

## LED matice 32x8 MAX7219

### 1. POPIS

LED matice se skládá ze čtyř segmentů, které dohromady obsahují 256 LED diod. Každý segment tvoří 8 x 8 LED diod. Matici lze využít jako displej, a to zejména pro zobrazení pohyblivého textu nebo časové informace. Avšak využití může být velmi univerzální a na matici lze zobrazovat také jednoduchou grafiku.

Zařízení komunikuje přes rozhraní SPI.

Základní charakteristika:

- celkový počet LED diod 32 x 8
- komunikační rozhraní SPI
- softwarově nastavitelný jas
- napájení 5 V

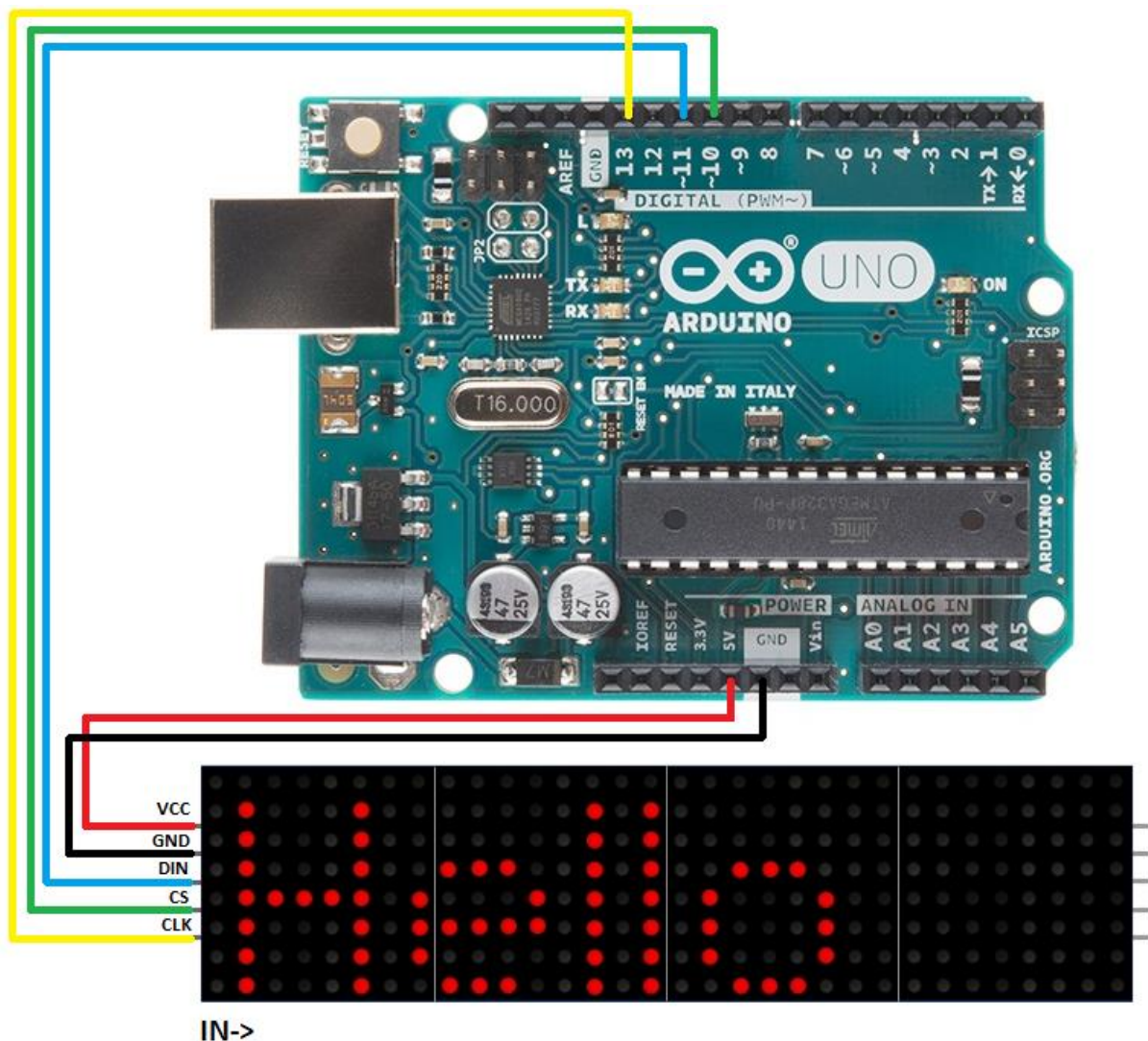


### 2. SPECIFIKACE

<b>Ovladač displeje</b>	MAX7219	<b>Celkový počet LED diod</b>	32 x 8
<b>Napájení</b>	5 V	<b>Barva LED diod</b>	oranžová
<b>Pracovní proud (ukázka programu)</b>	210 mA	<b>Rozhraní</b>	SPI
<b>Max. proud (zapnuté všechny LED)</b>	až 700 mA	<b>Rozměr segmentu (mm)</b>	32 x 32
<b>Počet segmentů</b>	4	<b>Rozměr matice (mm)</b>	129 x 32
<b>Počet LED diod na segment</b>	8 x 8	<b>Hmotnost</b>	46 g

### 3. ZAPOJENÍ

Pokud uživatel v aplikaci využije kompletní rozsvícení maticového displeje při plném jasu, je doporučeno použít samostatné napájení pro LED matici. Proudový odběr se v nárazových případech při plném rozsvícení může vyšplhat až k 700 mA, což by lineární regulátor, v případě externího napájení Arduino, nezvládl (pracuje spolehlivě do 500 mA). Při napájení Arduino přes USB konektor lze matici napájet přímo z vývojové desky, ale velmi závisí na zdroji (PC nebo USB huby by mohly mít s proudem 700 mA problém). Samostatné napájení LED matice lze realizovat např. pomocí 1476823567.





## 4. UKÁZKA PROGRAMU

K úspěšné kompilaci a nahrání programu je nutné stáhnout knihovny [Adafruit\\_GFX.h](#) a [Max72xxPanel.h](#).

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>

int pinCS = 10;
int numberOfHorizontalDisplays = 1;
int numberOfVerticalDisplays = 4;

Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays, numberOfVerticalDisplays);

String tape = "Arduino-shop.cz";
int wait = 150;

int spacer = 1;
int width = 5 + spacer;

void setup() {
  matrix.setIntensity(15);
  matrix.setRotation(3);
}

void loop() {

  for ( int i = 0 ; i < width * tape.length() + matrix.width() - 1 - spacer; i++ ) {

    matrix.fillScreen(LOW);

    int letter = i / width;
    int x = (matrix.width() - 1) - i % width;
    int y = (matrix.height() - 8) / 2; // center the text vertically

    while ( x + width - spacer >= 0 && letter >= 0 ) {
      if ( letter < tape.length() ) {
        matrix.drawChar(x, y, tape[letter], HIGH, LOW, 1);
      }
      letter--;
      x -= width;
    }

    matrix.write();

    delay(wait);
  }
}
```