

(မိုင်း - <https://circuitdigest.com/microcontroller-projects/arduino-8x8-led-matrix>
<https://www.instructables.com/id/Dot-Matrix-With-MAX7219-Interfacing-With-Arduino/>
<https://www.brainy-bits.com/scroll-text-using-the-max7219-led-dot-matrix/>)

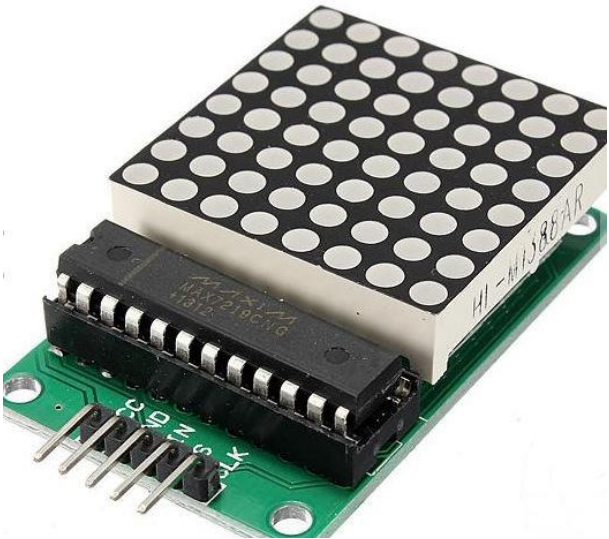
8x8 LED Dot Matrix MAX7219 Interfacing with Arduino (Tutorial 1)

MAX7219 chip ပါသော 8x8 LED Dot Matrix Module ကို အာဒွီနို နှင့် ချိတ်ဆက်အသုံးပြုခြင်း

ယခု ပရောဂျက်တွင် 8x8 LED matrix display module ကို အာဒွီနို Uno နှင့် ချိတ်ဆက်အသုံးပြုပါမည်။ 8x8 LED Matrix တွင် matrix ပုံစံထားသော LED မီးသီး ၆၄ လုံး ပါဝင်ပါသည်။ ထို့ကြောင့် LED matrix ဟုခေါ်ခြင်းဖြစ်ပါသည်။

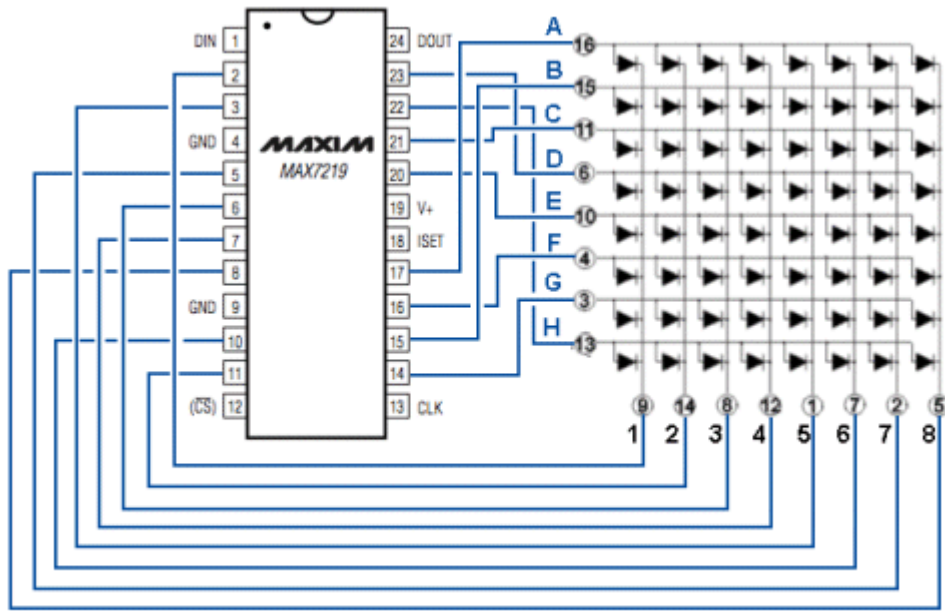
LED မီးသီး ၆၄ လုံးသုံးပြီး matrix ပုံစံ ချိတ်ဆက်ပေးခြင်းဖြင့် ယခုလိုမျိုး LED Matrixes များကို ပြုလုပ်နိုင်ပါသည်။ သို့သော် ချိတ်ဆက်ပေးရသည့် အချိန်မှာ ကြာမြင့်နိုင်ပါသည်။ ယခုခေတ်တွင် အောက်ပါပုံအတိုင်း ပိုမိုသပ်ရပ်ပြီး အဆင့်သင့်လာသော LED matrix module များ ကို အရွယ်အစားမျိုးစုံ ၊ အရောင်မျိုးစုံ ရွေးချယ် ဝယ်ယူရရှိနိုင်ပါသည်။

သာမန် LED matrix တွင် pin အရေအတွက် ၁၆ခု - common positive pin ၈ ခု နှင့် common negative pin ၈ ခု ပါဝင်ပါသည်။ LED matrix ကို အာဒွီနို Uno နှင့် တိုက်ရိုက် ချိတ်ဆက်အသုံးပြုရန် Uno အတွက် pin ၁၆ ခု သတ်မှတ် ထားရှိရန် လိုအပ်ပါသည်။ သို့သော် pin အရေအတွက်များသဖြင့် Uno နှင့် ချိတ်ဆက်အသုံးပြုရာတွင် processing တစ်ခုလုံးကို ရှုပ်ထွေးစေနိုင်ပါသည်။ ပိုမိုချောမွေ့သော processing အတွက် driver chip နှင့် တွဲဖက်ပါဝင်သော LED Matrix ကို အသုံးပြုရပါမည်။ Driver chip ပါဝင်သော Matrix မှာ အောက်ပါပုံ အတိုင်း ဖြစ်ပါသည်။



8x8 Dot Matrix Module

The MAX7219 Driver



Connecting the 8x8 Display to the MAX7219 chip

MAX72XX စီးရီး chip များအကြောင်းကို အသေးစိတ်ဖတ်ရှုလိုပါက

<https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf> ဆီ သို့သွားရောက်ပါ။

LED Matrix module များတွင် နောက်ထပ် LED Matrix module များ နှင့် series အလိုက် ထပ်မံ ချိတ်ဆက်အသုံးပြုနိုင်သော feature တစ်ခုလည်း ပါဝင်ပါသည်။ ထိုသို့ Module များကိုပေါင်းစပ်၍ Display တစ်ခုတည်း အနေဖြင့် program ထည့်သွင်း အသုံးပြုနိုင်ပါသည်။ စကားလုံး၊ သင်္ကေတ၊ ရုပ်ပုံ အစရှိသည်တို့ကို မျက်နှာပြင်ကျယ် display တစ်ခုတည်းတွင် တစ်ပြိုင်တည်း ပြသနိုင်ပါသည်။ LED Matrix module ဘယ်နှစ်ခုကိုပေါင်းစပ် အသုံးပြုထားသလဲ ဆိုသော အချက် ကို Control board တွင် သတ်မှတ်ဖော်ပြပေးရန်လိုအပ်ပါသည်။

ယခု module တွင် pin ၅ခု ပါဝင်ပါသည်။ GND နှင့် VCC pin အပြင် အာဒီနို Uno မှ module ဆီသို့ data များ ဖြတ်သန်းသွားလာရန် အတွက် အခြား pin သုံးခုပါဝင်ပါသည်။ ယင်း pin သုံးခုမှာ DIN / DATAIN (Data Receiving Pin) ၊ CLK / CLOCK (Clock pin) နှင့် CS / CHIPSELECT (Command receiving pin) တို့ဖြစ်ပါသည်။

ပရောဂျက်တွင် လိုအပ်သော ပစ္စည်းများ

- အာဒွီနို Uno
 - MAX7219 chip ပါဝင်သော 8x8 Dot Matrix Module
 - Breadboard
 - Jumper Wire
-

Pin Configuration

- Vcc to 5V Pin of Arduino.
 - Gnd to Gnd Pin of the Arduino.
 - DIN to Digital Pin 11 of the Arduino.
 - CS to Digital Pin 10 of the Arduino.
 - CLK to Digital Pin 13 of the Arduino.
-

Example Code

Scrolling Text နှင့် ပတ်သက်သော code မှာ အောက်ပါအတိုင်း ဖြစ်ပါသည်။

အာဒွီနို ထဲသို့ code မသွင်းခင် အရင်ဆုံး သက်ဆိုင်ရာ library ကို Arduino IDE ထဲသို့ ထည့်သွင်းပေးရပါမည်။

Library ကို download ဆွဲရန် - <https://code.google.com/archive/p/arduino-maxmatrix-library/downloads>

အောက်ပါ Code မှာ scrolling text ဟုခေါ်သော example code ဖြစ်ပါသည်။

```
// maxmatrix-disp-scroll-text-7219
// based on

// https://code.google.com/p/arduino-maxmatrix-library/wiki/Example_Display_Scrolling_Text

#include

#include

#define maxDisplays 2 // Number of MAX7219's in use.

byte Buf7219[7]; // "width,height,data[5]" single character buffer.

const int data = 11; // DIN or MOSI
```

```

const int load = 10;    // CS

const int clock = 13;  // SCK

MaxMatrix m(data, load, clock, maxDisplays);

// Data array is stored in program memory (see memcpy_P for access).

// Parameters are width, height, character data...

// There is a speed improvement for characters with height 8 bits see lib.

PROGMEM const unsigned char CH[] = {

3, 8, B0000000, B0000000, B0000000, B0000000, B0000000, // space

1, 8, B1011111, B0000000, B0000000, B0000000, B0000000, // !

3, 8, B0000011, B0000000, B0000011, B0000000, B0000000, // "

5, 8, B0010100, B0111110, B0010100, B0111110, B0010100, // #

4, 8, B0100100, B1101010, B0101011, B0010010, B0000000, // $

5, 8, B1100011, B0010011, B0001000, B1100100, B1100011, // %

5, 8, B0110110, B1001001, B1010110, B0100000, B1010000, // &

1, 8, B0000011, B0000000, B0000000, B0000000, B0000000, // '

3, 8, B0011100, B0100010, B1000001, B0000000, B0000000, // (

3, 8, B1000001, B0100010, B0011100, B0000000, B0000000, // )

5, 8, B0101000, B0011000, B0001110, B0011000, B0101000, // *

5, 8, B0001000, B0001000, B0111110, B0001000, B0001000, // +

2, 8, B10110000, B1110000, B0000000, B0000000, B0000000, // ,

4, 8, B0001000, B0001000, B0001000, B0001000, B0000000, // -

2, 8, B1100000, B1100000, B0000000, B0000000, B0000000, // .

```

4, 8, B1100000, B0011000, B0000110, B0000001, B0000000, // /
4, 8, B0111110, B1000001, B1000001, B0111110, B0000000, // 0
3, 8, B1000010, B1111111, B1000000, B0000000, B0000000, // 1
4, 8, B1100010, B1010001, B1001001, B1000110, B0000000, // 2
4, 8, B0100010, B1000001, B1001001, B0110110, B0000000, // 3
4, 8, B0011000, B0010100, B0010010, B1111111, B0000000, // 4
4, 8, B0100111, B1000101, B1000101, B0111001, B0000000, // 5
4, 8, B0111110, B1001001, B1001001, B0110000, B0000000, // 6
4, 8, B1100001, B0010001, B0001001, B0000111, B0000000, // 7
4, 8, B0110110, B1001001, B1001001, B0110110, B0000000, // 8
4, 8, B0000110, B1001001, B1001001, B0111110, B0000000, // 9
2, 8, B01010000, B0000000, B0000000, B0000000, B0000000, // :
2, 8, B10000000, B01010000, B0000000, B0000000, B0000000, // ;
3, 8, B0010000, B0101000, B1000100, B0000000, B0000000, // <
3, 8, B0010100, B0010100, B0010100, B0000000, B0000000, // =
3, 8, B1000100, B0101000, B0010000, B0000000, B0000000, // >
4, 8, B0000010, B1011001, B0001001, B0000110, B0000000, // ?
5, 8, B0111110, B1001001, B1010101, B1011101, B0001110, // @
4, 8, B1111110, B0010001, B0010001, B1111110, B0000000, // A
4, 8, B1111111, B1001001, B1001001, B0110110, B0000000, // B
4, 8, B0111110, B1000001, B1000001, B0100010, B0000000, // C
4, 8, B1111111, B1000001, B1000001, B0111110, B0000000, // D
4, 8, B1111111, B1001001, B1001001, B1000001, B0000000, // E

4, 8, B1111111, B0001001, B0001001, B0000001, B0000000, // F
4, 8, B0111110, B1000001, B1001001, B1111010, B0000000, // G
4, 8, B1111111, B0001000, B0001000, B1111111, B0000000, // H
3, 8, B1000001, B1111111, B1000001, B0000000, B0000000, // I
4, 8, B0110000, B1000000, B1000001, B0111111, B0000000, // J
4, 8, B1111111, B0001000, B0010100, B1100011, B0000000, // K
4, 8, B1111111, B1000000, B1000000, B1000000, B0000000, // L
5, 8, B1111111, B0000010, B0001100, B0000010, B1111111, // M
5, 8, B1111111, B0000100, B0001000, B0010000, B1111111, // N
4, 8, B0111110, B1000001, B1000001, B0111110, B0000000, // O
4, 8, B1111111, B0001001, B0001001, B0000110, B0000000, // P
4, 8, B0111110, B1000001, B1000001, B10111110, B0000000, // Q
4, 8, B1111111, B0001001, B0001001, B1110110, B0000000, // R
4, 8, B1000110, B1001001, B1001001, B0110010, B0000000, // S
5, 8, B0000001, B0000001, B1111111, B0000001, B0000001, // T
4, 8, B0111111, B1000000, B1000000, B0111111, B0000000, // U
5, 8, B0001111, B0110000, B1000000, B0110000, B0001111, // V
5, 8, B0111111, B1000000, B0111000, B1000000, B0111111, // W
5, 8, B1100011, B0010100, B0001000, B0010100, B1100011, // X
5, 8, B0000111, B0001000, B1110000, B0001000, B0000111, // Y
4, 8, B1100001, B1010001, B1001001, B1000111, B0000000, // Z
2, 8, B1111111, B1000001, B0000000, B0000000, B0000000, // [
4, 8, B0000001, B0000110, B0011000, B1100000, B0000000, // backslash

2, 8, B1000001, B1111111, B0000000, B0000000, B0000000, //]
3, 8, B0000010, B0000001, B0000010, B0000000, B0000000, // hat
4, 8, B1000000, B1000000, B1000000, B1000000, B0000000, // _
2, 8, B0000001, B0000010, B0000000, B0000000, B0000000, // `
4, 8, B0100000, B1010100, B1010100, B1111000, B0000000, // a
4, 8, B1111111, B1000100, B1000100, B0111000, B0000000, // b
4, 8, B0111000, B1000100, B1000100, B0000000, B0000000, // c // JFM MOD.
4, 8, B0111000, B1000100, B1000100, B1111111, B0000000, // d
4, 8, B0111000, B1010100, B1010100, B0011000, B0000000, // e
3, 8, B0000100, B1111110, B0000101, B0000000, B0000000, // f
4, 8, B10011000, B10100100, B10100100, B01111000, B0000000, // g
4, 8, B1111111, B0000100, B0000100, B1111000, B0000000, // h
3, 8, B1000100, B1111101, B1000000, B0000000, B0000000, // i
4, 8, B1000000, B10000000, B10000100, B1111101, B0000000, // j
4, 8, B1111111, B0010000, B0101000, B1000100, B0000000, // k
3, 8, B1000001, B1111111, B1000000, B0000000, B0000000, // l
5, 8, B1111100, B0000100, B1111100, B0000100, B1111000, // m
4, 8, B1111100, B0000100, B0000100, B1111000, B0000000, // n
4, 8, B0111000, B1000100, B1000100, B0111000, B0000000, // o
4, 8, B11111100, B0100100, B0100100, B0011000, B0000000, // p
4, 8, B0011000, B0100100, B0100100, B11111100, B0000000, // q
4, 8, B1111100, B0001000, B0000100, B0000100, B0000000, // r
4, 8, B1001000, B1010100, B1010100, B0100100, B0000000, // s

```

3, 8, B0000100, B0111111, B1000100, B0000000, B0000000, // t
4, 8, B0111100, B1000000, B1000000, B1111100, B0000000, // u
5, 8, B0011100, B0100000, B1000000, B0100000, B0011100, // v
5, 8, B0111100, B1000000, B0111100, B1000000, B0111100, // w
5, 8, B1000100, B0101000, B0010000, B0101000, B1000100, // x
4, 8, B10011100, B10100000, B10100000, B1111100, B0000000, // y
3, 8, B1100100, B1010100, B1001100, B0000000, B0000000, // z
3, 8, B0001000, B0110110, B1000001, B0000000, B0000000, // {
1, 8, B1111111, B0000000, B0000000, B0000000, B0000000, // |
3, 8, B1000001, B0110110, B0001000, B0000000, B0000000, // }
4, 8, B0001000, B0000100, B0001000, B0000100, B0000000, // ~
};

void setup()
{
    m.init();

    m.setIntensity(3);
}

// Scrolling Text
char string[] = " Hello, World !!";

void loop()
{
    delay(100);

    m.shiftLeft(false, true);
}

```



```

    printStringWithShift(string,100);
}

void printCharWithShift(char c, int shift_speed)
{
    if (c < 32) return;

    c -= 32;

    memcpy_P(Buf7219, CH + 7*c, 7);

    m.writeSprite(maxDisplays*8, 0, Buf7219);

    m.setColumn(maxDisplays*8 + Buf7219[0], 0);

    for (int i=0; i<=Buf7219[0]; i++)
    {
        delay(shift_speed);

        m.shiftLeft(false, false);
    }
}

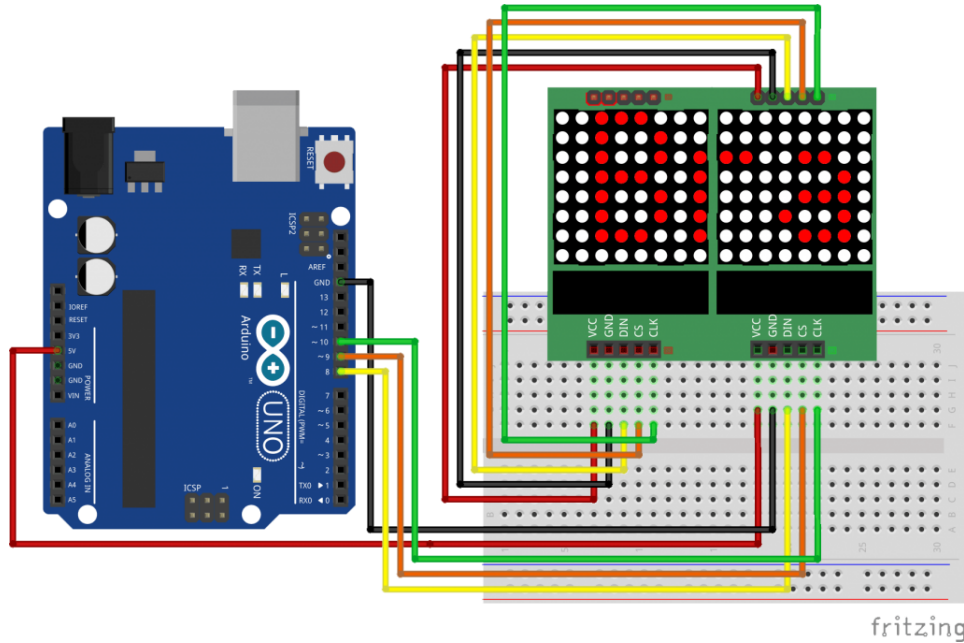
void printStringWithShift(char* s, int shift_speed)
{
    while (*s != 0)
    {
        printCharWithShift(*s, shift_speed);

        s++;
    }
}

```

8x8 LED Dot Matrix MAX7219 Interfacing with Arduino (Using 2 Modules) (Tutorial 2)

ယခု Tutorial တွင် LED Matrix Module နှစ်ခု ကိုချိတ်ဆက်၍ text scrolling ကိုပြုလုပ်ပါမည်။



ပုံပါအတိုင်း pin များကို ချိတ်ဆက်လိုက်ပါ။

Example code မှာအောက်ပါအတိုင်းဖြစ်ပါသည်။ code အား Uno board ထဲသို့ မသွင်းခင် သက်ဆိုင်ရာ library ကို Arduino IDE ထဲသို့ အရင်ဆုံး ထည့်သွင်းပေးရန်လိုအပ်ပါသည်။

MaxMatrix library ကို download ဆွဲရန်: <https://github.com/riyas-org/max7219/tree/master/MaxMatrix>

```
#include "MaxMatrix.h"
#include "avr/pgmspace.h"

PROGMEM const unsigned char CH[] = {
3, 8, B00000000, B00000000, B00000000, B00000000, B00000000,
// space
1, 8, B01011111, B00000000, B00000000, B00000000, B00000000,
// !
3, 8, B00000011, B00000000, B00000011, B00000000, B00000000,
// "
5, 8, B00010100, B00111110, B00010100, B00111110, B00010100,
// #
4, 8, B00100100, B01101010, B00101011, B00010010, B00000000,
// $
```

```
5, 8, B01100011, B00010011, B00001000, B01100100, B01100011,
// %
5, 8, B00110110, B01001001, B01010110, B00100000, B01010000,
// &
1, 8, B00000011, B00000000, B00000000, B00000000, B00000000,
// '
3, 8, B00011100, B00100010, B01000001, B00000000, B00000000,
// (
3, 8, B01000001, B00100010, B00011100, B00000000, B00000000,
// )
5, 8, B00101000, B00011000, B00001110, B00011000, B00101000,
// *
5, 8, B00001000, B00001000, B00111110, B00001000, B00001000,
// +
2, 8, B10110000, B01110000, B00000000, B00000000, B00000000,
// ,
4, 8, B00001000, B00001000, B00001000, B00001000, B00000000,
// -
2, 8, B01100000, B01100000, B00000000, B00000000, B00000000,
// .
4, 8, B01100000, B00011000, B00000110, B00000001, B00000000,
// /
4, 8, B00111110, B01000001, B01000001, B00111110, B00000000,
// 0
3, 8, B01000010, B01111111, B01000000, B00000000, B00000000,
// 1
4, 8, B01100010, B01010001, B01001001, B01000110, B00000000,
// 2
4, 8, B00100010, B01000001, B01001001, B00110110, B00000000,
// 3
4, 8, B00011000, B00010100, B00010010, B01111111, B00000000,
// 4
4, 8, B00100111, B01000101, B01000101, B00111001, B00000000,
// 5
4, 8, B00111110, B01001001, B01001001, B00110000, B00000000,
// 6
4, 8, B01100001, B00010001, B00001001, B00000111, B00000000,
// 7
4, 8, B00110110, B01001001, B01001001, B00110110, B00000000,
// 8
4, 8, B00000110, B01001001, B01001001, B00111110, B00000000,
// 9
2, 8, B01010000, B00000000, B00000000, B00000000, B00000000,
// :
2, 8, B10000000, B01010000, B00000000, B00000000, B00000000,
// ;
```

```
3, 8, B00010000, B00101000, B01000100, B00000000, B00000000,
// <
3, 8, B00010100, B00010100, B00010100, B00000000, B00000000,
// =
3, 8, B01000100, B00101000, B00010000, B00000000, B00000000,
// >
4, 8, B00000010, B01011001, B00001001, B00000110, B00000000,
// ?
5, 8, B00111110, B01001001, B01010101, B01011101, B00001110,
// @
4, 8, B01111110, B00010001, B00010001, B01111110, B00000000,
// A
4, 8, B01111111, B01001001, B01001001, B00110110, B00000000,
// B
4, 8, B00111110, B01000001, B01000001, B00100010, B00000000,
// C
4, 8, B01111111, B01000001, B01000001, B00111110, B00000000,
// D
4, 8, B01111111, B01001001, B01001001, B01000001, B00000000,
// E
4, 8, B01111111, B00001001, B00001001, B00000001, B00000000,
// F
4, 8, B00111110, B01000001, B01001001, B01111010, B00000000,
// G
4, 8, B01111111, B00001000, B00001000, B01111111, B00000000,
// H
3, 8, B01000001, B01111111, B01000001, B00000000, B00000000,
// I
4, 8, B00110000, B01000000, B01000001, B00111111, B00000000,
// J
4, 8, B01111111, B00001000, B00010100, B01100011, B00000000,
// K
4, 8, B01111111, B01000000, B01000000, B01000000, B00000000,
// L
5, 8, B01111111, B00000010, B00001100, B00000010, B01111111,
// M
5, 8, B01111111, B00000100, B00001000, B00010000, B01111111,
// N
4, 8, B00111110, B01000001, B01000001, B00111110, B00000000,
// O
4, 8, B01111111, B00001001, B00001001, B00000110, B00000000,
// P
4, 8, B00111110, B01000001, B01000001, B01011110, B00000000,
// Q
4, 8, B01111111, B00001001, B00001001, B01110110, B00000000,
// R
```

```
4, 8, B01000110, B01001001, B01001001, B00110010, B00000000,  
// S  
5, 8, B00000001, B00000001, B01111111, B00000001, B00000001,  
// T  
4, 8, B00111111, B01000000, B01000000, B00111111, B00000000,  
// U  
5, 8, B00001111, B00110000, B01000000, B00110000, B00001111,  
// V  
5, 8, B00111111, B01000000, B00111000, B01000000, B00111111,  
// W  
5, 8, B01100011, B00010100, B00001000, B00010100, B01100011,  
// X  
5, 8, B00000111, B00001000, B01110000, B00001000, B00000111,  
// Y  
4, 8, B01100001, B01010001, B01001001, B01000111, B00000000,  
// Z  
2, 8, B01111111, B01000001, B00000000, B00000000, B00000000,  
// [  
4, 8, B00000001, B00000110, B00011000, B01100000, B00000000,  
// \ backslash  
2, 8, B01000001, B01111111, B00000000, B00000000, B00000000,  
// ]  
3, 8, B00000010, B00000001, B00000010, B00000000, B00000000,  
// hat  
4, 8, B01000000, B01000000, B01000000, B01000000, B00000000,  
// _  
2, 8, B00000001, B00000010, B00000000, B00000000, B00000000,  
//   
4, 8, B00100000, B01010100, B01010100, B01111000, B00000000,  
// a  
4, 8, B01111111, B01000100, B01000100, B00111000, B00000000,  
// b  
4, 8, B00111000, B01000100, B01000100, B00101000, B00000000,  
// c  
4, 8, B00111000, B01000100, B01000100, B01111111, B00000000,  
// d  
4, 8, B00111000, B01010100, B01010100, B00011000, B00000000,  
// e  
3, 8, B00000100, B01111110, B00000101, B00000000, B00000000,  
// f  
4, 8, B10011000, B10100100, B10100100, B01111000, B00000000,  
// g  
4, 8, B01111111, B00000100, B00000100, B01111000, B00000000,  
// h  
3, 8, B01000100, B01111101, B01000000, B00000000, B00000000,  
// i
```

```

4, 8, B01000000, B10000000, B10000100, B01111101, B00000000,
// j
4, 8, B01111111, B00010000, B00101000, B01000100, B00000000,
// k
3, 8, B01000001, B01111111, B01000000, B00000000, B00000000,
// l
5, 8, B01111100, B00000100, B01111100, B00000100, B01111000,
// m
4, 8, B01111100, B00000100, B00000100, B01111000, B00000000,
// n
4, 8, B00111000, B01000100, B01000100, B00111000, B00000000,
// o
4, 8, B11111100, B00100100, B00100100, B00011000, B00000000,
// p
4, 8, B00011000, B00100100, B00100100, B11111100, B00000000,
// q
4, 8, B01111100, B00001000, B00000100, B00000100, B00000000,
// r
4, 8, B01001000, B01010100, B01010100, B00100100, B00000000,
// s
3, 8, B00000100, B00111111, B01000100, B00000000, B00000000,
// t
4, 8, B00111100, B01000000, B01000000, B01111100, B00000000,
// u
5, 8, B00011100, B00100000, B01000000, B00100000, B00011100,
// v
5, 8, B00111100, B01000000, B00111100, B01000000, B00111100,
// w
5, 8, B01000100, B00101000, B00010000, B00101000, B01000100,
// x
4, 8, B10011100, B10100000, B10100000, B01111100, B00000000,
// y
3, 8, B01100100, B01010100, B01001100, B00000000, B00000000,
// z
3, 8, B00001000, B00110110, B01000001, B00000000, B00000000,
// {
1, 8, B01111111, B00000000, B00000000, B00000000, B00000000,
// |
3, 8, B01000001, B00110110, B00001000, B00000000, B00000000,
// }
4, 8, B00001000, B00000100, B00001000, B00000100, B00000000,
// ~
};

int data = 8; // DIN pin of MAX7219 module
int load = 9; // CS pin of MAX7219 module

```

```

int clock = 10; // CLK pin of MAX7219 module

int maxInUse = 2; //how many MAX7219 are connected

MaxMatrix m(data, load, clock, maxInUse); // define Library

byte buffer[10];

char string1[] = " Brainy-Bits.com "; // Scrolling Text

void setup(){
  m.init(); // module MAX7219
  m.setIntensity(5); // LED Intensity 0-15
}

void loop(){

  byte c;
  delay(100);
  m.shiftLeft(false, true);
  printStringWithShift(string1, 100); // Send scrolling Text
}

// Put extracted character on Display
void printCharWithShift(char c, int shift_speed){
  if (c < 32) return;
  c -= 32;
  memcpy_P(buffer, CH + 7*c, 7);
  m.writeSprite(maxInUse*8, 0, buffer);
  m.setColumn(maxInUse*8 + buffer[0], 0);

  for (int i = 0; i < buffer[0]+1; i++)
  {
    delay(shift_speed);
    m.shiftLeft(false, false);
  }
}

// Extract characters from Scrolling text
void printStringWithShift(char* s, int shift_speed){
  while (*s != 0){
    printCharWithShift(*s, shift_speed);
    s++;
  }
}

```

Ref: Instructables, Circuit Digest, Brainy-Bits

8x8 LED Dot Matrix MAX7219 Interfacing with Arduino ၏ tutorial 1 နှင့် 2 ကို

<https://circuitdigest.com/microcontroller-projects/arduino-8x8-led-matrix> ၊

<https://www.instructables.com/id/Dot-Matrix-With-MAX7219-Interfacing-With> နှင့်

<https://www.brainy-bits.com/scroll-text-using-the-max7219-led-dot-matrix/> တို့မှ မှီငြမ်းပြီး

ဘာသာပြန်ရေးသားထားခြင်းဖြစ်ပါသည်။

ယခု ဘာသာပြန်စာ တွင် အားနည်းမှုများရှိပါက လွတ်လပ်စွာ ဝေဖန်ထောက်ပြနိုင်ပါသည်။

Translated by

Friend Online Store

31.05.2019

Friend Online Store