



Komunikace zařízení po sběrnici

Mikroprocesorová technika a embedded systémy

Přednáška 4

doc. Ing. Tomáš Frýza, Ph.D.

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů

Napájení, hodinový signál, obvod reset

Programovatelné propojky (Fuses)

Ukázka programu v JSA a v jazyce C pro ATmega16

Časovač watchdog

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů

Napájení, hodinový signál, obvod reset

Programovatelné propojky (Fuses)

Ukázka programu v JSA a v jazyce C pro ATmega16

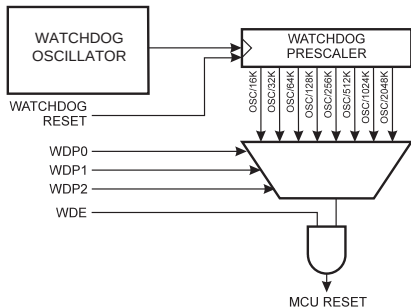
Časovač watchdog

Obvod watchdog

- ▶ Obvod watchdog je "bezpečnostní časovač". Jeho význam je vyvolat reset (a tím i znovu-nastartování) mikrokontroléru v případě, že se procesor "ztratí", příp. nevykonává program, který by měl.
- ▶ Nazávislý časovač s odděleným hodinovým systémem (frekvence je závislá na napájení; $V_{CC} = 5\text{ V} \Leftrightarrow f_{WDT} = 1\text{ MHz}$).
- ▶ Běžný chod programu: program opakovaně znovunačítá, příp. resetuje stav časovače (např. v nekonečné smyčce programu) a tím nedojde k přetečení jeho hodnoty.
- ▶ V případě "ztráty" programu, zacyklení, apod. dojde k přetečení a k vyvolání resetu.

Watchdog periférie

- Funkce watchdog časovače je běžně vypnuta, tj. není potřeba se obávat nechtěného resetu systému.
- Povolení generace resetu MCU a nastavení předděličky je řízeno řídicím registrem WDTCR (Watchdog Timer Control Register).



Obrázek: Struktura Watchdogu.

Tabulka: Příklad změny rychlosti čítání watchdogu (reg. WDTCR)

WDP2:WDP0	Přetečení ($V_{CC} = 5\text{ V}$)
0b000	16 ms
0b001	33 ms
0b010	65 ms
0b011	130 ms
0b100	260 ms
0b101	520 ms
0b110	1 s
0b111	2,1 s

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů

Napájení, hodinový signál, obvod reset

Programovatelné propojky (Fuses)

Ukázka programu v JSA a v jazyce C pro ATmega16

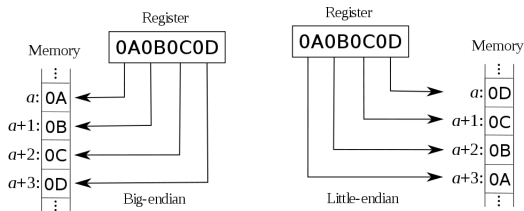
Časovač watchdog

Základní pojmy mikroprocesorové techniky

- ▶ Bit – zkratka z "Binary Digit", tj. jeden symbol v binární soustavě:
 - ▶ označení "b". Např. 8 b,
 - ▶ nastavení bitu (set bit) hodnota bitu = 1 (vysoká úroveň),
 - ▶ nulování bitu (clear bit) hodnota bitu = 0 (nízká úroveň).
- ▶ Paměťová buňka zařízení nebo elektrický obvod pro uchování hodnoty 1 bitu (klopný obvod, stav tranzistoru, jeden "spot" na CD, magnetický náboj, ...).
- ▶ Paměťové slovo množina paměťových buněk určité velikosti (registr z osmi klopných obvodů, ...); typická velikost od 4 do 64 bitů.
- ▶ Paměť součástka, umožňující uložení programu nebo dat v binární podobě. Soubor určitého počtu paměťových slov.
- ▶ Registr – skupina paměťových buněk k uložení binární informace. Kratší přístupová doba než u ostatních paměťových slov.
- ▶ Šířka registru – počet bitů v registru.
- ▶ Registrový pár – dvojice registrů.
- ▶ Pin – vodič vyvedený vně součástky.
- ▶ Port – skupina pinů (nejčastěji 8), umožňující vstupně/výstupní komunikaci mikrokontroléru s okolím.
- ▶ Sběrnice – skupina paralelních vodičů umožňující propojení vstupů a výstupů několika zařízení, apod.

Základní pojmy a terminologie

- ▶ Byte [bajt] – slovo o šířce 8 bitů; jedno z nejpoužívanějších paměťových slov v mikroprocesorové technice:
 - ▶ označení "B". Např. 16 B,
- ▶ Nibl [nibl] – slovo o šířce 4 bitů; polovina bytu; dříve hojně využíváno.
- ▶ MSB (Most Significant Bit/Byte) – nejvýznamnější bit/byte.
- ▶ LSB (Least Significant Bit/Byte) – nejméně významný bit/byte.
- ▶ Big-endian, Little-endian – pořadí ukládání bytů do paměti.



Obrázek: [Wiki].

- ▶ Paměťová kapacita – celkový počet bitů v paměti, nebo v systému. Udává se ve tvaru počet slov $\times$ šířka slova:
 - ▶ např. $8k \times 16$, $1k \times 8$, 512×8 .
- ▶ Pozn.: Pracujeme ve dvojkové soustavě, proto:
 - ▶ $1k [\text{kilo}] = 2^{10} = 1024$ (pozor: $1k \neq 1000$),
 - ▶ $1M [\text{mega}] = 2^{20} = 1\,048\,576$,
 - ▶ $1G [\text{giga}] = 2^{30} = 1\,073\,741\,824$.

Základní pojmy a terminologie

Příklad

Co představuje zápis paměťové kapacity $4\,096 \times 20$?

Řešení

- ▶ Počet slov = $4\,096$ ($4k$),
- ▶ šířka slova, tj. počet bitů na slovo = $20\,b$.

Příklad

Nechť je paměťová kapacita vyjádřena hodnotou $2k \times 8$

- (a) *Kolik slov obsahuje tato součástka?*
- (b) *Jaká je šířka jednoho slova?*
- (c) *Jaký je celkový počet bitů v paměťové součástce?*

Základní pojmy a terminologie, paměti

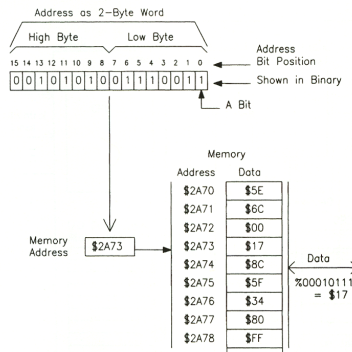
- ▶ Adresa – index, který identifikuje pozici slova v paměti; každé slovo má unikátní adresu; nejčastěji se udává v šestnáctkové soustavě (prefix 0x nebo \$).
- ▶ Čtení z paměti (anglicky Fetch) – operace, při které je binární slovo (např. *slovo 10* na obrázku) vyzvednuto z konkrétní adresy v paměti (0x0A) a přeneseno na jiné místo.
- ▶ Zápis do paměti (anglicky Store) – operace, při které je nové slovo (např. *slovo 1* na obrázku) přeneseno na konkrétní pozici v paměti (0x01); původní informace je ztracena.

0x00	slovo 0
0x01	slovo 1
0x02	slovo 2
...	...
0x09	slovo 9
0x0a	slovo 10
...	...
0x0e	slovo 14
0x0f	slovo 15

Obrázek: Uspořádání paměti.

Ukázka adresování paměťového slova

- ▶ Ukázka adresování 8bitového slova pomocí 16bitové adresy.
- ▶ MSB tvoří vyšší byte adresy, LSB tvoří nižší byte adresy
 - ▶ MSB: 0b0010 1010 @ \$2A,
 - ▶ LSB: 0b0111 0011 @ \$73.
- ▶ Všechna paměťová slova jsou stejně široká. Adresovaná data:
 - ▶ 0b0001 0111 @ \$17.



Obrázek: Adresování paměťového slova.

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů

Napájení, hodinový signál, obvod reset

Programovatelné propojky (Fuses)

Ukázka programu v JSA a v jazyce C pro ATmega16

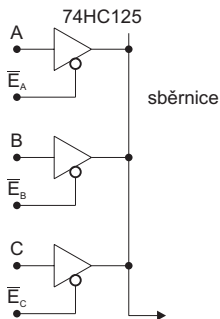
Časovač watchdog

Komunikace mezi zařízeními

- ▶ Typická komunikace uvnitř mikrokontroléru, ale i mezi mikrokontrolérem a zařízeními probíhá po paralelním vedení, tj. po sběrnici (u 8bitových systémů tvořena zpravidla 8 vodiči).
- ▶ Všechna zařízení obsahují 8bitový vstupně/výstupní port, pomocí něhož jsou připojena na jedinou skupinu vodičů sběrnici. Nejedná se tedy o spojení point-point.
- ▶ Aby byly eliminovány kolize na sběrnici, kdy několik zařízení vysílá současně – stav na sběrnici je dán kombinací více signálů; data jsou "nečitelná" – se používají systémy s tří-stavovými výstupy (Three State), příp. oddělovací bufery s třístavovým výstupem.
- ▶ Tří-stavový výstup je řízen povolovacím vstupem (Enable), který zajistí aktivaci vždy jen jednoho zařízení, výstupy ostatních jsou ve stavu vysoké impedance.

Tří-stavový výstup, dokončení

- Tří-stavové buffery se běžně používají k připojení několika zařízení/signálů ke sběrnici.



Obrázek: Tři odlišné buffery pro signály A, B a C.

- V případě vysílání více zařízení, dojde ke kolizi na sběrnici (nežádoucí). Řídící signály $\overline{EN}$ musí zajistit, aby ke sběrnici bylo připojeno jen jedno zařízení.

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů

Napájení, hodinový signál, obvod reset

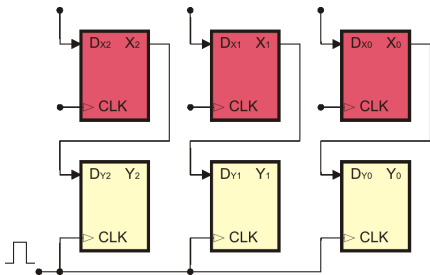
Programovatelné propojky (Fuses)

Ukázka programu v JSA a v jazyce C pro ATmega16

Časovač watchdog

Paralelní přenos dat mezi registry

- ▶ Registry hrají významnou úlohu v mikroprocesorové technice. Ze softwarového pohledu je mikrokontrolér systémem registrů, mezi kterými je binární informace přesunována a zpracovávána.
- ▶ Existují dva způsoby přesunu dat mezi registry:
 - ▶ paralelní, sériový.

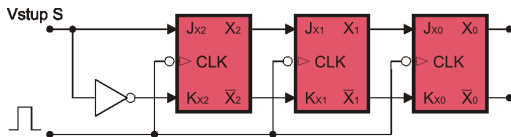


Obrázek: Paralelní přenos dat.

- ▶ Ukázka: Dvojice 3bitových registrů X a Y, které jsou tvořeny klopnými obvody typu D, řízených hranou.
- ▶ Přesun dat je řízen hodinovým signálem CLK.

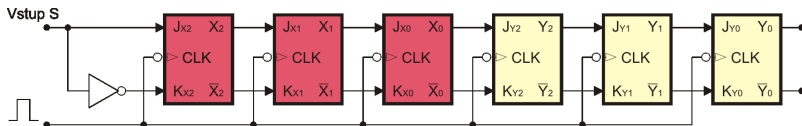
Paralelní a sériový přenos dat mezi registry

- Symbolický zápis přenosu dat: $[X] \rightarrow [Y]$ (přesun obsahu registru X do registru Y).
- Hodinový pulz aplikován na vstupy CLK způsobí přesun dat z výstupů X_2, X_1, X_0 (vstupy D_Y) na výstupy Y_2, Y_1, Y_0 .
- Jedná se o paralelní přesun, protože celý obsah registru X se překopíruje do Y najednou (pomocí jedné hrany řídicího signálu).
- Ukázka: Sériový přenos dat prostřednictvím 3bitového posuvného registru (klopné obvody typu JK, řízené hranou).



Obrázek: Posuvný (shift) registr.

Sériový přenos dat mezi registry



Obrázek: Sériový přenos dat.

- ▶ Při aktivní hraně signálu CLK jsou hodnoty vstupních signálů klopných obvodů překopírovány na výstupy (tj. na vstupy obvodů vpravo).
- ▶ Data z výstupu posledního KO jsou ztracena. Sériový přenos vyžaduje více času (jeden bit vyžaduje jeden pulz), ale méně vodičů.

Sériový přenos dat mezi registry, příklad

Příklad

Přenos dat $[X] \rightarrow [Y]$; necht' $[X]=101$, $[Y]=011$ a $S = 0$

Řešení

1. pulz: $[X]=010$, $[Y]=101$
2. pulz: $[X]=001$, $[Y]=010$
3. pulz: $[X]=000$, $[Y]=101$

Tabulka: Postupné posouvání dat.

S	X_2	X_1	X_0	Y_2	Y_1	Y_0	
0	1	0	1	0	1	1	před 1. pulzem
0	0	1	0	1	0	1	po 1. pulzu
0	0	0	1	0	1	0	po 2. pulzu
0	0	0	0	1	0	1	po 3. pulzu

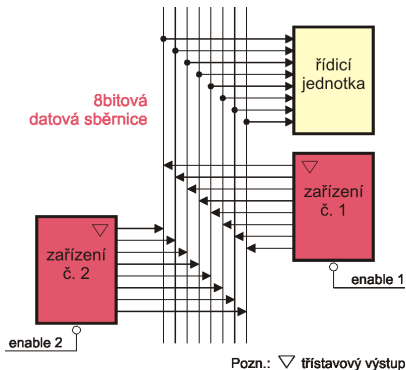
Komunikace mezi zařízeními

Příklad

Jakým způsobem zajistit bezpečný přenos dat ze zařízení č. 2 do řídicí jednotky? (Nechť řídicí jednotka v tomto případě shromažďuje informace od jednotlivých zařízení).

Příklad

Jaké napěťové úrovně budou na datové sběrnici v případě, že všechna zařízení budou ve stavu vysoké impedance?



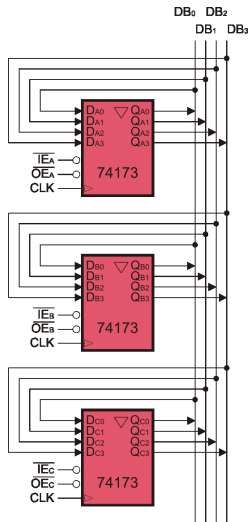
Obrázek: Komunikace po sběrnici.

Příklad přenosu dat po sběrnici

- ▶ Příklad zapojení tří 4bitových registrů 74173 (4bitový 3stavový klopný obvod typu D) na 4bitovou sběrnici, tvořenou vodiči DB_3 – DB_0 .
- ▶ Registry obsahují tři řídicí signály:
 - ▶ $\overline{IE}$ (Input Enable) – nízká úroveň aktivuje vstupní obvody,
 - ▶ $\overline{OE}$ (Output Enable) – nízká úroveň aktivuje výstupní obvody,
 - ▶ CLK – hodinový signál.

Příklad

Zajistěte přenos dat z registru B do registru A, $[B] \rightarrow [A]$. Necht' $[B] = 1100$.



Obrázek: 4bitové registry 74173.

Příklad přenosu dat po sběrnici, řešení

Řešení

- ▶ *Výstupy:*
 - ▶ *pouze registr B musí mít aktivní výstupní obvody, tj. $\overline{OE}_A = 1$, $\overline{OE}_C = 1$, $\overline{OE}_B = 0$,*
 - ▶ *obsah registru B je tím zapsán na datovou sběrnici.*
- ▶ *Vstupy:*
 - ▶ *pouze registr A musí mít aktivní vstupní obvody, tj. $\overline{IE}_A = 0$, $\overline{IE}_B = 1$, $\overline{IE}_C = 1$.*
- ▶ *Následující aktivní hrana hodinového signálu zajistí přenos (přečtení) dat ze sběrnice do klopných obvodů registru A.*
- ▶ *Přenos dat je tedy zajištěn generováním korektní posloupností řídicích signálů – zajišťuje řídicí jednotka.*

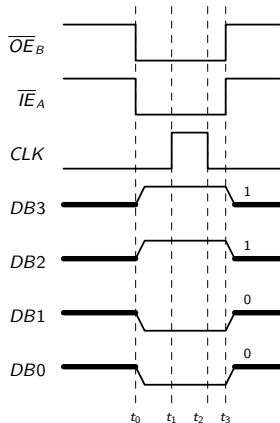
Příklad přenosu dat po sběrnici, řešení

Řešení

- Platná data jsou v registru A přeneseny v časovém intervalu $t_1 - t_2$.

Příklad

Proč je aktivní úroveň řídicích signálů v nízké úrovni?



Obrázek: Časové průběhy na řídicí a datové sběrnici.

Rozšíření sběrnice

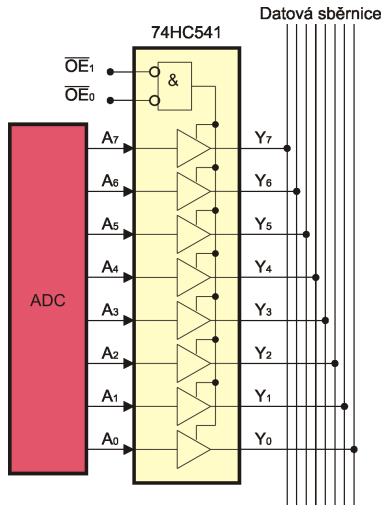
- ▶ Běžně se používá větší počet registrů než 3. Obecně lze na sběrnici připojit libovolný počet registrů. Zvyšuje se tím ale počet řídicích signálů $\overline{OE}$, $\overline{IE}$ a vstupně/výstupních vodičů.
- ▶ Obdobně jako u 4bitové sběrnice fungují sběrnice 8bitové, 16bitové, nebo 32bitové.
- ▶ Zobecněný princip:
 - ▶ jeden registr (paměťová buňka, zařízení) má aktivované výstupní obvody a zapisuje data na sběrnici,
 - ▶ jiné zařízení má aktivované vstupní obvody, čímž může data ze sběrnice číst,
 - ▶ při aktivní hraně hodinového signálu dojde k zápisu dat do registru – synchronní řízení.
- ▶ Šířka sběrnice (počet vodičů) je dána šířkou datového slova, přenášeného v systému. 8bitové slovo $\Rightarrow$ 8 vodičů, 16bitové slovo $\Rightarrow$ 16 vodičů, ... Některé systémy obsahují kombinaci několika různě-širokých sběrnic.
- ▶ Počet zařízení, připojených na sběrnici je rozdílný případ od případu. Závisí na velikosti paměti systému a na počtu vstupních a výstupních zařízeních, které musí s CPU komunikovat po sběrnici.
- ▶ Všechna zařízení musí být připojena přes 3stavové buffery:
 - ▶ některá zařízení tyto buffery již obsahují přímo na čipu, např. registry 74173 (viz výše),
 - ▶ ostatní zařízení musí být ke sběrnici připojeny přes tzv. řadič sběrnice (Bus Driver).

Využití řadiče sběrnice

- ▶ Řadič sběrnice obsahuje 3stavový výstup s velmi nízkou výstupní impedancí, což způsobuje rychlé nabíjení a vybíjení celkové kapacitní reaktance (odpor) sběrnice $\Rightarrow$ krátké doby přechodu. (Řadič sběrnice tedy nezhoršuje doby přechodu.)
- ▶ Kapacitní odpor je tvořen kumulací parazitních kapacitancí všech vstupů a výstupů, připojených ke sběrnici a může zhoršit doby přechodu.
- ▶ Příklad řadiče sběrnice – 8bitový 74HC541: připojení výstupu 8bitového A/D převodníku k datové sběrnici pomocí 74HC541.

Tabulka: Funkční tabulka.

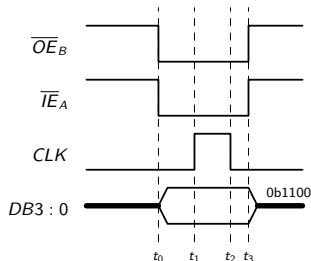
$\overline{OE}_1$	$\overline{OE}_0$	A_n	Y_n
0	0	0	0
0	0	1	1
1	X	X	Z
X	1	X	Z



Obrázek: Řadič sběrnice

Zjednodušená reprezentace časových průběhů

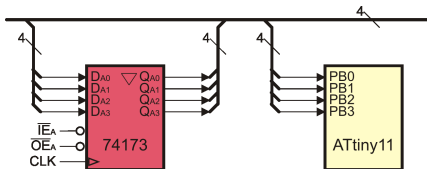
- ▶ Časové průběhy jednotlivých signálů na sběrnici lze z důvodu přehlednosti zápisu sdružit do jediného průběhu – výhodné u systémů s 8, 16, nebo s 32 vodiči.
- ▶ Ukázka: Přenos dat ze zařízení B do A, viz předešlý případ [B]=1100:
 - ▶ $\overline{OE}_B = 0$; $\overline{IE}_A = 0$,
 - ▶ ostatní řídicí vstupy/výstupy nejsou zakresleny.
- ▶ Platná data na sběrnici v době trvání aktivní úrovně hodinového impulsu (interval $t_2 - t_1$).
- ▶ Neplatná data v intervalech $t_1 - t_0$ a $t_3 - t_2$.



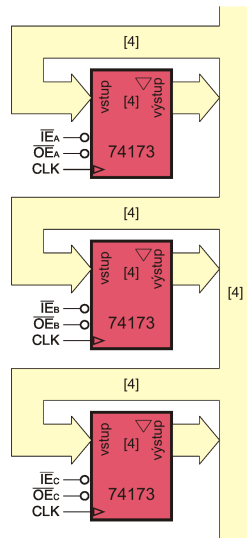
Obrázek: Zjednodušené časové průběhy na datové sběrnici.

Zjednodušená reprezentace sběrnic

- ▶ Z důvodu připojení velkého množství zařízení ke sběrnici se používá zjednodušené znázornění vodičů ve schématech:
 - ▶ pomocí širokých šipek; číslo v [-] znázorňuje šířku registrů/počet připojených vodičů,
 - ▶ pomocí jediného vodiče (fyzicky nespojeno) s indikovanou šířkou (počtem bitů) \4.



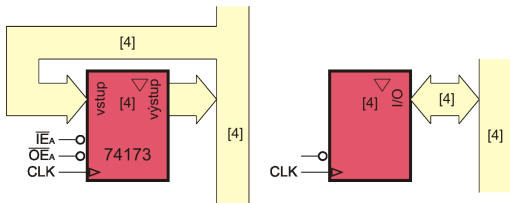
Obrázek: 4bitový registr 74173 a MCU ATtiny11.



Obrázek: Zjednodušené znázornění sběrnice.

Obousměrná sběrnice

- ▶ Uvažovaný obvod (registr 74173) má vstupy i výstupy napojené na shodný vodič datové sběrnice, např. vstupní signál D_{C0} je propojen s výstupem O_{C0} přes vodič DB_0 (neplatí při použití řadiče sběrnice):
 - ▶ D_{C0} – vstup č. nula registru C,
 - ▶ O_{C0} – výstup č. nula registru C.
- ▶ Z důvodu snížení počtu pinů, bylo vyvinuto uspořádání s propojením vstupů a výstupů uvnitř integrovaných obvodů.
- ▶ Každý pin (I_0/O_0 až I_n/O_n) může pracovat jako vstupní nebo výstupní v závislosti na řídicích signálech. Proto jsou piny I/O nazývány jako obousměrné datové linky.
- ▶ Mnoho paměťových obvodů a mikroprocesorů samozřejmě umožňuje obousměrný přenos.



Obrázek: Propojení I/O pinů uvnitř obvodu.

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů

Napájení, hodinový signál, obvod reset

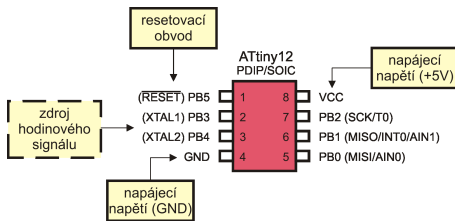
Programovatelné propojky (Fuses)

Ukázka programu v JSA a v jazyce C pro ATmega16

Časovač watchdog

Pomocné obvody řídicí (embedded) aplikace

- ▶ Není vždy nutné využívat výkonné mikrokontroléry. U jednoduchých aplikací postačí např. ATtiny12 (pouzdro PDIP8).
- ▶ Jednoduché řídicí aplikace s mikrokontroléry vyžadují (po hardwarové stránce):
 - ▶ zdroj napájení,
 - ▶ zdroj hodinového signálu,
 - ▶ resetovací obvod.

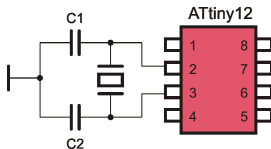


Obrázek: Blokové zapojení řídicí aplikace.

- ▶ Typ napájení závisí na mobilní/desktopové aplikaci:
 - ▶ napájení z baterií, z rozvodné sítě s transformací, z datových portů PC (RS-232, USB), pomocí solárních panelů, . . .

Pomocné obvody řídicí aplikace, pokračování

- ▶ Hodinový signál procesoru řídí časování AVR jádra. Od něho jsou odvozeny časovací signály pro všechny periférie (Flash, A/D převodník, I/O modul, čítač, ...).
- ▶ Obecné možnosti generování hodinového signálu:
 - ▶ externí zdroj hodinového signálu na vstupním pinu XTAL1,
 - ▶ vnější RC oscilátor: RC oscilátor způsobuje nízkou časovou přesnost,
 - ▶ interní RC oscilátor: nízká časová přesnost; běžné hodnoty jsou 1, 2, 4 a 8 MHz, závislost přesnosti oscilací na napětí, nejsou potřeba žádné externí součástky.
 - ▶ interní hodinový generátor s externím krystalem: nejrozšířenější varianta; vstupy XTAL1 a XTAL2 představují vstup a výstup invertujícího zesilovače oscilátoru; vyžaduje jen dva kondenzátory (typické hodnoty od 12 pF do 22 pF).



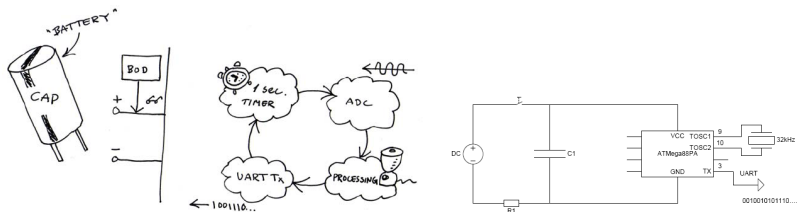
Obrázek: Externí krystal.

Pozn.: Zdroj hodinového signálu je vybrán na základě tzv. programovatelné propojky (viz dále). Propojky je možné reprogramovat.

ATmega16 je dodáván s přednastaveným interním RC oscilátorem o frekvenci 1 MHz.

Snižování spotřeby

- Vhodnou změnou softwaru lze snížit proudový odběr; vyšší životnost baterií.



Obrázek: Testovací aplikace s ATmega88PA [AVR4013].

- Viz např. Application Note AVR4013: picoPower Basics:
 - jeden A/D převod,
 - simulace 1000 fází "zpracování signálu",
 - převod hodnoty na ASCII řetězec,
 - vyslání dat přes UART,
 - a za 1 vteřinu znovu ...

Snižování spotřeby

Tabulka: Doba trvání testovací aplikace.

Metody redukce spotřeby	Životnost baterie
Žádná	6 s
Ošetření nepoužitých pinů a interních periférií	9 s
Předdělička hodinového signálu	40 s
Režimy snížené spotřeby	198 s
Redukce doby v aktivním módu	217 s

- (1) $f_{CPU} = 8 \text{ MHz}$, UART 19 200 Bd, polling (opakované dotazování) časovače 2, zda uplynula 1 s.
- (2) Použití pull-up rez. u nevyužitých pinů – nepřeklápí se; odpojení napájení od nevyužívaných periférií (SPI, TWI, Timer0, Timer1).
- (3) Snížení frekvence $f_{CPU} = 2 \text{ MHz}$, UART 19 200 Bd.
- (4) Během čekání na další zpracování → sleep mód (idle, sleep, power save).
- (5) Softwarové kalibrování oscilátoru pro zvýšení přenosové rychlosti 115,5 kbps; předdělička CPU zůstává.

Snížování spotřeby – ukázka některých funkcí avr-libc

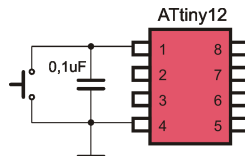
```
1  /* init application */
2  #include <avr/io.h>           // header file for ATtiny44A microcontroller
3  #include <avr/sleep.h>        // sleep_enable(), sleep_disable(), sleep_cpu()
4  #include <avr/power.h>        // see google: ``avr libc power''
5  ...
6
7  power_timer0_disable() ;      // turn off timer/counter0
8  power_timer1_disable() ;      // turn off timer/counter1
9  power_adc_disable() ;         // turn off ADC
10 power_usi_enable() ;           // turn on usi (Universal Serial Interface)
11
12 /* prescale system clock from 8 MHz to 500 kHz */
13 clock_prescale_set( clock_div_16 ) ;
14
15 /* set sleep mode to Power-save */
16 set_sleep_mode( SLEEP_MODE_PWR_SAVE ) ;
17
18 /* enable sleep (possible to put the device into sleep mode
19  * when executing the sleep instruction)
20  */
21 sleep_enable() ;
22
23 sleep_bod_disable() ;          // disable BOD while in sleep mode
24 sleep_cpu() ;                  // execute sleep instruction
25 ...
```

Reset systému

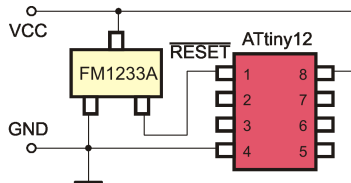
- ▶ Přerušení reset má nejvyšší prioritu může přerušit již běžící obsluhy všech přerušení, nebo vykonávající program.
- ▶ Události vyvolávající reset systému:
 - ▶ úroveň na resetovacím pinu mikrokontroléru je nižší než práh V_{POT} (Power-on reset threshold),
 - ▶ externí reset: přítomnost nízké úrovně na vstupu $\overline{Reset}$
 - ▶ přetečení ochranného časovače watchdog (musí být povoleno),
 - ▶ napájecí napětí má nižší úroveň než práh V_{BOT} (Brown-out reset threshold) (musí být povoleno),
 - ▶ reset pomocí JTAG rozhraní.
- ▶ Reset způsobí opětovný výkon programu od adresy 0x0000 (reset vektor) a slouží k nastavení mikrokontroléru do definovaného (známého) stavu, tj.:
 - ▶ všechny periférie (včetně watchdogu) jsou odpojeny,
 - ▶ všechny paralelní I/O porty jsou nastaveny jako vstupní,
 - ▶ všechna přerušení jsou vypnuta.

Generování externího resetu

- Procesory mají kromě vnitřního resetovacího obvodu, který je závislý na napájecím napětí, také externí resetovací obvod:
 - pomocí tlačítka,
 - pomocí generátorů resetovacího signálu, kdy je napájecí napětí monitorováno a při jeho poklesu je generován reset impuls. Např.: FM1233A (Fairchild Semiconductor), ADM709 (Analog Devices), ...,
 - ošetření stisku je hardwarově realizováno uvnitř mikrokontroléru, nebo uvnitř externího generátoru.



Obrázek: Tlačítko.



Obrázek: Externí generátor reset.

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů

Napájení, hodinový signál, obvod reset

Programovatelné propojky (Fuses)

Ukázka programu v JSA a v jazyce C pro ATmega16

Časovač watchdog

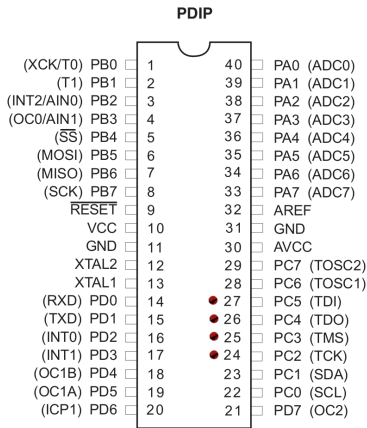
Programovatelné propojky

- ▶ Programovatelné propojky (anglicky: Fuse bits) upřesňují použití některých periférií. ATmega16 obsahuje dva byty těchto propojek:
 - ▶ povolení/zakázání ladění na čipu, použití rozhraní JTAG (programování + ladění aplikací), použití sériového programování, určení velikosti bootovací sekce ve Flash, volba start-up prodlevy, volba hodinového zdroje, ...

Fuse	Popis funkce
OCDEN	Povolení ladění přímo na čipu.
JTAGEN	Povolení JTAG rozhraní.
SPIEN	Povolení sériového programování ISP.
CKOPT	Clock options.
EESAVE	Možnost uchování EEPROM dat i po smazání čipu.
BOOTSZ1:0	Velikost bootovací sekce ve Flash (128 až 1 024 slov).
BOOTRST	Vektory přerušení do bootovací sekce.
BODLEVEL	Výběr úrovně detekce výpadku napájení (brown-out).
BODEN	Povolení detekce výpadku napájení.
SUT1:0	Výběr start-up prodlevy (0; 4,1 ms, 65 ms).
CKSEL3:0	Výběr zdroje hodinového signálu.

- ▶ Pozn.: AVR Fuse Calculator (<http://www.engbedded.com/fusecalc/>).

Umístění JTAG (Joint Test Action Group)



Obrázek: Pouzdro mikrokontroléru ATmega16 – umístění periférie JTAG.

Bootovací část paměti Flash

- ▶ Programová paměť Flash je rozdělena na dvě části:
 - ▶ aplikační část (anglicky: Application section): k uložení aplikace, která se má vykonávat,
 - ▶ bootovací část (anglicky: Boot loader section): zde může být uložen program, který umožňuje přeprogramování aplikační (i bootovací) části Flash.
- ▶ Bootloader umožňuje komunikovat s "okolním světem" pomocí dostupných protokolů (UART, SPI, I2C, ...) a také umožňuje měnit obsah aplikační části programové paměti.
- ▶ Využití: programátor (propojení PC softwaru a prog. paměti MCU) – aktualizace firmwaru elektronických zařízení.
- ▶ Velikost bootovací části je nastavitelná pomocí propojky BOOTSZ1:0 na hodnoty 128, 256, 512 nebo 1 024 paměťových slov.
- ▶ Je možnost přesunout vektory přerušení do bootovací části.

Obsah přednášky

Watchdog časovač

Základní pojmy a terminologie z mikroprocesorové techniky

Obvody s tří-stavovým výstupem

Komunikace po sběrnici

Funkce, struktura a provedení pomocných obvodů

Napájení, hodinový signál, obvod reset

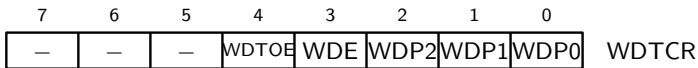
Programovatelné propojky (Fuses)

Ukázka programu v JSA a v jazyce C pro ATmega16

Časovač watchdog

Časovač watchdog v JSA

- ▶ Interní ("hlídací") časovač watchdog je nutné opakovaně nulovat – typicky v nekonečné smyčce. Při přetečení způsobí reset systému.
 - ▶ Nulování pomocí instrukce WDR – Watchdog Reset.
- ▶ Přetečení může být způsobeno pomalou obsluhou přerušení/podprogramů, nebo zacyklením vně nekonečné smyčky.
- ▶ Ovládání prostřednictvím WDTCR (Watchdog Timer Control Register):
 - ▶ WDE – Watchdog Enable,
 - ▶ WDP – Watchdog Timer Prescaler.



Obrázek: Struktura kontrolního registru WDTCR (Watchdog Timer Control Register).

Časovač watchdog v JSA

```
1 ; inicializace aplikace
2 .include <m16def_mod.inc> ; vložit upravený popisný soubor pro ATmega16
3
4 ; vektory přerušení
5 .cseg ; paměťový segment Flash
6 .org 0x0000 ; ulož od adresy 0x0000
7 rjmp reset ; skoč na návěští reset
8 .org INT0addr ; vektor přerušení INT0
9 rjmp tlacitko ; skoč na návěští tlacitko
10
11 ; hlavní aplikace
12 .org 0x002a ; ulož od adresy 0x002a
13 reset:
14 ... ; definice zásobníku, tj. naplnění registrů SPH:L
15 ... ; naplnění použitých kontrolních registrů
16 ldi r16, (1<<WDE) | (1<<WDP2) | (1<<WDP1) | (1<<WDPO)
17 out WDTCR, r16 ; povolení watchdogu s předděličkou @ 2,1 s
18 sei ; globální povolení všech přerušení
19
20 ; nekonečná smyčka
21 loop:
22 wdr ; nulování watchdogu
23 rjmp loop ; skok na návěští/místo v programu loop
24
25 ; obsluhy přerušení
26 tlacitko: ; obsluha externího přerušení INT0
27 ... ; kód obsluhy přerušení
28 rjmp tlacitko ; CHYBNÉ zacyklení programu
29 reti ; ukončení obsluhy přerušení
```

Časovač watchdog v jazyce C

- ▶ Vložení instrukce AVR do zdrojového kódu v jazyce C bez nutnosti předávání parametrů:
 - ▶ funkce `asm()` ;
 - ▶ parametrem je řetězec obsahující posloupnost instrukcí, které se mají přímo vložit do překompilované aplikace,
 - ▶ jednotlivé instrukce lze oddělovat `\n`:
Např.: `asm( "wdr\ndec r16" ) ;`

```
1  /* inicializace aplikace */
2  #include "avr/io.h"           // vložení popisného/hlavičkového souboru AVR
3  #include <avr/interrupt.h>    // hlavičkový soubor pro přerušení
4  ...                           // případné další knihovny
5
6  /* obsluha přerušení */
7  ISR( INTO_vect ){            // obsluha externího přerušení INTO
8  ...                           // kód obsluhy přerušení
9  while( 1 ) ;                  // CHYBNÉ zacyklení
10 }
11
12 /* hlavní funkce aplikace */
13 int main( void )
14 {
15     ...                       // kód hlavní funkce
16     // povolení watchdogu s předděličkou @ 2,1 s
17     WDTCR |= (1<<WDE) | (1<<WDP2) | (1<<WDP1) | (1<<WDPO) ;
18     sei() ;                    // globální povolení všech přerušení
19     while( 1 ){                // nekonečná smyčka
20         asm( "wdr" ) ;         // nulování watchdogu
21     }
22
23     /* formální ukončení funkce main */
24     return( 1 ) ;              // návratová hodnota funkce main = 1
25 }
```

Zdroje informací



VÁŇA, V.

Mikrokontroléry ATMEL AVR; popis procesorů a instrukční soubor.

Ben – technická literatura, Praha, 2003, ISBN 80-7300-083-0.



Atmel Corporation.

Atmel Microcontrollers, (září 2013).

<http://www.atmel.com/products/microcontrollers/default.aspx>



Atmel Corporation.

Application Note AVR4013: picoPower Basics, (říjen 2013).

www.atmel.com/Images/doc8349.pdf



NonGNU.org.

Power Reduction Management, (říjen 2013).

http://www.nongnu.org/avr-libc/user-manual/group__avr__power.html