

## MATRIX

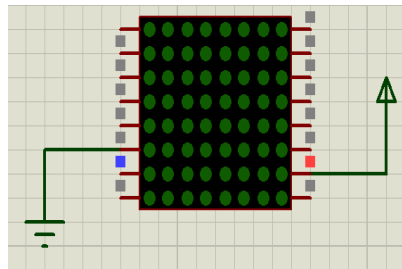
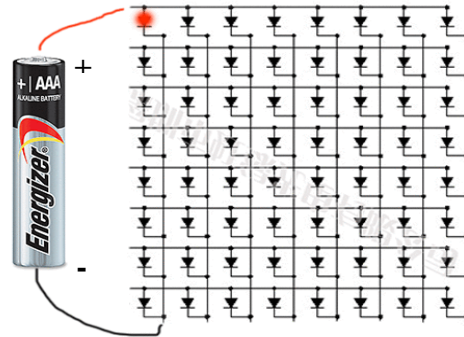
### 1. Cấu tạo Led matrix

- Led Matrix được cấu tạo từ 8 Led đơn ( 8 chân kết nối đến Cột và 8 chân kết nối đến Hàng) => cần 8 chân của vi điều khiển để kết nối đến hàng, 8 chân vi điều khiển kết nối đến cột
- Để 1 Led sáng: cấp điện áp VCC và GND vào 2 chân của Led đó
- Trong 1 thời điểm chỉ điều khiển được 8 Led

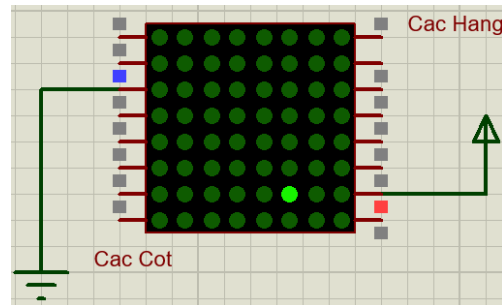
(Tham khảo Ví dụ 1)

- Chú ý: để Test xem HÀNG và CỘT
  - o Quy định (+): HÀNG
  - o Quy định (-): CỘT

Ví dụ: để kiểm tra đâu là chân HÀNG, đâu là chân CỘT trong Matrix, nối với 1 chân nguồn(+); và 1 chân với nguồn (-). Nếu đèn SÁNG, đã xác định đúng HÀNG và CỘT

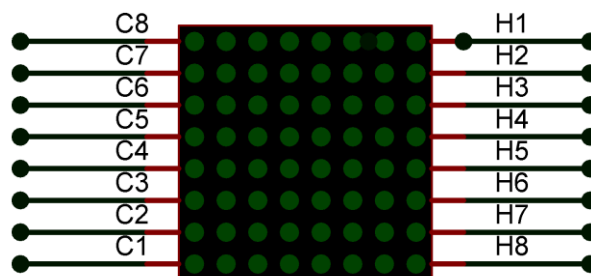


(a) Không nối đúng HÀNG và CỘT



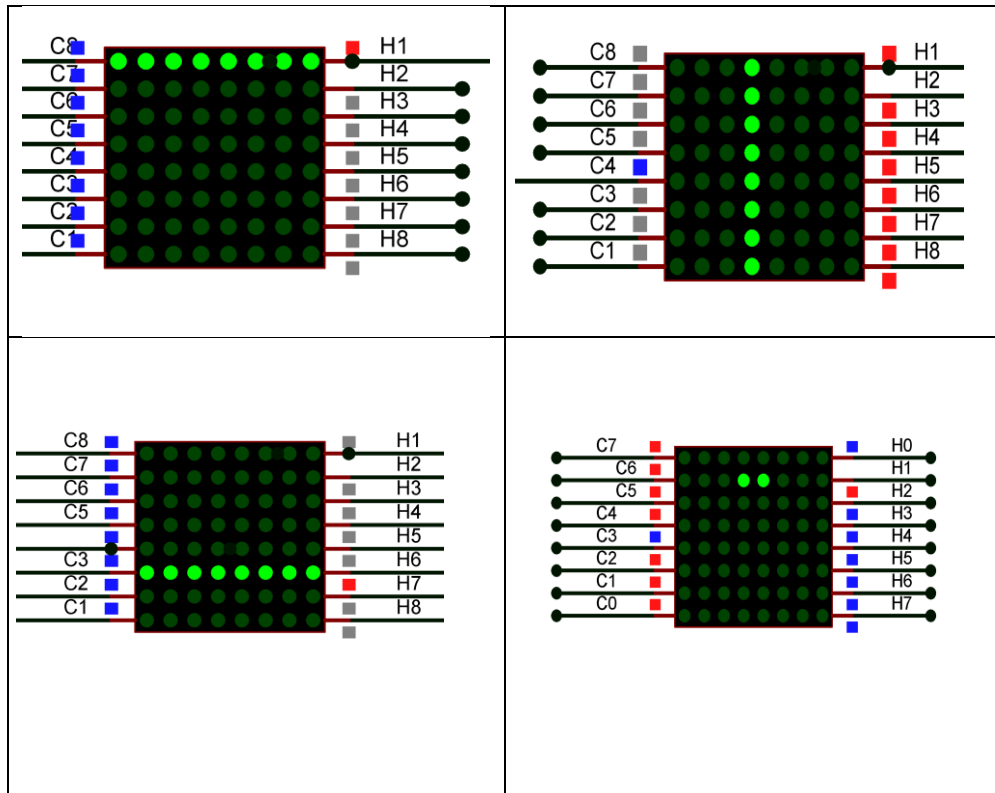
(b) Nối đúng HÀNG và CỘT

Nguyên tắc gán cho HÀNG và CỘT như sau



Ví dụ: Hãy xác định các HÀNG và CỘT để Led Matrix sáng hàng các vị trí như sau

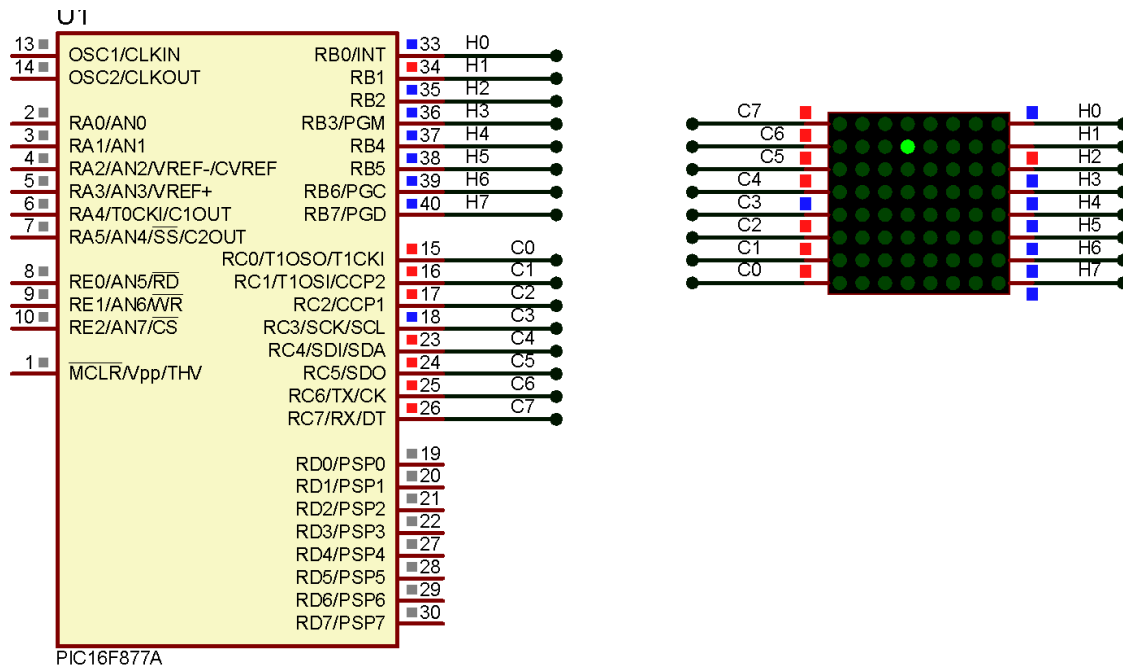
--	--



## 2. Nguyên lý quét Led matrix

- Chốt tín hiệu vào từng hàng và từng cột
- Duy trì trạng thái trong 1 khoảng thời gian đủ để mắt người cảm nhận được (khoảng 1ms)
- Tắt hết các Led
- Chốt tín hiệu vào các hàng và các cột tiếp theo

Ví dụ: Viết chương trình để hiển thị ký tự “A” lên matrix



Code:

**VỪA DÀI VỪA DẠI, AI CHƯA HIỂU QUÉT MATRIX THÌ ĐỌC , AI HIỂU RỒI CANCEL NHÉ!**

```
#include <16f877a.h>
#use delay(clock=20M)
void main()
{
    while(TRUE)
    {
        output_B(0b00000010);
        output_C(0b11110111);
        delay_ms(1);

        output_B(0b00000000); // Tat het cac Led

        output_B(0b00000010);
        output_C(0b11101111);
        delay_ms(1);

        output_B(0b00000000);

        output_B(0b00000100);
```

```
    output_C(0b11111011);  
    delay_ms(1);  
  
    output_B(0b00000000);  
  
    output_B(0b00000100);  
    output_C(0b11011111);  
    delay_ms(1);  
  
    output_B(0b00000000);  
  
    output_B(0b00001000);  
    output_C(0b10111101);  
    delay_ms(1);  
  
    output_B(0b00000000);  
  
    output_B(0b00010000);  
    output_C(0b10000001);  
    delay_ms(1);  
  
    output_B(0b00000000);  
  
    output_B(0b00100000);  
    output_C(0b10111101);  
    delay_ms(1);  
  
    output_B(0b00000000);  
  
    output_B(0b01000000);  
    output_C(0b10111101);  
    delay_ms(1);  
  
    output_B(0b00000000);  
  
    output_B(0b10000000);  
    output_C(0b10111101);  
    delay_ms(1);  
}  
}
```

## AI NẮM ĐƯỢC PHƯƠNG PHÁP QUÉT RỒI ĐỪNG BỎ QUA CODE NÀY NHÉ

Quét Matrix theo Hàng thông qua cách xây dựng Mảng cho Hàng và Mảng cho Cột

```
#include <16f877a.h>
#include delay(clock=20M)
unsigned int8 Ma_Hang[8]={0x02, 0x04, 0x08, 0x10, 0x20,
0x40, 0x80 };
unsigned int8 Ma_Cot[8]={0xE7, 0xDB,0XBD , 0XBD, 0x81,
0XBD, 0XBD};
int8 i;
void main()
{
    while(true)
    {
        for(i=0; i<=6;i++)
        {
            output_B(0x00);
            output_B(ma_hang[i]);
            output_c(ma_cot[i]);
            delay_ms(1);
        }
    }
}
```



(Tham khảo Ví dụ 1 và Ví dụ 2)

### 3. Ví dụ

- a. Cho Cột C<sub>0</sub> sáng, các cột còn lại tắt

```
#include <16f877a.h>
```

```
#use delay(clock=20M)
```

```
unsigned int8 bang_ma[8]={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
```

```
unsigned int8 i;
```

```
// cot C0 SANG cac cot con lai TAT
```

```
// theo PP quet theo hang, Hang=1 còn
```

```
COT=0;
```

```
// de COT C7 sang thì COT=0000.0000
```

```
// HANG=1111.1110
```

```
void main()
```

```
{
    while(TRUE)
```

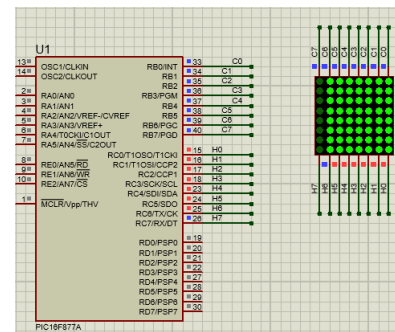
```
{
    output_B(0x00);
```

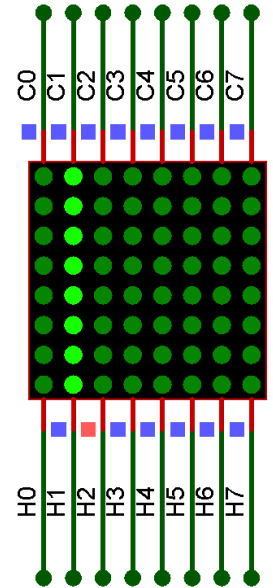
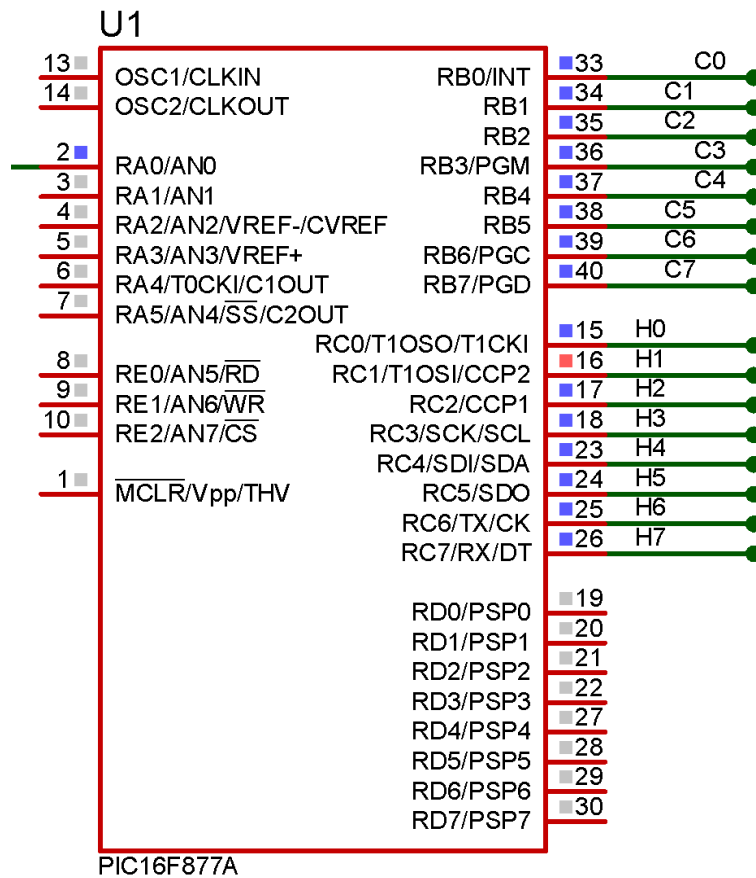
```
    output_C(0x7F);
```

```
}
```

```
}
```

- b. Cho các cột lần lượt sáng từ Cột C<sub>0</sub> đến Cột C<sub>7</sub>





```
#include <16f877a.h>
#use delay(clock=20M)
unsigned int8
bang_ma[8]={0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
unsigned int8 i;
void main()
{
    while(TRUE)
    {
        for(i=0; i<=7; i++)
        {
            output_C(bang_ma[i]);
            output_B(0x00);
            delay_ms(500);
        }
    }
}
```

c. Cho sáng chữ cái đầu của tên