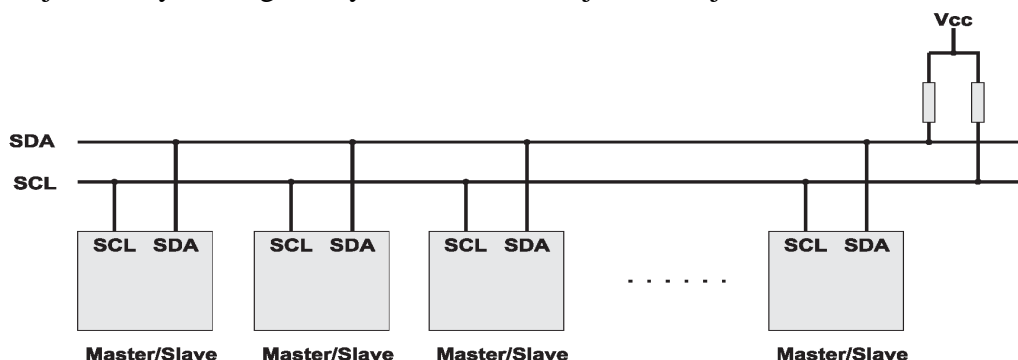


# PERIFERNÍ OBVODY A ROZHRANÍ V MIKROPROCESOROVÝCH SYSTÉMECH

## 1.1 Sběrnice I<sup>2</sup>C

I<sup>2</sup>C bus je zkratka, která vznikla z IIC bus, tedy Internal-Integrated-Circuit Bus. Jak již název napovídá, jedná se o interní datovou sběrnici sloužící pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody většinou v rámci jednoho zařízení. Vyvinula ji firma Philips přibližně před 20-ti lety. V dnešní době tento způsob komunikace podporuje řada integrovaných obvodů nejen firmy Philips. Jedná se především o mikrokontroléry, sériové paměti, inteligentní LCD, audio a video obvody, A/D a D/A převodníky a některé další digitálně řízené obvody. Hlavní výhodou je, že obousměrný přenos probíhá pouze po dvou vodičích - "data SDA (serial data)" a "hodiny SCL (serial clock)". To především u mikrokontrolérů výrazně optimalizuje nároky na počet vstupně-výstupních pinů a celkově zjednodušuje výsledné zapojení. Na jednu sběrnici může být připojeno více integrovaných obvodů. V základní verzi jsou obvody adresovány 7-mi bitově a v rozšířené verzi 10-ti bitově. To umožňuje připojení 128, respektive 1024 čipů s různou adresou na jednu společnou sběrnici. V praxi jsou tato čísla však podstatně nižší, protože adresu čipu většinou nelze nastavit libovolně, ale je z části pevně daná výrobcem a ovlivnit je možné např. jen tři bity. Někdy ji nelze určit vůbec a je určena napevno pro daný typ čipu - takových čipů tedy na jedné sběrnici nemůže být více. Přenosová rychlost sběrnice je pro většinu aplikací dostatečná i v základní verzi, kde je frekvence hodin 100kHz. Ve vylepšených verzích to může být i 400kHz nebo 1MHz. Rychlost přenosu musí být pochopitelně přizpůsobena nejpomalejšímu čipu na sběrnici. Oba vodiče musí být implicitně v logické jedničce a to je zajištěno pull-up rezistory. Jejich odpory mají hodnotu v řádech jednotek kΩ. Čím je vyšší komunikační frekvence, tím musí být nižší hodnoty těchto odporů. Pro 100kHz postačuje 4k7. Způsob propojení jednotlivých integrovaných obvodů ukazuje následující obrázek.



Obr.: 1.1 Propojení obvodů na sběrnici I<sup>2</sup>C

### 1.1.1 Princip přenosu

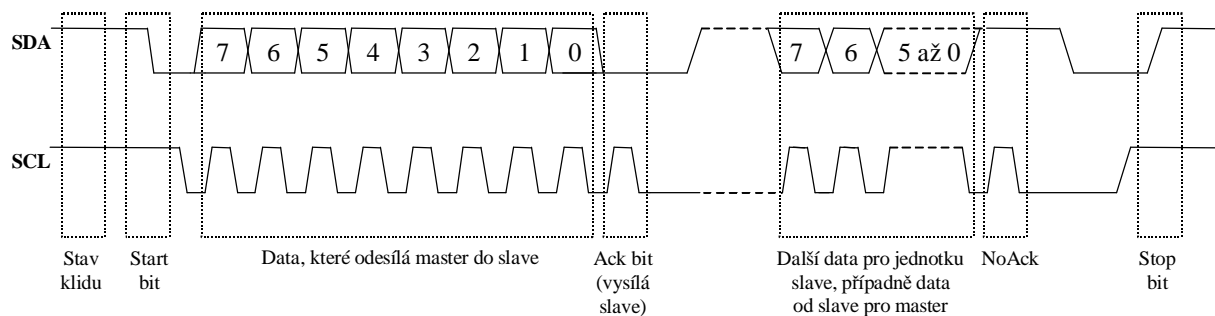
Jeden z integrovaných obvodů (většinou mikrokontrolér) je nastaven jako master a všechny ostatní obvody jsou slave. Obvody se dají zapojit i jako tzv. multi-master, kdy je čipů master několik. Pro účely úlohy bude však popis omezen na zapojení s jedním master čipem a 7-mi bitovou adresací.

Master při jakémkoli přenosu generuje hodinový signál na vodiči SCL. Když jeden čip vysílá, všechny ostatní přijímají a podle adresy určují, zda jsou data určena jim. Čip, který chce vyslat/přijmout data, musí tedy nejprve definovat adresu čipu s kterým chce

komunikovat a zda půjde o příjem nebo vysílání - tedy o čtení nebo zápis. To určuje R/W (read/write) bit, který je součástí adresy. Přenos probíhá kombinováním následujících kroků:

- **stav klidu** - Je zajištěn logickými jedničkami na obou vodičích, master tedy negeneruje hodinový signál a neprobíhá žádný přenos. Logické jedničky jsou na obou vodičích zajištěny pull-up rezistory (rezistory mezi vodičem a napájecím napětím), takže klidový stav nastane i tehdy, když jsou výstupy obvodu master ve stavu vysoké impedance (tedy odpojeny).
- **START bit** - Zahajuje přenos nebo jeho další část. Je vygenerován tak, že se změní úroveň *SDA* z 1 na 0 zatímco je *SCL* v logické 1.
- **STOP bit** - Ukončuje přenos. Je vygenerován podobně jako start bit. Logická úroveň *SDA* se změní z 0 na 1 zatímco je *SCL* v logické 1. Stop bit může být generován pouze po nepotvrzení přenosu, tedy pouze po přijmutí ACK v logické 1. (viz níže)
- **přenos dat** - Data jsou přenášena po 1Byte, tedy 8 po sobě jdoucích bitů od nejvyššího po nejnižší. Při přenosu dat se může logická úroveň na *SDA* měnit pouze pokud je *SCL* v logické 0. Při každém pulzu na *SCL* je přenesen jeden bit.
- **potvrzující bit ACK** (Acknowledge) - Tento bit slouží k potvrzení správného přijmutí dat. ACK bit se odesílá stejným způsobem, jako by se odesílal devátý bit dat s tím rozdílem, že ho generuje čip, který přijímal (slave) a nikoliv ten který data odesílal. Pokud přenos proběhl v pořádku tak odešle logickou 0. Logická 0 potvrzujícího bitu rovněž znamená, že je přijímač připraven na příjem dalšího byte, který následuje okamžitě po něm při dalším pulzu na *SCL*. Pokud přenos selhal, odešle logickou 1. Pokud má dojít k ukončení přenosu, tak neodešle nic. Pull-up rezistor pak zajistí, že bude na *SDA* logická 1 a ACK bit (v logické 0) odešle vysílač.

Časový průběh logických úrovní na vodičích *SDA* a *SCL* je na následujícím obrázku.

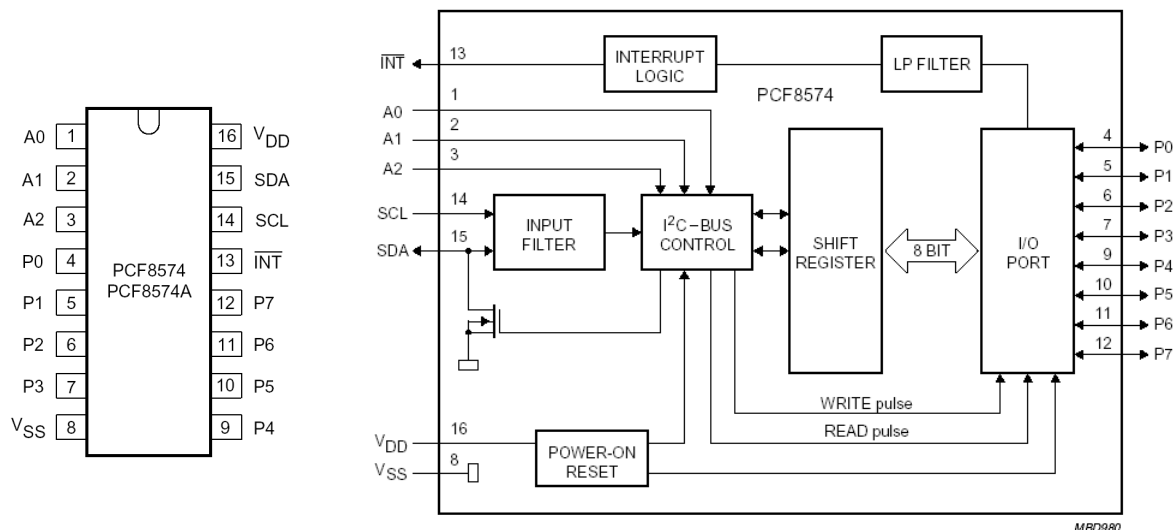


Obr.: 1.2 Časový průběh komunikace po sběrnici I<sup>2</sup>C

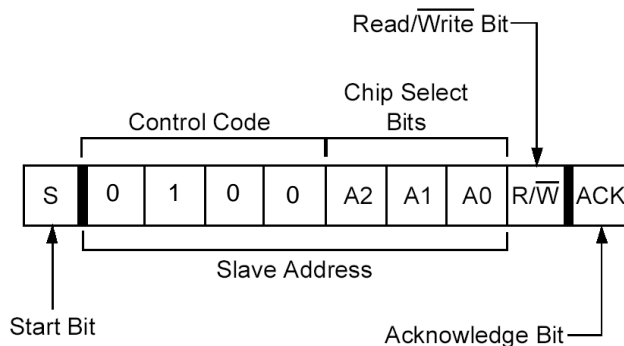
### 1.1.2 8 bitový I/O expandér s rozhraním I<sup>2</sup>C PCF8574

Obvod PCF8574 [1.3] slouží pro rozšíření počtu vstupů/výstupů mikrokontroléru. Blokové schéma obvodu je na obrázku 1.3. I/O brána P0÷P7 je řešena tak, že každý bit může být použit jako vstup i výstup. Pokud je na příslušný pin zapsána hodnota log 0, je sepnut příslušný tranzistor a výstup může dodávat proud až 25 mA do zátěže zapojené proti napájecímu napětí. Při zápisu log 1 se k příslušnému pinu připojí přes interní rezistor napájecí napětí V<sub>dd</sub>. Zatížitelnost výstupu je v tomto případě pouze cca 100µA, což je dostatečné pro buzení vstupů případných dalších logických obvodů, zároveň to ale umožňuje „vnutit“ vnějším signálem hodnotu log 0. Pokud je tedy na příslušném pinu nastavena hodnota log 1, je možné používat jej i jako vstup (například kontakt připojený proti GND),

jehož hodnota je dostupná přečtením hodnoty vstupního registru přes rozhraní I<sup>2</sup>C. Na výstupu  $\overline{INT}$  se objeví puls v případě změny libovolného vstupního pinu. To umožňuje nadřazenému obvodu, aby vyčkával na případnou změnu a teprve poté přečetl novou hodnotu (není nutné cyklické čtení).



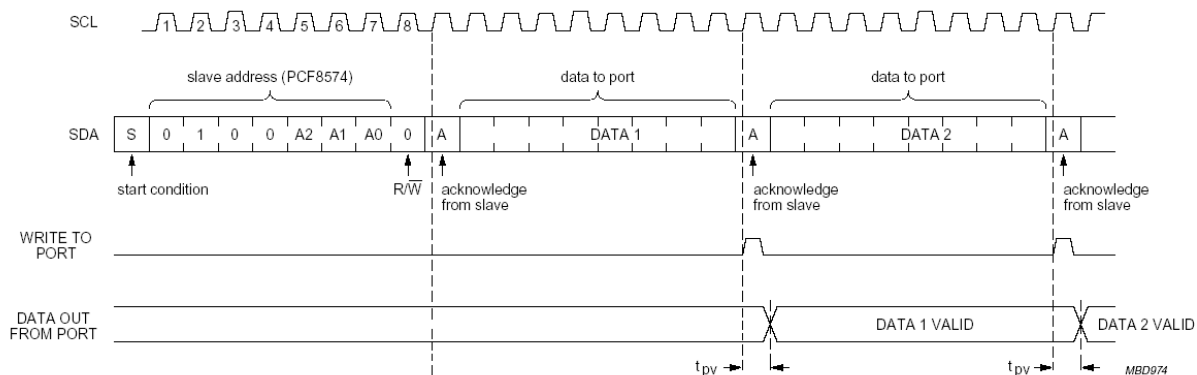
**Obr.: 1.3** Uspořádání vývodů a blokové schéma obvodu PCF8574



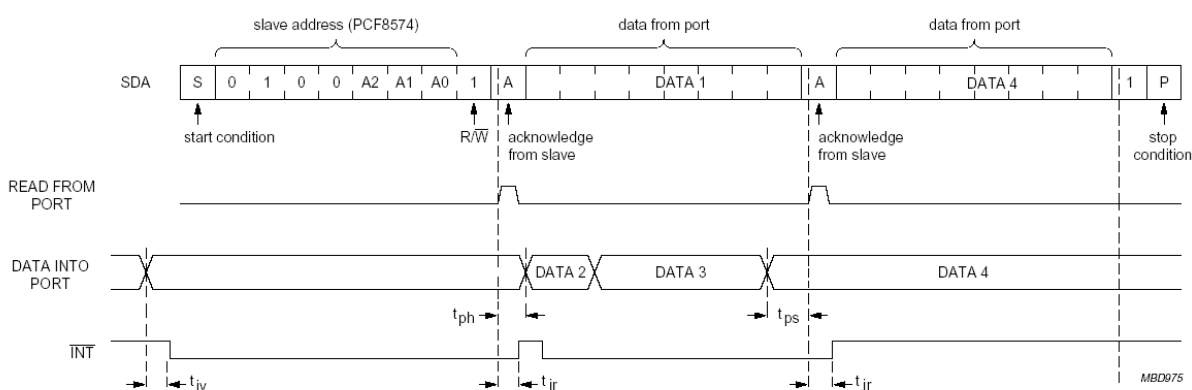
**Obr.: 1.4** Způsob adresování obvodu PCF8574

7 bitová adresa v rámci I<sup>2</sup>C je určena podle obrázku 1.4. Čtyři vyšší bity jsou tedy pevně dány výrobcem obvodu, spodní tři je pak možné nastavit pomocí vstupů A0÷A2. Poslední bit  $R/\overline{W}$  pak určuje, zda jsou data do obvodu zapisována nebo čtena.

Následující obrázky pak ukazují kompletní cyklus zápisu resp. čtení. Je zde také vidět, že dokud není vyslán STOP bit, je možné data opakovaně zapisovat resp. číst bez nutnosti znovu odesílat adresu zařízení. Jakmile je však přenos ukončen STOP bitem, je nutné opakovat celou sekvenci včetně adresy zařízení.



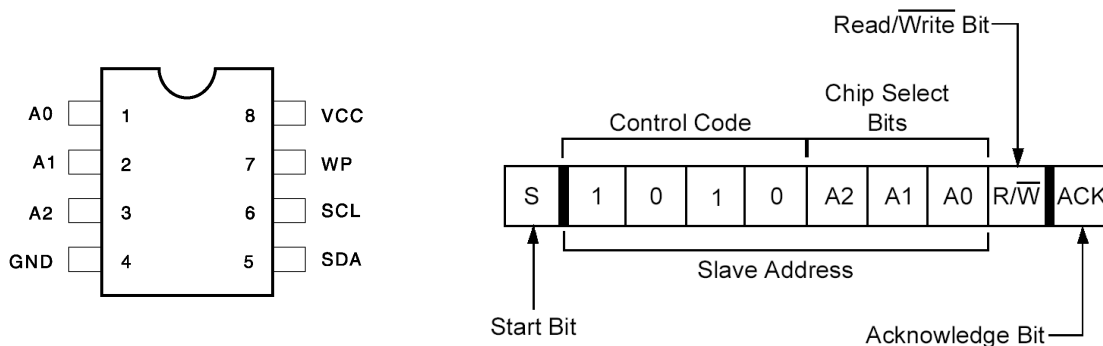
Obr.: 1.5 Časový diagram zápisu



Obr.: 1.6 Časový diagram čtení

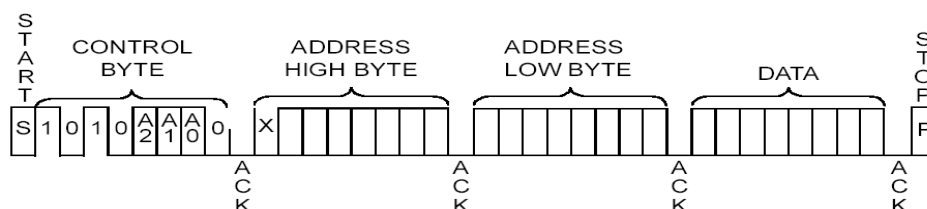
### 1.1.3 Sériová paměť EEPROM

Tato paměť je dalším typickým příkladem obvodu s rozhraním I<sup>2</sup>C. Existuje několik verzí lišících se především kapacitou, v následujícím popisu se ale zaměříme pouze na konkrétní typ 24LC256 s kapacitou 256 Kbit (32K x 8). Stejně jako u obvodu PCF8574 je část adresy pevná a pouze 3 bity lze nastavit pomocí vstupů A0÷A2. Obvod má navíc vstup WP (Write Protect), který umožňuje při hodnotě log 1. blokovat možnost zápisu.



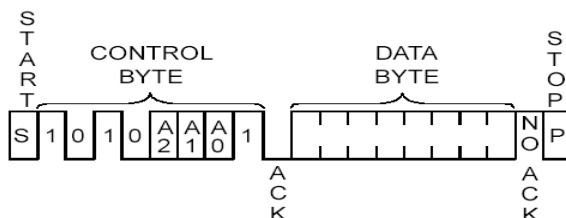
Obr.: 1.7 Zapojení vývodů a způsob adresace paměti 24LC256

Způsob zápisu a čtení je opět podobný jako u obvodu PCF8574, proto již bude uvedena pouze sekvence na datovém signálu *SDA*. Následující obrázek ukazuje způsob zápisu jednoho bytu na konkrétní adresu. Pokud přenos není ukončen okamžitě *STOP* bitem, je možné zapisovat další byty s tím, že adresa je automaticky inkrementována. Maximálně je možné takto zapsat 64 po sobě následujících bytů.



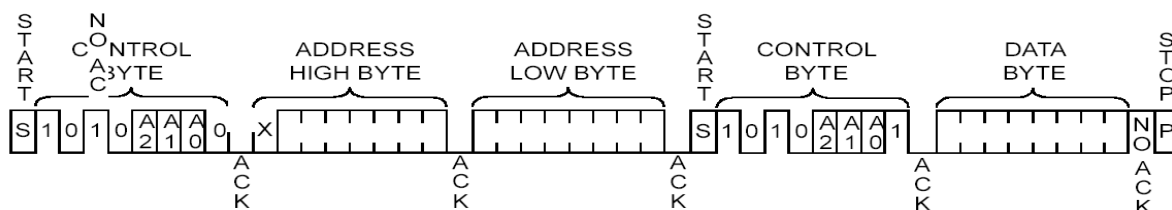
**Obr.: 1.8 Časový diagram zápisu**

Obvod v sobě obsahuje čítač adres, který vždy ukazuje na naposledy čtený nebo zapisovaný byte inkrementovaný o 1. Čtení bytu určeného tímto interním ukazatelem pak vypadá následovně:



**Obr.: 1.9 Časový diagram čtení bytu určeného interním ukazatelem**

Pokud se přenos ukončí *ACK* (nikoli *NO ACK* jak je zobrazeno na obrázku), může následovat čtení následujících bytů. Přenos se ukončí až signálem *NO ACK*. Pokud je třeba změnit ukazatel na konkrétní pozici je třeba použít následující postup:

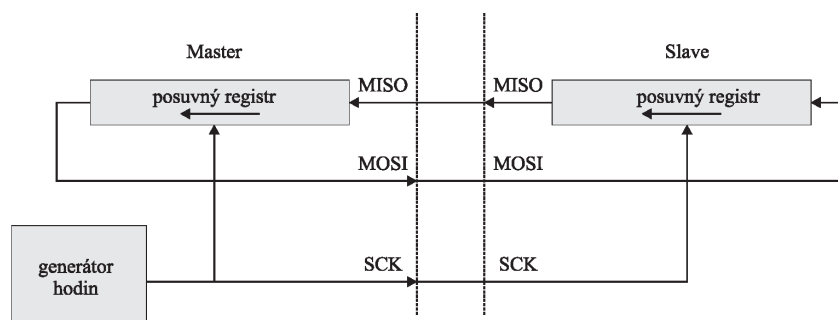


**Obr.: 1.10 Časový diagram čtení bytu na konkrétní pozici**

Začátek je tedy shodný jako při zápisu (tím je dosaženo nastavení interního ukazatele na požadovanou adresu), ale před zapsáním bytu je vyslán opět *START* bit. Poté již následuje čtení zapsané výše.

## 1.2 Rozhraní SPI

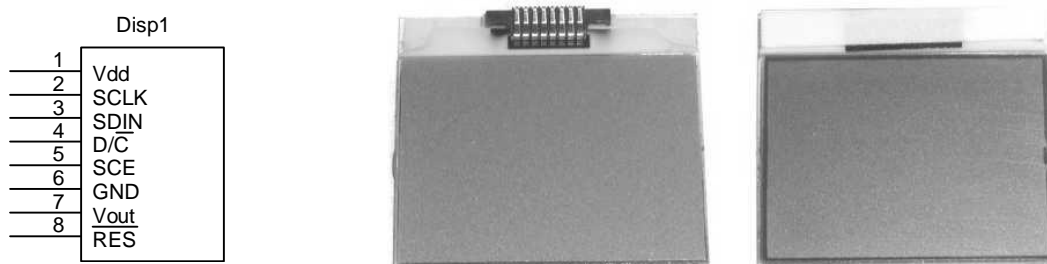
Sběrnice SPI (Serial Peripheral Interface) slouží podobně jako I<sup>2</sup>C k propojení procesoru s podřízenými obvody. Přenos dat je obousměrný a sběrnici pak tvoří tři vodiče pro vlastní přenos dat a volitelně jeden vodič pro výběr podřízeného obvodu. Propojení procesoru s jedním podřízeným obvodem je na obr. 1.11. K přenosu slouží signály *MISO* (Master In/Slave Out – tj. přenos z periferie do procesoru) a *MOSI* (Master Out/Slave In). Synchronizaci zajišťuje hodinový signál *SCK* (Seriól Clock). Pokud je v zapojení více obvodů slave, je k jejich výběru třeba použít ještě signál  $\overline{SS}$  (Slave Select). Způsob přenosu je tedy zřejmý – master vygeneruje 8 hodinových pulsů a tím přeneše 8 datových bitů do obvodu slave a zároveň sám přijme nová data.



Obr.: 1.11 Princip komunikace pomocí SPI

### 1.2.1 Grafický displej z NOKIA 3310 s rozhraním SPI

Tento displej je použit jako jednoduchá ukázka periferie s SPI rozhraním. Displej pracuje s napájecím napětím 3,3V. Zapojení vývodů a náhled displeje je na obrázku 1.12.

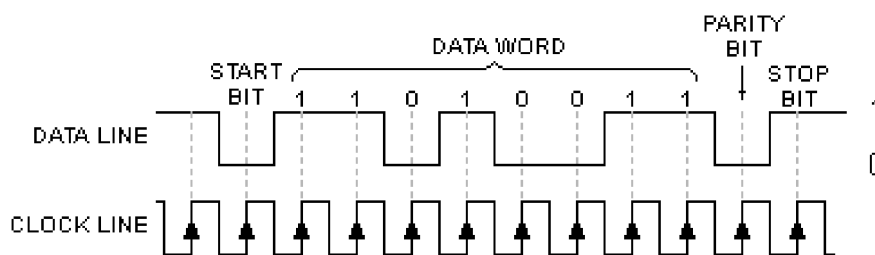


Obr.: 1.12 Zapojení vývodů a pohled na displej

Na vývody Vdd a GND se připojuje napájení. Vout je výstup z interního měniče napětí, který zajišťuje napájení samotného LCD. Pro správnou funkci je na tento vývod třeba zapojit kapacitor alespoň 1 $\mu$ . *SCLK* a *SDIN* odpovídají signálům *SCK* a *MOSI*, které byly vysvětleny výše (jedná se pouze o jednosměrnou komunikaci a signál *MISO* proto chybí).  $\overline{RES}$  slouží k nulování řadiče displeje po zapnutí,  $D/\overline{C}$  určuje, zda se zapisují data nebo řídicí příkazy a *SCE* povoluje komunikaci s displejem (Chip Enable). Konkrétní popis komunikačního protokolu přesahuje rámec této úlohy a proto nebude uváděn.

### 1.3 UART a sériové rozhraní RS232

Většina mikrokontrolérů v sobě obsahuje tzv. UART (Universal Asynchronous serial Receiver and Transmitter). Někdy se používá ještě samostatný signál pro přenos hodin a v tom případě mluvíme o synchronním přenosu. Dále se však budeme zabývat pouze používanějším asynchronním přenosem. Rozhraní používá dva základní signály *RXD* (Receive Data) a *TXD* (Transmit Data). Rámec se vždy skládá ze START bitu, datových bitů, paritního bitu (volitelný) a STOP bitu. Časový průběh přenosu typického rámce je obrázku 1.13.



**Obr.: 1.13 Časový průběh při sériovém přenosu**

Hodinový signál si v případě asynchronního přenosu vytváří vysílací i přijímací strana nezávisle. Je proto nutné, aby přenosová rychlost byla předem známa a nastavena shodně na obou stranách. Běžně používané hodnoty jsou 1200, 2400, 4800, 9600, 1440, 19200 Baudů (bitů za sekundu), většina obvodů však dovoluje nastavit i jiné hodnoty. K synchronizaci hodinového signálu vysílače a přijímače slouží START bit, kterým je zahájen přenos každého rámce (v klidovém stavu je datový signál trvale ve stavu log 1 a sestupná hrana START bitu zajistí synchronizaci). Poté následuje dohodnutý počet datových bitů (např. ATmega 128 umožňuje 5, 6, 7, 8 a 9 datových bitů). Pro jednoduchou kontrolu neporušenosti dat slouží paritní bit. Ten doplní počet jedniček v datech tak, aby byla splněna předem určená podmínka. Nejpoužívanější je lichá (celkový počet jedniček včetně paritního bitu je lichý – případ na obr. 1.13 nebo sudá parita. Je také možno paritní bit úplně vynechat. Je zřejmé, že pomocí paritního bitu lze spolehlivě detekovat pouze chybu v jednom bitu, při vícenásobné chybě se může rámec vzhledem k paritě jevit jako správný. Každý přenos končí STOP bitem (volitelně může být i dvojitý). Všechny parametry přenosu musí být samozřejmě předem správně nastaveny na vysílací a i přijímací straně, jinak nelze přenos uskutečnit. Protože jsou použity samostatné signály pro každý směr přenosu, lze u některých obvodů dokonce vysílat i přijímat data současně.

Norma RS232 pak pouze definuje konkrétní fyzickou vrstvu, tedy napěťové úrovně (a další elektrické parametry) zmíněných signálů *RXD*, *TXD* (a případných dalších řídicích signálů), protože přímé použití TTL úrovní by bylo málo odolné proti rušení. Konkrétní napěťové úrovně shrnuje tab. 1.1.

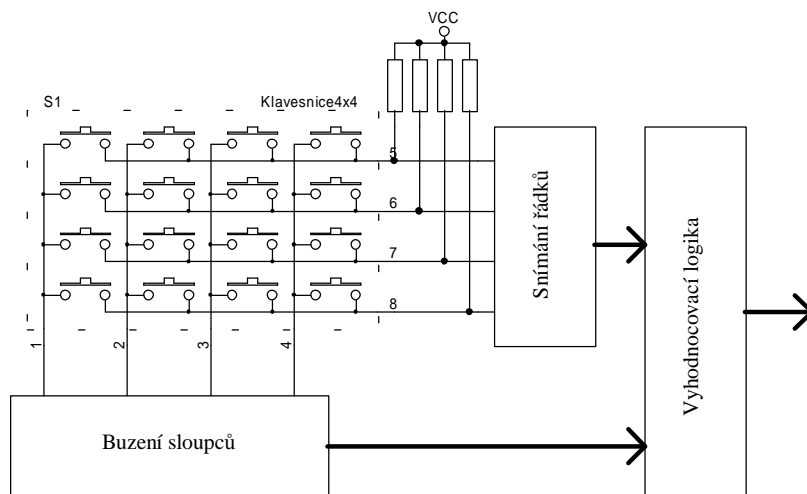
Logická úroveň	Napěťové úrovně vysílače	Napěťové úrovně přijímače
Log. 0	+5 V až +15 V	+3 V až +25 V
Log. 1	-5 V až -15 V	-3 V až -25 V
Nedefinovaný stav	-3 V až +3 V	

**Tab.: 1.1 Napět'ové úrovně standardu RS232**

Někdy se při přenosu používají ještě další řídicí a pomocné signály, jako je např. RTS (Request To Send), CTS (Clear To Send) a několik dalších. Pro účely této úlohy však jejich význam není důležitý.

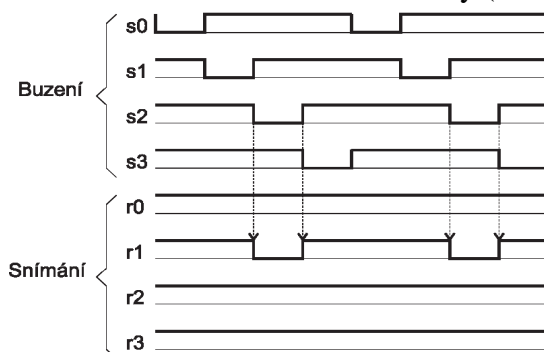
### 1.4 Snímání maticové klávesnice

Pro snímání stavu většího počtu spínačů (například klávesnice) je obvyklé maticové zapojení. Některé prodávané klávesnice jsou svým uspořádáním pro tento druh snímání přímo určeny. Schéma zapojení vyhodnocovacího obvodu pro 16 kláves v matici 4x4 je na obr. 1.14.



**Obr.: 1.14 Zapojení pro snímání maticové klávesnice**

Všechny řádkové vodiče jsou v klidu udržovány na úrovni H pomocí rezistorů R. Budicí obvod pak postupně připojuje na jednotlivé sloupcové vodiče úroveň L. Je-li stisknuta některá z kláves připojených na právě buzený vodič (tj. na úrovni L), je tato úroveň klávesnicovým spínačem připojena i na příslušný řádkový vodič. Vyhodnocovací elektronika může z takto zjištěných souřadnic snadno odvodit kód klávesy (viz obr. 1.15).



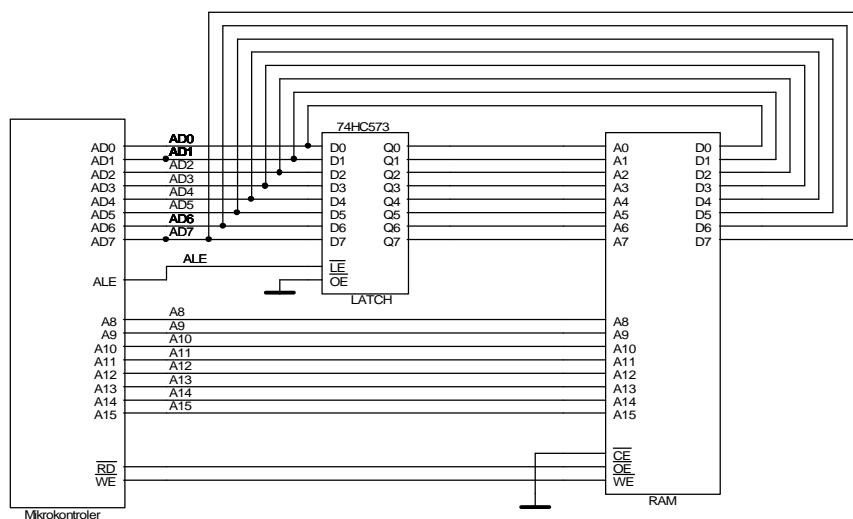
**Obr.: 1.15 Časový diagram snímání maticové klávesnice**



Uvedený způsob snímání klávesnice je snadno realizovatelný pomocí mikrokontroléru, který svými porty může přímo budít sloupce a snímat řádky klávesnice. Je také zřejmé, že ekvivalentní je budít řádky a snímat sloupce.

## 1.5 Připojování externí paměti k mikrokontroléru

U většiny mikrokontrolérů se z důvodů úspory potřebných vývodů sdružuje nižší část adresní sběrnice s datovou sběrnicí. Pro připojení paměti (nebo jiného zařízení v adresním prostoru) je pak nutné použít záchytný registr (Latch), který „podrží“ spodní část adresy a piny procesoru pak mohou být použity pro zápis nebo čtení dat. Připojení je patrné z obrázku 1.16.

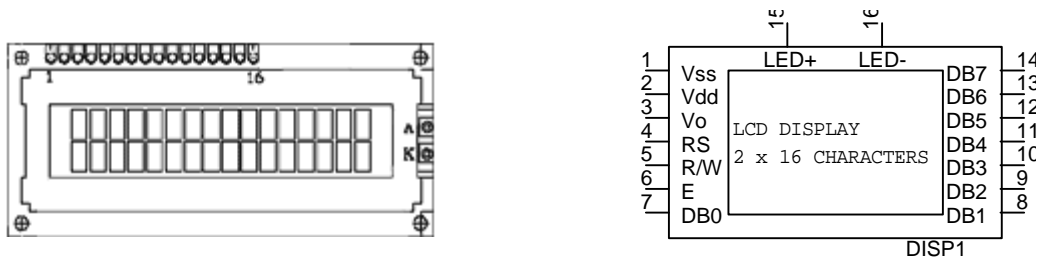


Obr.: 1.16 Princip připojení externí paměti k mikrokontroléru

Mikrokontrolér nejprve nastaví adresu a poté aktivuje signál ALE (Address Latch Enable), čímž je spodní část adresy uložena do záchytného registru. Poté již může být proveden zápis nebo čtení obvyklým způsobem (viz úloha 10 - Polovodičové paměti). Vstup paměti  $\overline{CE}$  může být využit v případě, že je v jednom zapojení více pamětí a je třeba určit, se kterou se právě pracuje.

## 1.6 Připojování inteligentního znakového LCD displeje

V dnešních konstrukcích se velmi často používají znakové LCD s integrovaným řadičem, které značně zjednodušují jeho spolupráci s mikrokontrolérem. Existuje několik typů řadičů, ale pro účely této úlohy se omezíme pouze na popis nejpoužívanějších typů kompatibilních s HD44780. Displeje jsou vyráběny v celé řadě velikostí (2x8 znaků, 2x16 znaků, 4x 16 znaků atd.), v této úloze je konkrétně použit typ EL1602A s velikostí 2x16 znaků.



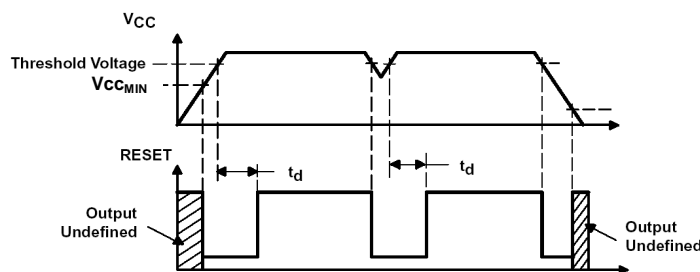
**Obr.: 1.17** Náhled displeje a popis vývodů

Vývody  $V_{ss}$  a  $V_{dd}$  slouží pro napájení displeje (v našem případě 5V), napětím na vstupu  $V_o$  se nastavuje požadovaný kontrast. Vstupy  $LED+$  a  $LED-$  slouží pro napájení podsvětlovacích LED. Samotné komunikační rozhraní se pak skládá ze tří řídicích a osmi datových signálů  $DB0$ ,  $DB7$ . Signál  $RS$  (Register Select) určuje, zda je do displeje posílána řídicí instrukce nebo se jedná o data. Pinem  $R/\overline{W}$  (Read/Write) se přepíná mezi režimem zápis nebo čtení a vstupem  $E$  (Enable) se provede samotný zápis dat nastavených na pinech  $DB0$ ,  $DB7$  do paměti displeje (konkrétně jeho sestupnou hranou). Displej může pracovat také v tzv. 4 bitovém režimu, ve kterém jsou využity pouze piny  $DB4$ ,  $DB7$  a nižší polovina datové sběrnice je nezapojena. V tom případě je potřebných 8 bitů přeneseno nadvakrát, nejprve horní polovina a poté dolní. Jsou proto nutné dvě sestupné hrany signálu  $E$  pro úspěšné zapsání celých 8-mi bitů. Hlavní výhodou tohoto režimu je úspora potřebných vývodů procesoru.

Samotný řadič v sobě obsahuje znakovou sadu, umožňuje definici uživatelských znaků, umí zobrazovat blikající kurzor, posouvat text a další funkce. Vše se ovládá pomocí sady příkazů, které se displeji předávají výše popsaným postupem. Přesný popis příkazů řadiče přesahuje rámec této úlohy, lze je případně nalézt na stránkách výrobce.

## 1.7 Hlídací obvod napájecího napětí

Aby se předešlo nedefinovanému stavu procesoru po náběhu napájecího napětí, případně při jeho neočekávaném poklesu, připojuje se na vstup  $RESET$  speciální obvod (Supply-Voltage Supervisor). V popisovaném přípravku je použit obvod MCP120 [1.4], jehož funkce je dobře patrná z obr. 1.18. Po náběhu napětí na pevně danou mez (Threshold Voltage) je vygenerován puls konstantní délky, který zaručeně zajistí správný reset mikrkontroléru. Podobně se obvod chová i při náhlém poklesu a opětovném zvýšení napájecího napětí.



**Obr.: 1.18** Princip funkce obvodu hlídání napájecího napětí