,0x81,0xFF,0xFF,0x81,0x81,0x00, // I
19. 0x00,0xFF,0xC0,0x30,0x0C,0x03,0xFF,0x00, // N
20. 0x00,0x7E,0xFF,0x81,0x81,0xFF,0x7E,0x00 // O
21. };
22. // Chương trình con chạy chuỗi ký tự
23. void scroll() {
24. matrix.clearDisplay(0);
25. int pos = 8;
26. for (int j = pos; j > -56; j--) { // Vòng lặp thay đổi vị trí
27. for (int i = 0; i < 56; i++) { // Vòng lăp để hiển thị ký tự
28. matrix.setRow(0, i + j, A[i]);
29. }
30. delay(100);
31. }
32. }
34. // Chương trình con hiển thị từng ký tự
35. void show() {
36. matrix.clearDisplay(0);
37. for (int i = 0; i < 8; i++) matrix.setRow(0, i, A[i]); //Hiển thị từng hàng để được ký tự A
38. delay(200);
39. for (int i = 0; i < 8; i++) matrix.setRow(0, i, A[i + 8]);
40. delay(200);
41. for (int i = 0; i < 8; i++) matrix.setRow(0, i, A[i + 16]);
42. delay(200);
43. for (int i = 0; i < 8; i++) matrix.setRow(0, i, A[i + 24]);
44. delay(200);
45. for (int i = 0; i < 8; i++) matrix.setRow(0, i, A[i + 32]);
46. delay(200);
47. for (int i = 0; i < 8; i++) matrix.setRow(0, i, A[i + 40]);
48. delay(200);
49. for (int i = 0; i < 8; i++) matrix.setRow(0, i, A[i + 48]);
50. delay(200);
51. }
52. void loop() {
53. scroll();
54. show();
55. }