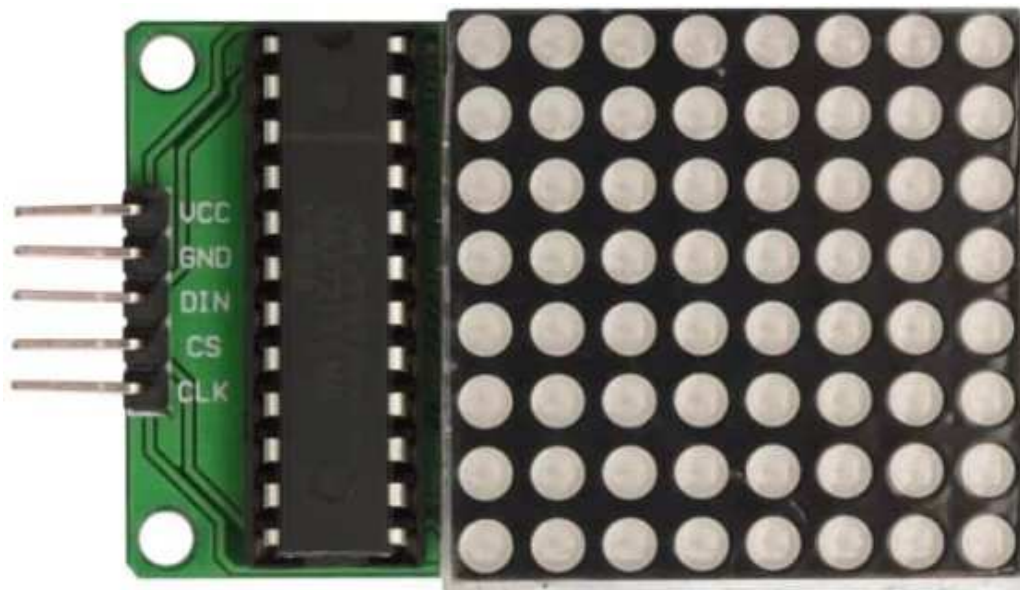
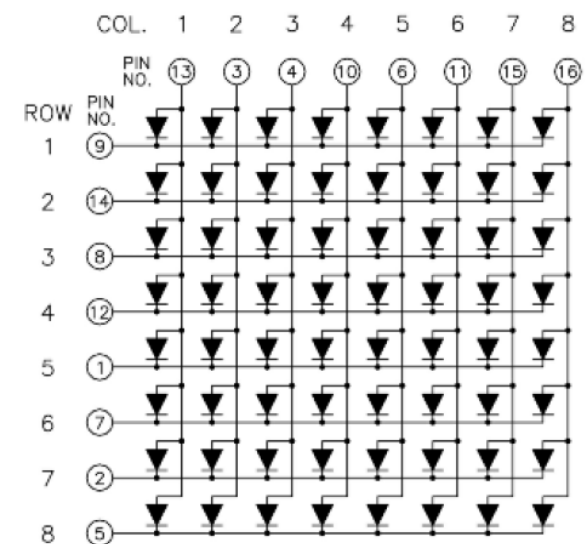


Modul LED matice s řadičem MAX7219.

MAX7219 je sériový řadič displejů, který umí obsluhovat 8 řádků (displejů) a 8 sloupců (segmentů). Aby bylo možné využít kaskádního řazení obvodů MAX7219 a modulů odpovídá rozsvícení displeje **řádku matice** a nikoliv sloupci matice. Tím je usnadněn posun nápisů po kaskádně zapojených maticích.



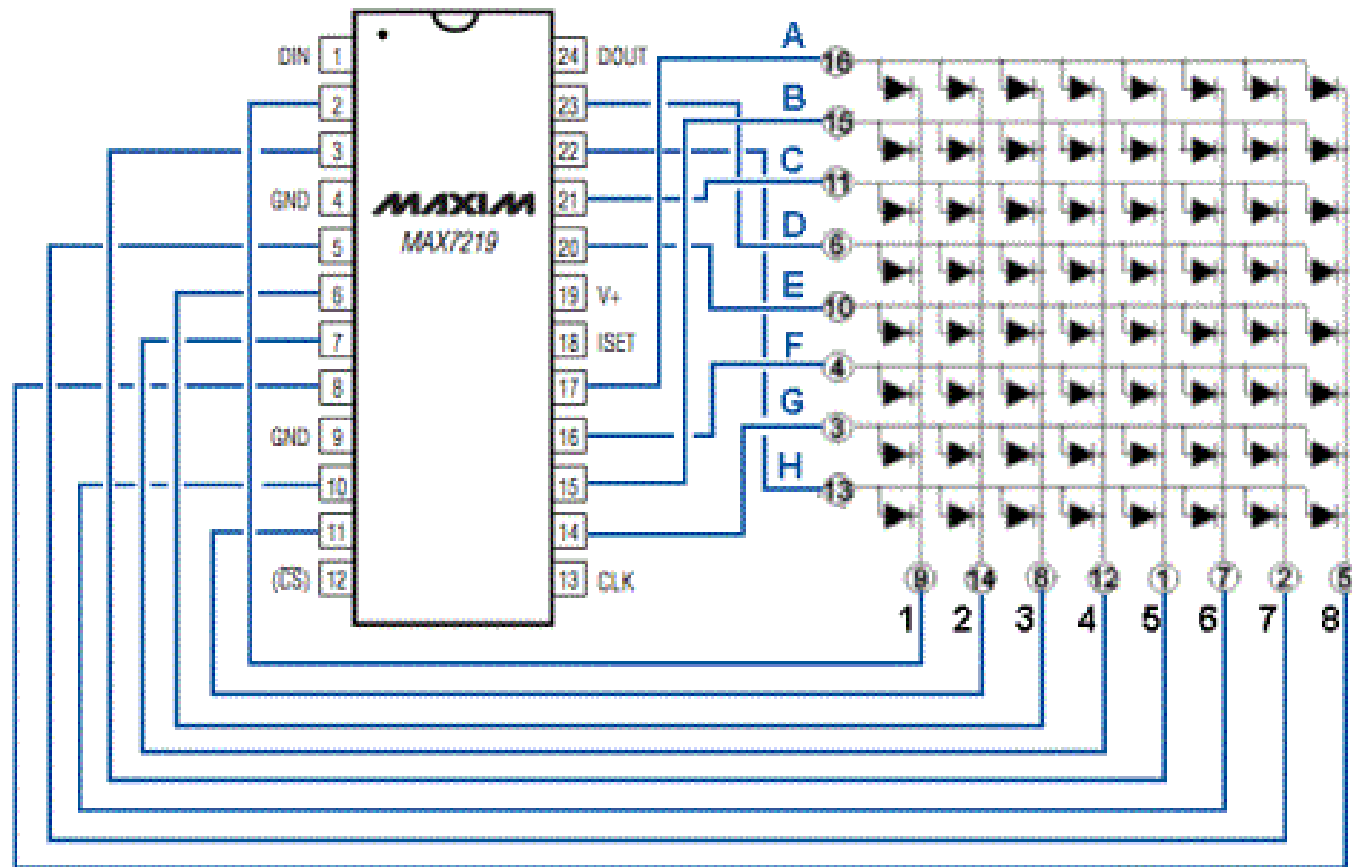
Řádek 0
Řádek 1
Řádek 2
Řádek 3
Řádek 4
Řádek 5
Řádek 6
Řádek 7



Radič obsahuje 64-bitovou paměť pro uložení nastavení rozsvícených LED a sám zajišťuje jejich dynamické střídání. Programový cyklus musí realizovat procesor pouze při změně informace. Spotřeba se odvíjí od počtu rozsvícených LED a nastaveném jasu.

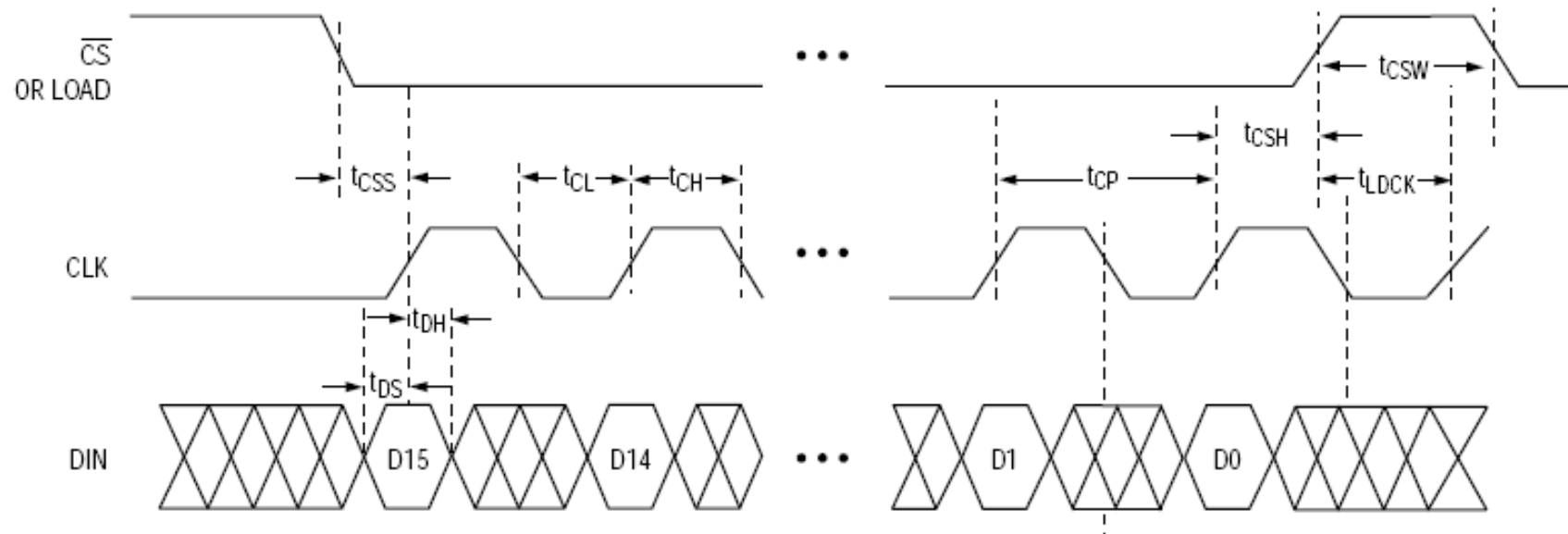
ZAPOJENÍ BODOVÉHO DISPLEJE

Mikrokontrolér komunikuje s řadičem po sběrnici SPI. Z řadiče vede 16 vodičů do LED matice - jeden na každý sloupec i řádek. Schéma připojení matice k řadiči je na obrázku. Matice diod je oproti obrázku na str.1 pootočená o 90° doleva.



OVLÁDÁNÍ BODOVÉHO DISPLEJE

Komunikace s řadičem probíhá po rozhraní SPI, kdy se přenáší 16 bitů potvrzovaných náběžnou hranou CLK při signálu CS=0.



Přenášená data mají následující formát

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

Adresy X1 až X8 slouží k zápisu stavu LED do jednotlivých řádků matice. Adresy X9, XA, AB, XC a XF ovládají chování řadiče.

OVLÁDÁNÍ BODOVÉHO DISPLEJE

- K ověření displeje slouží testovací mód odesláním hodnoty:
 - xxxx 1111 xxxx xxx1, při kterém se rozsvítí všechny diody.
 - Normální činnost zajistíme příkazem xxxx 1111 xxxx xxx0.
- Následně nastavíme registr shutdown příkazem:
 - xxxx 1100 xxxx xxx0 = Shutdown
 - xxxx 1100 xxxx xxx1 = Normální činnost
- Posledním důležitým příkazem je registr Scan-Limit, který určuje počet ovládaných displejů (v našem případě řádků). Pro všechny řádky pošleme příkaz

xxxx 1011 xxxx x111

- Intenzitu jasu ovládáme příkazem xxxx 1010 xxxx bbbb, kde bbbb je binární hodnota od 0 do 15. 0 nejmenší jas, 15 největší.
- Po inicializaci V/V vývodů a nastavení do potřebného stavu vyšleme sekvenci posli(0x0F00); posli(0x0B07); posli(0x0A02); posli(0x0C01); a for (j=1; j<9; j++) { posl=(j)<<8; posli(posl);}

- Ovládání tlačítka pro změnu obrázku nebo stav sekvenčního posunu je vhodné realizovat přes přerušení, v kterém je spuštěn časovač. Při uvolnění tlačítka bude čítač zastaven a provedeno vyhodnocení.
- Alternativně může být využito záchytného systému po spojení PB13 s jedním vývodem záchytného systému.
- Optimální bude jeden řádek generovaných obrázků realizovat 32 bitovou proměnnou.
- Sekvenční posun pak bude odpovídat realizaci HADA, tvořenou potřebnou sekvencí v jazyce C, vloženým assemblerem nebo volanou assemblerovskou funkcí.