



Řízení sériové komunikace

Mikroprocesorová technika a embedded systémy

Přednáška 7

doc. Ing. Tomáš Frýza, Ph.D.

říjen 2012

Obsah přednášky

SPI komunikace (Serial Peripheral Interface)

Synchronní a asynchronní komunikace

Sériové rozhraní RS-232

IrDA přenos

I2C sběrnice

Ukázka programu v jazyce C pro ATmega16

Komunikace prostřednictvím TWI (I2C)

Obsah přednášky

SPI komunikace (Serial Peripheral Interface)

Synchronní a asynchronní komunikace

Sériové rozhraní RS-232

IrDA přenos

I2C sběrnice

Ukázka programu v jazyce C pro ATmega16

Komunikace prostřednictvím TWI (I2C)

Programování pomocí SPI (Serial Peripheral Interface)

Pozn.: Platí obecně pro každou SPI komunikaci (A/D převodníky, paměti, čidla, ...).

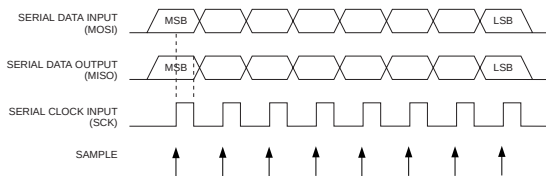
Připom.: Master $\times$ Slave.

Připom.: SPI je duplexní, synchronní přenos pomocí čtyř vodičů: *SCK* (hodinový signál), *MOSI* (Master Out Slave In), *MISO* (Master In Slave Out), $\overline{SS}$ (Slave Select).

- ▶ Spojení Master (programuje) a Slave (je programováno) obsahuje dvojici posuvných registrů + generátor hodin od nadřazeného obvodu Master.
- ▶ Postup programování (komunikace Master→Slave):
 - (1) zahájení komunikace $\overline{SS} = 0$ pro požadovaný podřízený obvod Slave,
 - (2) příprava dat k přenosu do posuvných registrů,
 - (3) přenos je řízen (synchronizován) hodinovým signálem *SCK* od programovacího obvodu Master,
 - (4) od Master ke Slave probíhá komunikace vždy po vodiči *MOSI*,
od Slave k Master probíhá komunikace vždy po vodiči *MISO*,
 - (5) po odeslání bytu – zastavení hodin, možnost generování přerušení,
 - (6) po odeslání všech dat odpojení od obvodu Slave, tj. $\overline{SS} = 1$.

Programování pomocí SPI

- Posílání jednotlivých bitů zpravidla od MSB po LSB. Mnohdy lze volit.



Obrázek: Ukázka časového diagramu SPI komunikace.

Obsah přednášky

SPI komunikace (Serial Peripheral Interface)

Synchronní a asynchronní komunikace

Sériové rozhraní RS-232

IrDA přenos

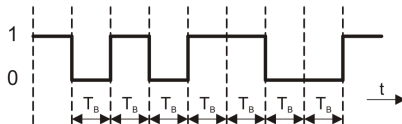
I2C sběrnice

Ukázka programu v jazyce C pro ATmega16

Komunikace prostřednictvím TWI (I2C)

Synchronní a asynchronní komunikace

- ▶ V sériové komunikaci mezi mikrokontrolérem a I/O zařízením se používají dva typy přenosů: asynchronní a synchronní.
- ▶ Asynchronní:
 - ▶ vysílač může posílat data v libovolném okamžiku s libovolně dlouhou pauzou mezi vyslanými daty (tj. doba, kdy vysílač nevysílá není pevně dána),
 - ▶ hodinový signál přijímače nemusí být synchronizován s hodinovým signálem vysílače.
- ▶ Synchronní:
 - ▶ vysílač nepřetržitě vysílá data,
 - ▶ data jsou obvykle posílána ve skupinách (blocích), přičemž jsou jednotlivé bloky odděleny synchronizačními symboly,
 - ▶ tyto symboly slouží k synchronizaci hodinového signálu přijímače a tím ke správné interpretaci vyslaných dat,
 - ▶ v případě, že vysílač nemá k dispozici žádná data, vysílá jen synchronizační symboly.
- ▶ Většina sériových přenosů dat probíhá v mikroprocesorové technice asynchronně.
- ▶ Datový signál je rozdělen na časové intervaly pro jednotlivé bity (T_B – bit time):
 - ▶ v těchto intervalech je signál ustálen v hodnotě log. 1 nebo log. 0,
 - ▶ ke změně stavu může dojít pouze na začátku nebo na konci časového intervalu T_B .



Obrázek: Dělení sériového signálu na bitové periody.

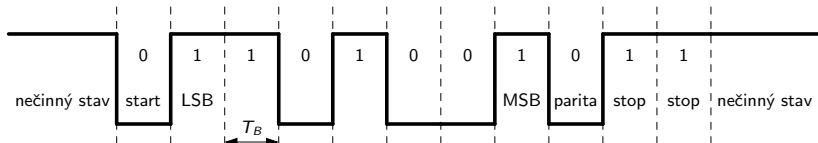
- ▶ Při asynchronní komunikaci mezi dvěma zařízeními je struktura přenášeného signálu pevně dána, tj. tvoří rámec:
 - ▶ start bit, 5 až 9 datových bitů, volitelný paritní bit, stop bit(y).

Rámec pro asynchronní přenos dat

► Formát asynchronního rámce:

- start bit je vždy ve stavu log. 0,
- přenášená informace je reprezentována 5 až 9 datovými bity v pořadí od LSB k MSB,
- volitelný paritní bit představuje snadnou detekci 1 chybného bitu. Používá se jak lichá (odd), tak i sudá (even) parita,
- stop bit je vždy ve stavu log. 1. Nejčastěji se používá 1 nebo 2. Výjimečně i 1,5 bitu, kdy je doba trvání stop bitu $1,5 \cdot T_B$,
- mód komunikace se často udává symbolicky, např. 7E2 (7 datových bitů, sudá parita, 2 stop bity), 8N1 (8 datových bitů, bez parity, 1 stop bit), apod.

Příklad asynchronního přenosu v módu 7E2



Obrázek: Struktura asynchronního rámce pro mód 7E2 (7 datových bitů, sudá parita, 2 stop bity).

► Popis časového průběhu:

- je nastaven mód se 7 datovými bity, sudou paritou a 2 stop bity. Tento mód je používán pro přenos znaků, definovaných ASCII tabulkou, (7 bitů/znak – tabulka bez národních znaků),
- v nečinném stavu je signál ve stavu log. 1,
- zahájení komunikace (start bit) je charakterizováno přechodem ze stavu log. 1 → log. 0,
- přenášený signál reprezentuje data 0b1001011, vyslaná od LSB k MSB. (V ASCII se jedná o znak "K"),
- následuje bit sudé parity. V tomto případě je roven "0", protože datové bity obsahují sudý počet jedničkových bitů,
- rámec je ukončen dvojicí stop bitů, které jsou vždy reprezentovány log. 1,
- přenos následujícího symbolu může následovat bezprostředně za posledním stop bitem.

Rychlost přenosu

- ▶ Symbolová rychlost identifikuje jak často může docházet ke změně datového signálu při sériové komunikaci; koresponduje s šířkou pásma přenosového kanálu:
$$SR = \frac{1}{\text{Cas mezi prechody}} \text{ [Bd]}.$$
- ▶ Bitová rychlost udává frekvenci s jakou jsou přenášeny jednotlivé bity
$$BR = \frac{1}{T_B} \text{ [bit/s]}.$$
- ▶ U asynchronní sériové komunikace je symbolová rychlost číselně rovna bitové rychlosti (jsou ale vyjádřeny v odlišných jednotkách).

Příklad

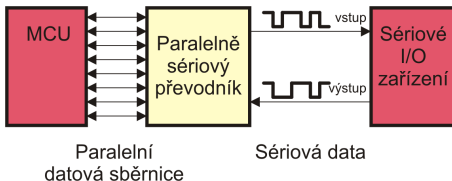
Pokud se signál mění každou 1 ms, je symbolová rychlost 1 000 Bd.

Příklad

Pokud je $T_B = 1 \text{ ms}$, je bitová rychlost 1 000 bit/s.

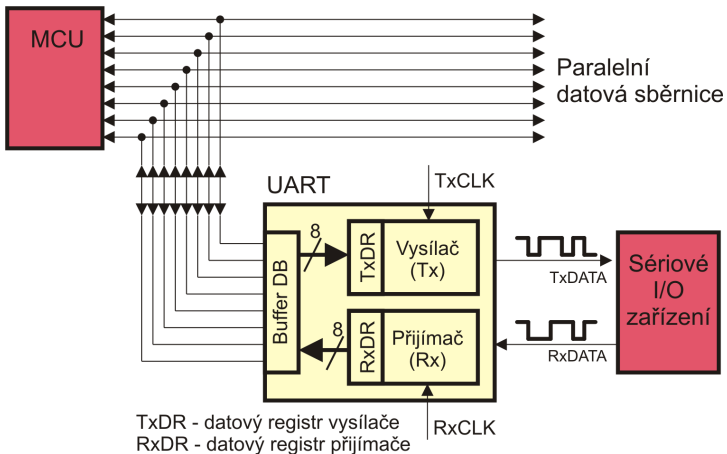
Paralelně sériový převod

- Popis funkce paralelně sériového převodníku:
 - (a) načíst 8bitová paralelní data z datové sběrnice mikrokontroléru, převést je na sériové slovo a poslat do sériového zařízení,
 - (b) načíst sériová data ze sériového I/O zařízení, konvertovat je na 8bitová a přenést do mikrokontroléru pomocí datové sběrnice.
- Paralelně/sériový převod lze provádět pomocí zařízení zvané UART (Universal Asynchronous Receiver Transmitter).



Obrázek: Princip paralelně/sériového převodníku.

Univerzální asynchronní přijímač/vysílač (UART)

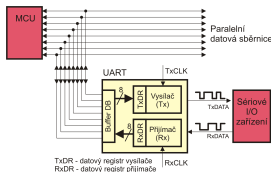


Obrázek: Vnitřní struktura jednotky UART.

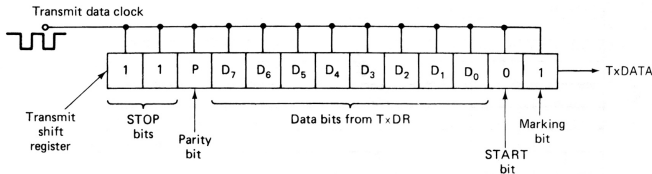
Struktura jednotky UART

- ▶ Všechna UART zařízení obsahují následující základní části.
- ▶ Sériový přijímač (Rx):
 - ▶ přijímá sériová vstupní data,
 - ▶ převádí je do paralelní reprezentace, která je uložena v přijímacím datovém registru RxDR pro případný přenos do mikrokontroléru.
- ▶ Sériový vysílač (Tx):
 - ▶ přijímá paralelní datové slova z vysílacího datového registru TxDR,
 - ▶ převádí tato slova do sériové podoby k následnému přenosu.
- ▶ Obousměrný buffer datové sběrnice:
 - ▶ zprostředkovává přenos paralelních dat od MCU do TxDR a od RxDR do mikrokontroléru prostřednictvím datové sběrnice.
- ▶ Externí hodinové signály RxCLK a TxCLK pro přijímací a vysílací část jednotky.

Proces vysílání dat pomocí UART



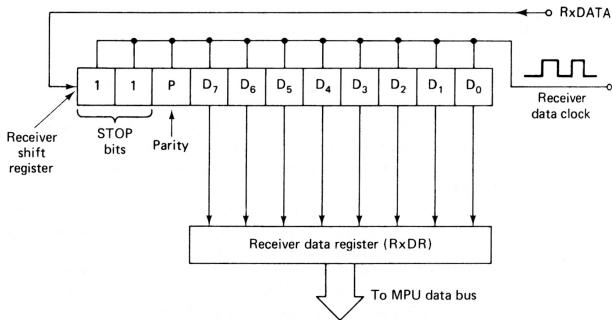
- (1) Při vysílání jsou paralelní data poslána do vysílacího datového registru TxDR.
- (2) Řídící logika v Tx doplní tato data start bitem, případným paritním bitem a požadovaným počtem stop bitů. Celý rámec je umístěn do posuvného vysílacího registru.
- (3) Obsah posuvného registru je vyslán zvolenou symbolovou rychlostí, která je řízena hodinovým signálem TxCLK. Tím je vytvořen datový výstup TxDATA (viz obrázek).



Obrázek: Proces vysílání dat pomocí jednotky UART.

Proces příjmu dat pomocí jednotky UART

- (1) Příjem sériových dat je zahájen změnou stavu na lince RxDATA. Start bit je vždy reprezentován přechodem ze stavu log. 1 → log. 0.
- (2) Zbývající symboly jsou postupně uloženy v posuvném registru, přičemž rychlost posuvu je definována hodinovým signálem RxCLK (viz synchronizace přijímače).
- (3) Po načtení celého slova jsou datové bity paralelně přesunuty do datového registru RxDR, který může být načten do MCU.

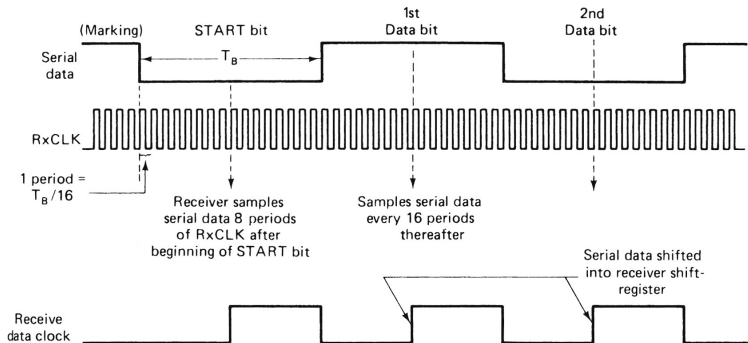


Obrázek: Proces příjmu dat pomocí jednotky UART.

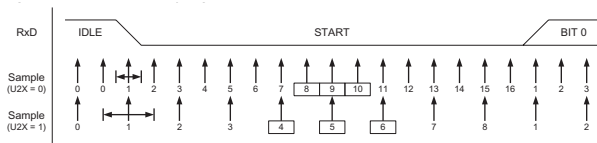
Synchronizace přijímače UART

- ▶ Pro usnadnění synchronizace přijímače sériových dat, používá UART externí hodinový signál RxCLK s podstatně vyšší frekvencí než je symbolová rychlost (zpravidla $16\times$).
- ▶ Princip synchronizace přijímače:
 - (1) po první sestupné hraně přijímač čeká 8 period hodinového signálu RxCLK. Následně je testován stav na lince, zda je stále ve stavu log. 0, tj. že se nejedná o falešný start (impulz),
 - (2) následně je vstupní signál vzorkován každých 16 period hodinového signálu RxCLK, tj. přesně uprostřed doby trvání bitové periody,
 - (3) navzorkované hodnoty jsou přesunuty do posuvného registru přijímače. Tento přesun je řízen náběžnou hranou hodinové signálu přijímače odvozeného od RxCLK (dělička kmitočtu 16),
 - (4) interní logika přijímače kontroluje počet stop bitů v přijatém rámci; pokud ne, nastaví se příznak chyby rámce (**Framing Error Flag**).
 - (5) Dále obvody UART vypočtou paritu přijatého datového slova (pokud je využívána). Tato hodnota je porovnána s hodnotou vyslanou. Chyba parity je signalizována příznakem (**Parity Error Flag**),
 - (6) Po přenosu datových bitů do registru RxDR, je možné přijímat další data do posuvného registru. Tato data již ale nejsou přesunuta do RxDR, pokud jeho původní obsah nebyla přenesen do MCU. Pokud přijímač v tomto okamžiku zachytí další rámec, je obsah v posuvném registru přepsán (ztráta 2. datového slova), což je signalizováno příznakem ztráty dat (**OverRun Error Flag**),
- ▶ Všechny jmenované chyby jsou signalizovány příslušnými příznakovými bity v řídicích registrech jednotky UART.
- ▶ Princip časové synchronizace přijímače vychází z předpokladu, že jak struktura datového rámce, tak i symbolová rychlost je TOTOŽNÁ s vysílačem.

Synchronizace přijímače UART



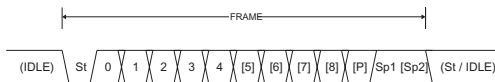
Obrázek: Synchronizace přijímače UART.



Obrázek: Princip rozhodování (arbitráž) o úrovni přijatého bitu u ATmega16 – možnost dvojí rychlosti vzorkování.

Jednotka USART u ATmega16

- ▶ Mikrokontrolér ATmega16 obsahuje vnitřní univerzální synchronní a asynchronní sériový přijímač a vysílač (USART):
 - ▶ datové signály této periférie jsou vyvedeny na piny RxD (PD0) a TxD (PD1),
 - ▶ umožňuje přenos v duplexním režimu, tj. současný příjem i vysílání dat,
 - ▶ v synchronním režimu může pracovat jako Master nebo Slave,
 - ▶ kromě normálního asynchronního režimu (Normal mode) je možné použít i režim s dvojnásobnou přenosovou rychlostí (Double Speed mode) – ukázka viz předchozí obrázek,
 - ▶ umožňuje rozmanitou volbu přenosové rychlosti, podporuje datová slova o šířce 5 až 9 bitů, volitelnou paritu, a 1 nebo 2 stop bity.



Obrázek: Datový rámec UART u mikrokontroléru ATmega16.

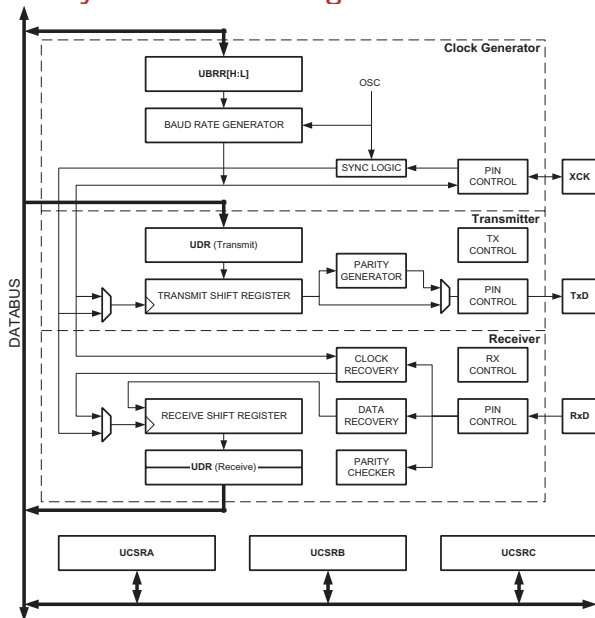
Jednotka USART u ATmega16

- ▶ Číslicová dolní propust pro filtraci zákmitů datových bitů a detekce falešných start bitů. Pro rozhodnutí přijaté úrovně se používají 3 vzorky - většina rozhodne.
- ▶ Tři zdroje přerušení: Odvysílání rámce, příjem rámce, vyprázdnění vysílacího registru.
- ▶ Jednotka USART umožňuje generování sudé/liché parity pro vysílač a testování parity u příjmu; detekce chyb Frame a OverRun Error
- ▶ Paritní bit doplňuje datové slovo na sudý/lichý počet bitů s hodnotou 1:
 - ▶ sudá parita (even):
$$P_E = d_{n-1} \oplus d_{n-2} \oplus \dots \oplus d_0 \oplus 0,$$
 - ▶ lichá parita (odd):
$$P_O = d_{n-1} \oplus d_{n-2} \oplus \dots \oplus d_0 \oplus 1.$$

Příklad

Jakým způsobem vyhodnotí asynchronní přijímač vyslaná data 0b10100, byl-li zvolen mód 5N1, ale symbolová rychlost přijímače je chybně nastavena oproti vysílací jako dvojnásobná?

Struktura jednotky USART u ATmega16



Obrázek: Bloková struktura vnitřní jednotky USART u ATmega16

Obsah přednášky

SPI komunikace (Serial Peripheral Interface)

Synchronní a asynchronní komunikace

Sériové rozhraní RS-232

IrDA přenos

I2C sběrnice

Ukázka programu v jazyce C pro ATmega16

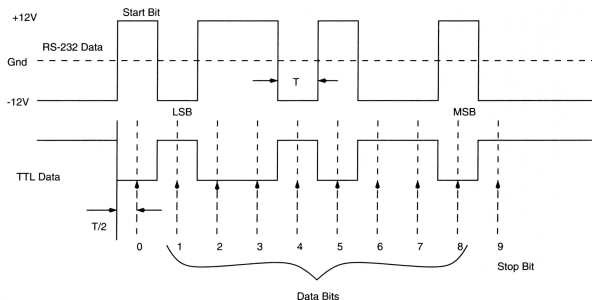
Komunikace prostřednictvím TWI (I2C)

Sériová komunikace RS-232

- Rozšířenou aplikací UART je sériová komunikace pomocí rozhraní RS-232.
- RS-232 obsahuje bipolární, inverzní reprezentaci logických úrovní (viz tab.)
- V počítačové technice se pro RS-232 využívá úrovní $\pm 12\text{ V}$ z důvodu vyšší odolnosti proti rušení.
- Maximální délka vedení závisí na symbolové rychlosti.

Tabulka: Definované intervaly napětí pro logické hodnoty RS-232.

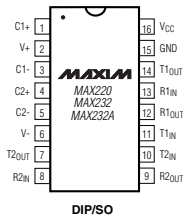
Úroveň	Vysílač	Přijímač
Log. 0	+5 V až +15 V	+3 V až +25 V
Log. 1	-5 V až -15 V	-3 V až -25 V
Nedefinováno	-3 V až +3 V	



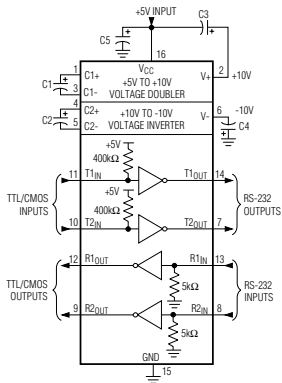
Obrázek: Úrovně totožného asynchronního rámce u RS-232 a TTL.

Zapojení sériového portu RS-232

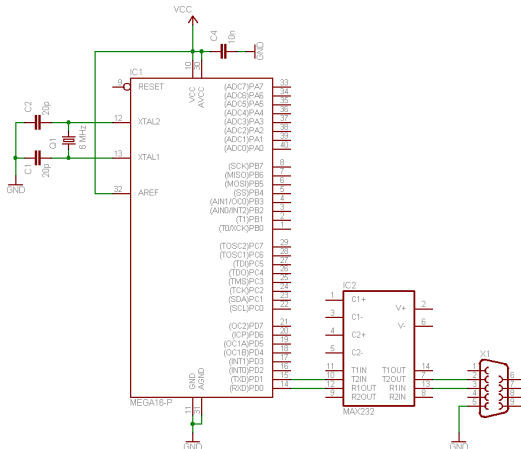
- ▶ Vzhledem k odlišným napěťovým úrovním RS-232 a MCU (typ. +5 V) není přímé propojení s TTL, CMOS možné.
- ▶ Napěťové úrovně lze sjednotit pomocí převodníků:
 - ▶ MAX232 (Maxim): konvertuje log. 0 na +3,15 V a log. 1 na -3,15 V,
 - ▶ ADM232 (Analog Devices),
 - ▶ ...



CAPACITANCE (μF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1



Propojení PC a MCU pomocí RS-232



Obrázek: Asynchronní komunikace mezi PC a MCU (ATmega16) prostřednictvím RS-232.

- ▶ Příklad aplikace. Přenos ASCII kódu klávesy z PC do mikrokontroléru.
- ▶ Zaslání potvrzení přijetí zpět do PC.

Pozn.: Program pro ovládání sériového portu PC: např.
<http://rs232.hw.cz/#Terminal>.

Obsah přednášky

SPI komunikace (Serial Peripheral Interface)

Synchronní a asynchronní komunikace

Sériové rozhraní RS-232

IrDA přenos

I2C sběrnice

Ukázka programu v jazyce C pro ATmega16

Komunikace prostřednictvím TWI (I2C)

Sériový přenos pomocí infračerveného záření

- ▶ Bezdrátový přenos dat pomocí infračerveného záření je standardizován: IrDA. Strukturu vytvořilo stejnojmenné konzorcium (Infrared Data Association) a je určen pro bezdrátové spojení na krátkou vzdálenost.
- ▶ Ve standardu je popsána jak fyzická realizace koncových zařízení, tak i komunikační protokol.
- ▶ Přenos je realizován pomocí sériové komunikace prostřednictvím infračervených diod (cca 875 nm) a PIN diod jako přijímače.
- ▶ IrDA umožňuje přenos dat rychlostí od 2,4 kbits/s do 4 Mbits/s.
- ▶ Existují dvě normy fyzické vrstvy IrDA s označením 1.0 a 1.1, které se liší přenosovou rychlostí.

Fyzická vrstva IrDA

- ▶ Norma 1.0 je označována jako SIF (Serial InfraRed):
 - ▶ přenosové rychlosti do 115,2 kbits/s,
 - ▶ dosah 1 m. Používá se shodný struktura asynchronního rámce jako u UART, liší se ale kódování jednotlivých bitů,
 - ▶ vstupní data jsou kódována pomocí kódování RZI (Return to Zero Inverted). K vysílači IrDA musí být tedy předřazen převodník datového rámce UART (NRZ) na rámec IrDA (RZI).
- ▶ Popis RZI kódování:
 - ▶ log. 0 je reprezentována krátkým pulzem s dobou trvání $\frac{3}{16} T_B$ nebo pevnou šířkou $1,63 \mu s$,
 - ▶ log. 1 je reprezentována nízkou úrovní.

Příklad

Nakreslete časový průběh signálu pro bezdrátový přenos dat 0b1001011 pomocí normy 1.0 s módem 7E1 (sudá parita).

Fyzická vrstva IrDA

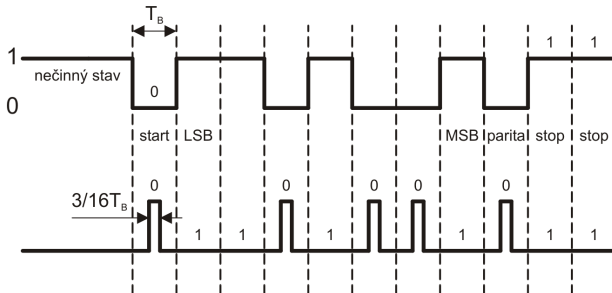
- ▶ Norma 1.1 je označována jako FIR (Fast InfraRed):
 - ▶ využívá se dvojí kódování jednotlivých bitů,
 - ▶ vstupní data pro přenosové rychlosti 0,576 Mbits/s a 1,152 Mbits/s jsou kódována pomocí RZI kódování s šířkou pulzu $\frac{1}{4} T_B$,
 - ▶ pro sériový přenos dat rychlostí 4 Mbits/s se využívá tzv. 4PPM (Pulse Positioning Modulation).
- ▶ 4PPM:
 - ▶ dvojice bitů je kódována pozicí jednoho pulzu uvnitř jedné periody T_B ,
 - ▶ vyznačuje se polovičním počtem "blikání" LED diody, což umožňuje dvojnásobnou přenosovou rychlost,
 - ▶ pro přijímač je také snadnější udržet úroveň okolního osvětlení – dopadá na něj konstantní počet pulzů za jednotku času.

Příklad

Nakreslete časový průběh signálu pro bezdrátový přenos dat 0b10011101 pomocí normy 1.1 a přenosovou rychlost 4 Mbit/s.

Pozn.: Infrared komunikační protokoly: RC-5, RC-6, ...

Kódování u IrDA



Obrázek: Kódování bitového toku pomocí UART a RZL.

Tabulka: Kódování 2bitů pomocí 4PPM.

2bit	Pozice pulzu
0b00	1000
0b01	0100
0b10	0010
0b11	0001

Obsah přednášky

SPI komunikace (Serial Peripheral Interface)

Synchronní a asynchronní komunikace

Sériové rozhraní RS-232

IrDA přenos

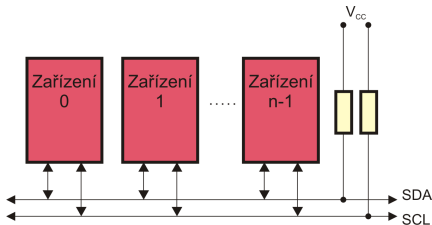
I2C sběrnice

Ukázka programu v jazyce C pro ATmega16

Komunikace prostřednictvím TWI (I2C)

Sériová sběrnice I2C

- ▶ Sběrnice I2C (Inter-Integrated Circuit) [aj skve:r si:] byla vyvinuta firmou Philips Semiconductor pro komunikaci mezi dílčími bloky jednoho zařízení.
 - ▶ Firma Atmel ji nazývá TWI (Two-wire Serial Interface). Jedná se však o totožnou komunikaci!
 - ▶ Komunikační protokol sběrnice I2C umožňuje propojení až 128 zařízení pomocí dvou obousměrných vodičů:
 - ▶ datový kanál SDA (Synchronous Data),
 - ▶ hodinový signál SCL (Synchronous Clock).
- Pozn.:** U ATmega16 se jedná o piny PC1 a PC0.
- ▶ Každý vodič sběrnice obsahuje pull-up rezistor, který zajistí vysokou úroveň signálu v klidovém stavu.



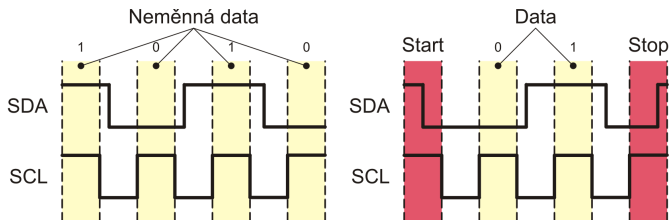
Obrázek: Připojení zařízení na sběrnici I2C.

Připojení zařízení ke sběrnici I2C

- ▶ Všechna zařízení připojená na sběrnici musí mít individuální 7bitovou adresu a implementovaný mechanismus komunikace pomocí I2C sběrnice.
- ▶ Rozdíly oproti SPI komunikaci:
 - ▶ obsahuje komunikační protokol (vysílání dat a jejich potvrzování),
 - ▶ realizována pomocí dvojice vodičů,
 - ▶ neumožňuje duplexní přenos; v jednom okamžiku vysílá jen jedno zařízení.
- ▶ Zařízení obsahují výstup s otevřeným kolektorem, nebo tří-stavový výstup.
- ▶ Nízká úroveň je na sběrnici generována jen v případě, kdy jedno nebo více zařízení nastaví svůj výstup na log. 0.
- ▶ Komunikační protokol využívá dvojí typ zařízení:
 - Master** zařízení, které zahajuje a ukončuje komunikaci na sběrnici, generuje hodinový signál SCL, a vysílá všechny požadavky,
 - Slave** podřízené zařízení, řízené hodinovým signálem a vždy adresované Masterem.
- ▶ Počet připojených zařízení je limitován pouze celkovou kapacitou sběrnice (typ. 400 pF), nebo 7bitovou adresou podřízených obvodů.
- ▶ Komunikace je zahájena startovací podmínkou (Start Condition) a ukončena ukončovací podmínkou (Stop Condition).
- ▶ Komunikační protokol definuje dvojí typ rámců: adresní a datové.

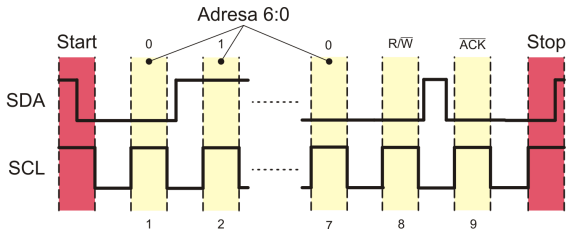
Startovací a ukončovací podmínka

- ▶ Přenos každého bitu je řízen jedním pulsem hodinového signálu SCL:
 - ▶ platí, že úroveň datové linky SDA musí být konstantní v okamžiku vysoké úrovně hodinového signálu,
 - ▶ jedinou výjimkou je generování startovací a ukončovací podmínky,
 - ▶ Start i Stop je signalizován změnou stavu datové linky SDA během doby, kdy je hodinový signál SCL na vysoké úrovni (viz obrázek).



Obrázek: Časování přenosu bitů na sběrnici I2C, startovací a ukončovací podmínky.

Adresní paket na sběrnici I2C



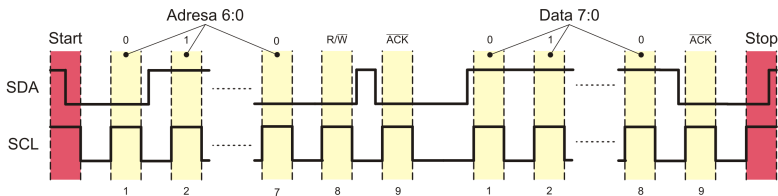
Obrázek: Struktura adresního paketu na sběrnici I2C.

► Formát adresního paketu (rámce):

- všechny adresní pakety jsou 9bitové a obsahují 7bitovou adresu podřízeného zařízení, jeden $R/\overline{W}$ řídicí bit ($R/\overline{W} = 1$: čtení) a potvrzovací bit ($\overline{ACK}$ – Acknowledge bit),
- potvrzovací bit posílá zařízení Slave v případě, že rozpoznalo svou adresu, tím že nuluje stav signálu SDA během 9. hodinového cyklu SCL.
- pokud je Slave zaneprázdněn, zůstane signál SDA=1 (což je identifikováno jako NACK – Not Acknowledge) a Master ukončí komunikaci ukončovací podmínkou,
- adresní pakety se schématicky označují jako SLA+R v případě následného čtení a SLA+W, pokud je následně vyžadován zápis dat do Slave zařízení,
- nejvýznamnější adresní bit (MSB) je odeslán jako první. Adresa zařízení je vždy implementována již výrobcem (viz katalogové listy) a musí být různá od 0,
- část adresy je možné specifikovat také pomocí úrovně signálů na několika externích pinech součástky. Např. pro expandér portu PCF8574 je adresa 0b0100A2A1A0.

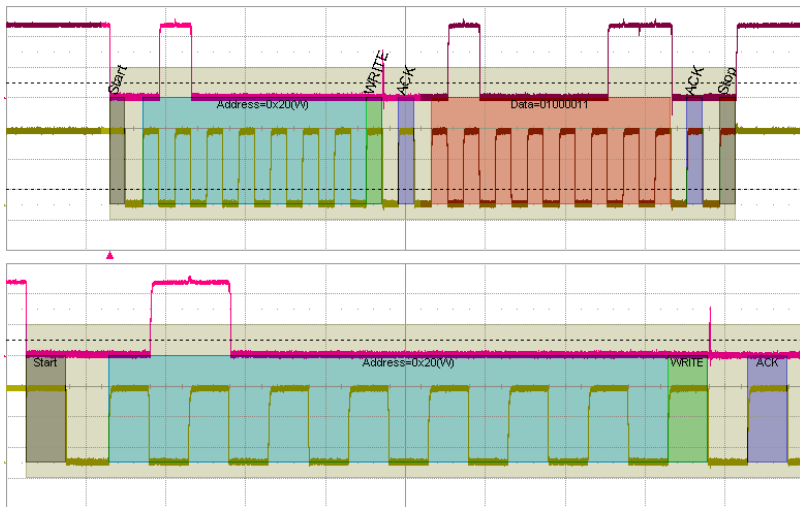
Datový paket na sběrnici I2C

- ▶ Formát datového paketu (rámce):
 - ▶ všechny datové pakety přenášené po sběrnici I2C jsou 9bitové a obsahují 1bytová data a jeden potvrzovací bit $\overline{ACK}$ od Slave zařízení během 9. SCL cyklu (shodné s adresním paketem),
 - ▶ nejvýznamnější bit (MSB) je opět vyslán jako první.
- ▶ Kombinace adresního a datového paketu:
 - ▶ typická komunikace na sběrnici I2C začíná startovací podmínkou, následuje adresování obvodu Slave SLA+R/W, jeden nebo více datových paketů a ukončení komunikace ukončovací podmínkou.



Obrázek: Kombinace adresního a datového paketu.

Komunikace s expandérem portu PCF8574

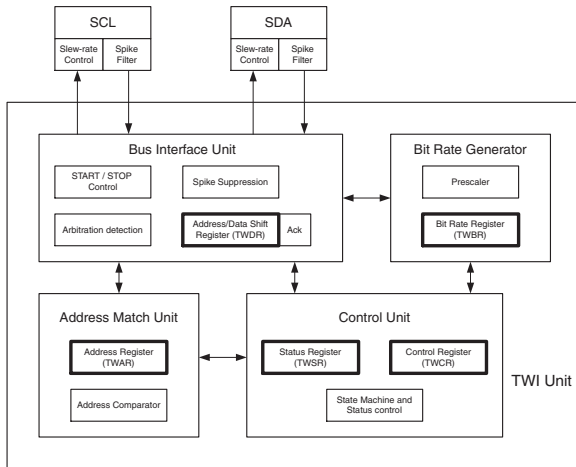


Obrázek: Vyslání adresního a datového paketu po sběrnici I2C. Adresa expandéru PCF8574 je 0b0100000.

Komunikační módy sběrnice I2C

- ▶ Komunikace mezi zařízením Master a Slave může probíhat ve 4 základních módech:
 - ▶ MT (Master Transmitter) – Master vysílá,
 - ▶ MR (Master Receiver) – Master přijímá,
 - ▶ ST (Slave Transmitter) – Slave vysílá,
 - ▶ SR (Slave Receiver) – Slave přijímá.
- ▶ V některých případech, musí být módy kombinovány. Např. čtení dat ze sériové paměti EEPROM:
 - (1) zahájení komunikace,
 - (2) adresace zařízení (MT),
 - (3) adresace požadované paměťové buňky (MT),
 - (4) proces čtení vybrané buňky (MR),
 - (5) ukončení komunikace.

Struktura jednotky TWI u ATmega16

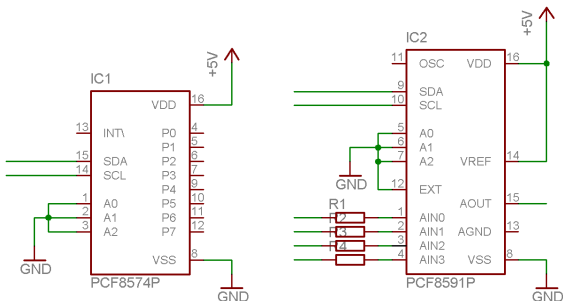


Obrázek: Blokova struktura vnitřní jednotky TWI u ATmega16.

- Možnost generování přerušení při libovolné události na sběrnici, např. při startovací podmínce, přenosu adresy zařízení SLA+R/W, přijetí datového bytu, ukončovací podmínce, ...

Příklady obvodů komunikující pomocí I2C

- ▶ **Expandér portu PCF8574:**
 - ▶ obousměrný expandér umožňuje přidat do systému 8bitový port,
 - ▶ vyšší část adresy je pevně dána 0b0100. Nejnižší bity je možné specifikovat pomocí pinů A2 až A0.
- ▶ **AD/DA převodník PCF8591:**
 - ▶ 4kanálový, 8bitový A/D převodník,
 - ▶ obsahuje jeden řídicí registr pro nastavení vstupního, příp. výstupního kanálu, automatické přepínání, diferenční převod, . . .



Obrázek: Expandér portu a AD/DA převodník

Příklady obvodů komunikující pomocí I2C

- ▶ Obvod reálného času PCF8563TD:
 - ▶ uchovává informaci o datumu a čase na základě interního oscilátoru s frekvencí 32,768 kHz,
 - ▶ napájecí napětí 1,8 V až 5,5 V,
 - ▶ datová komunikace pomocí I2C s maximální rychlostí 400 kHz,
 - ▶ výstup programovatelné frekvence hodinového signálu pro případná externí zařízení (32,768 kHz, 1 024 Hz, 32 Hz, 1 Hz),
 - ▶ funkce časovače a alarmu,
 - ▶ I2C Slave adresa: 0b1010001.

Příklad

Nakreslete časový průběh signálů na I2C sběrnici při čtení minut z obvodu reálného času PCF8563 (nechť adresa registru s minutami je 0b01000101).

Obsah přednášky

SPI komunikace (Serial Peripheral Interface)

Synchronní a asynchronní komunikace

Sériové rozhraní RS-232

IrDA přenos

I2C sběrnice

Ukázka programu v jazyce C pro ATmega16
Komunikace prostřednictvím TWI (I2C)

Komunikace prostřednictvím TWI (I2C)

- ▶ Aplikace s jednorázovým vysláním adresního a datového paketu na sběrnici.
- ▶ Bez přerušení.
- ▶ Nastavení bitové rychlosti:

$$f_{SCL} = \frac{f_{CPU}}{16 + 2 \cdot TWBR} \quad (1)$$

Komunikace prostřednictvím TWI (I2C)

```
1 #include <avr\io.h>           // hlavičkový soubor pro mikrokontrolér
2 #include "twi.h"              // knihovna pro sběrnici I2C (TWI)
3
4 int main( void ){             // hlavní funkce aplikace
5
6     TWCR = ( 1<<TWEN ) ;      // zapnutí jednotky TWI
7     TWBS = 152 ;              // bitová rychlost 50kbit/s
8
9     twi_start() ;              // vyslání startovací podmínky
10    twi_address_w( ... ) ;      // vyslání adresy SLA+W
11    twi_write( 0b01000011 ) ;   // vyslání datového paketu
12    twi_stop() ;               // vyslání ukončovací podmínky
13
14    while( 1 ) ;               // nekonečná smyčka
15    return( 1 ) ;              // návratová hodnota funkce 1
16 }
```