Dot Matrix Display Interfacing With PIC16F877A Microcontroller

Theory:

Dot Matrix Display:

A Dot-Matrix Display is a display device which contains light emitting diodes aligned in the form of matrix. Dot matrix displays are used in applications where symbol, graphic, characteristics, alphabets numbers are need to be displayed in static as well as scrolling motion. A typical dot matrix display is shown in figure below:

Figure: A typical dot matrix display is using in lift.

Figure: Another dot matrix display generally used as banner or digital indication board

Types of Matrix Display:

Dot matrix Display is manufactured invarious dimensions like 5x7, 8x8, 16x8, 32x8, 64x64 and 128x64 where the numbers represent LED's in rows and columns. Also these displays comes in different colors such as Red, Green, Yellow, Blue, Orange and White.
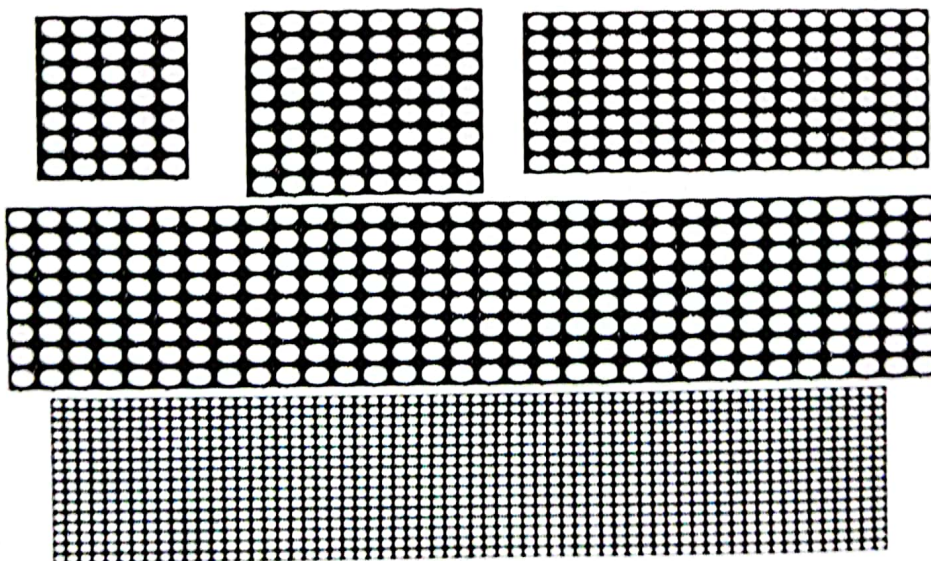

Figure: Types of Dot Matrix display

Construction of Matrix Display:

In Dot matrix display, multiple LED's are wired together in rows and columns. The matrix pattern is made either in row anode and column cathode or row cathode and column anode pattern. In row anode column cathode pattern, the entire row is anode while all columns serve as cathode and it is vice-versa in row cathode column anode pattern.
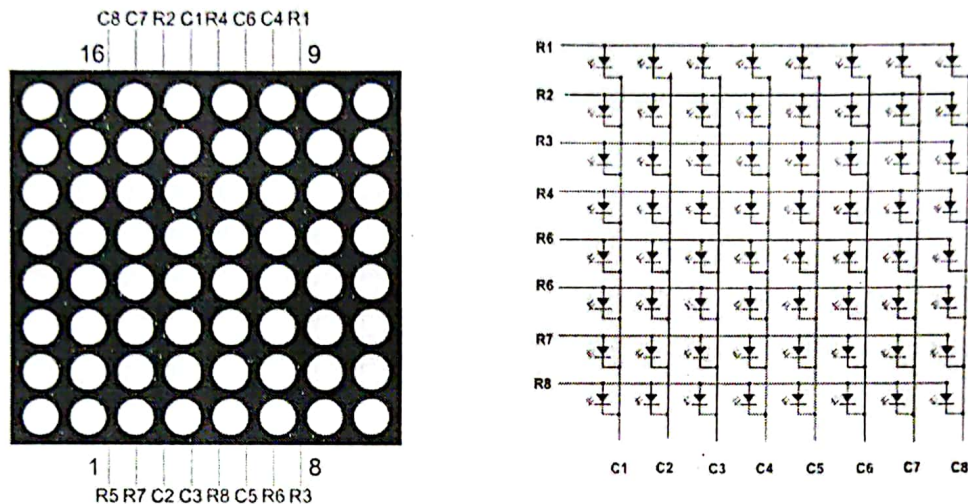

Figure: Construction of dot matrix display

How Dot Matrix Display Work:

In Dot Matrix display each LED can be control individually by controlling the current through each pair of column or row.
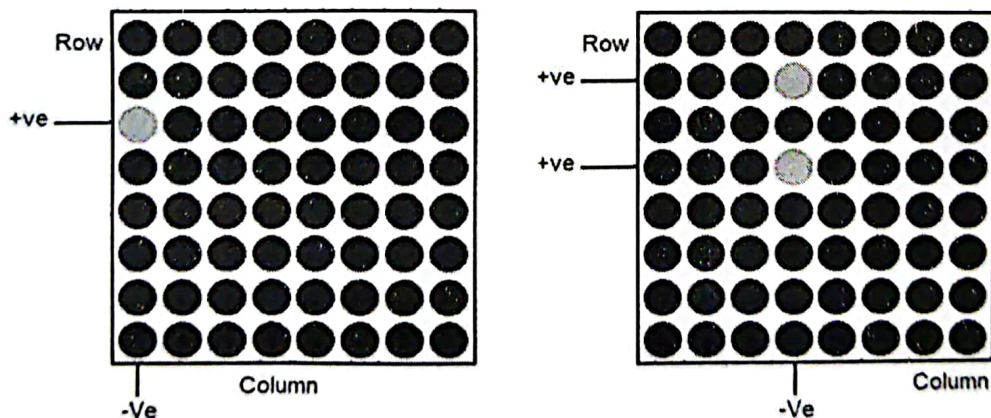

Figure: Working mechanism of dot matrix display

How To Display Character on Dot Matrix Display:

The method of displaying message on dot matrix display is same as seven segment multiplexing. The desire character can be displayed by switching ON/OFF a desire configuration of LEDs. The figure below shows the arrangement of LEDs in a typical 8x8 dot matrix display. Any individual LED or group of LEDs in the matrix can be activated by switching the required number of row and columns.
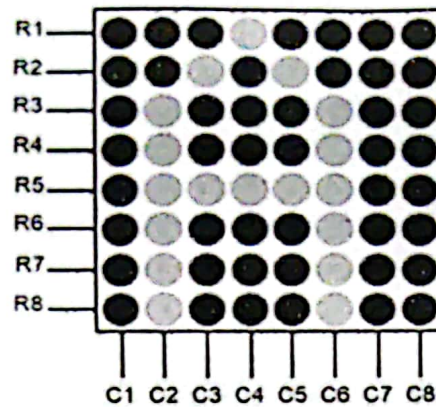
Figure: Character display in dot matrix display

How to Generate Code for Dot Matrix Display:

	1	2	3	4	5	6	7	8	Bin	HEX
1	0	1	1	0	0	1	1	0	0b00000000	0x00
2	0	1	1	0	0	1	1	0	0b11111111	0xff
3	0	1	1	0	0	1	1	0	0b11111111	0xff
4	0	1	1	1	1	1	1	0	0b00011000	0x18
5	0	1	1	1	1	1	1	0	0b00011000	0x18
6	0	1	1	0	0	1	1	0	0b11111111	0xff
7	0	1	1	0	0	1	1	0	0b11111111	0xff
8	0	1	1	0	0	1	1	0	0b00000000	0x00

Figure: Generation of HEX Code

Mikro C Code:

```
void MSDelay(unsigned char Time)
{
    unsigned char y,z;
    for(y=0;y<Time;y++)
    for(z=0;z<20;z++);
}
void main()
{
    TRISC = 0x00;
    TRISD = 0x00;
    while(1)
    {
        PORTD = 0x80;
        PORTC = 0x00;
        MSDelay(10);

        PORTD = 0x40;
        PORTC = 0xff;
        MSDelay(10);
```

```
PORTD = 0x20;
PORTC = 0xff;
MSDelay(10);

PORTD = 0x10;
PORTC = 0x18;
MSDelay(10);

PORTD = 0x08;
PORTC = 0x18;
MSDelay(10);

PORTD = 0x04;
PORTC = 0xff;
MSDelay(10);

PORTD = 0x02;
PORTC = 0xff;
MSDelay(10);

PORTD = 0x01;
PORTC = 0x00;
MSDelay(10);

    }
}
```
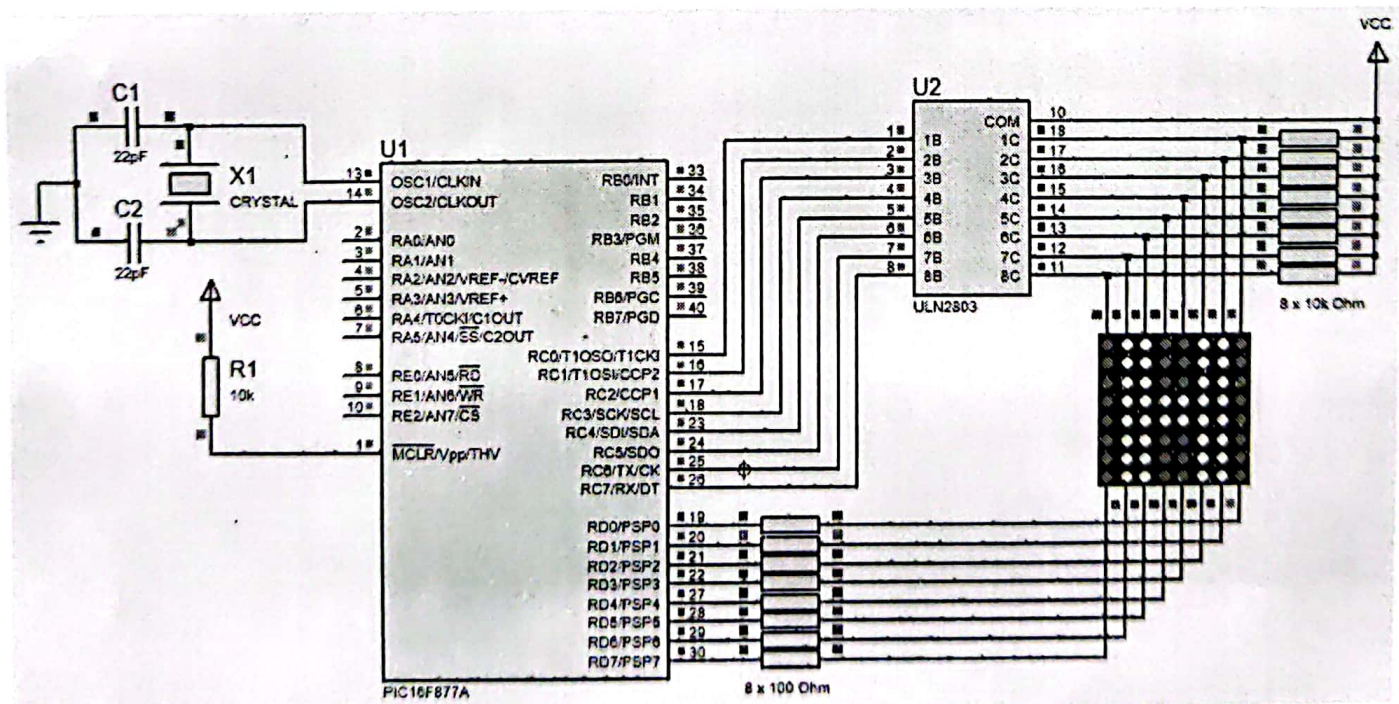
Circuit Diagram:

Figure: Proteus simulation of dot matrix LED Display