

## Implementace Univerzálního sériového rozhraní do jednočipového mikropočítače

### Abstrakt

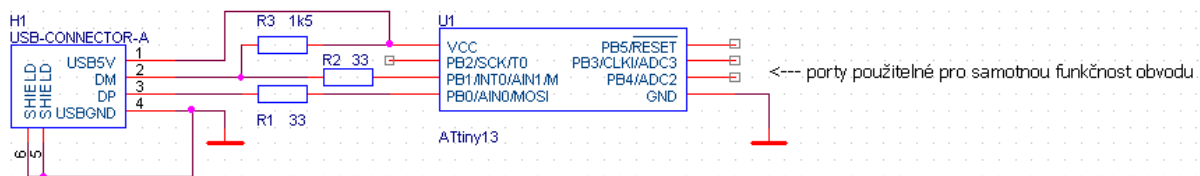
Univerzální sériové rozhraní (USB) [1] je moderní prostředek vzájemného propojení elektronických zařízení. Nejčastěji se používá pro připojení periférií k počítači. Mezi výhody této technologie patří její všeobecné rozšíření na různých platformách a zdarma dostupná specifikace. Při návrhu přenositelného zařízení připojitelného k počítači je USB mnohdy jedinou možnou volbou. Konstruktor má k dispozici řadu integrovaných obvodů, které se starají o implementaci celé USB specifikace a na svém výstupu poskytují čistá data. Zpracování těchto dat se poté zpravidla děje v jiném programovatelném obvodu zajišťujícím samotnou funkčnost zařízení. Cílem práce je vytvořit knihovnu funkcí pro jednočipový mikropočítač firmy Atmel rodiny AVR (Advanced Virtual RISC) [2], která bude implementovat USB a umožní tak sloučení řadiče USB a samotné funkčnosti do jednoho programovatelného obvodu.

### Motivace

Valná většina současných USB periférií řeší implementaci specifikace hardwarově. Často se používají USB převodníky (například FTDI USB→serial [3]), které převádějí datový tok USB na jinou sběrnici (RS-232, RS-485, IEEE 1284) a neumožňují využít všech výhod USB sběrnice. Použití takových obvodů s sebou nese nutnost používat ovladače zařízení vydané výrobcem obvodu bez možnosti vlastní úpravy a zlepšení. Existují též USB řadiče, které jsou součástí programovatelných jednočipových mikrokontrolérů. Řízení řadiče, jakožto i přístup k přijímaným a odesílaným datům se zajišťuje prováděním instrukcí mikropočítače. Pohodlné používání a efektivita obvodu je vyvážena vysokou cenou několikanásobně převyšující cenu ostatních programovatelných obvodů bez integrovaného USB řadiče. Aby bylo možné přistupovat k USB sběrnici i pomocí programovatelných obvodů, které nejsou pro takovou komunikaci nijak hardwarově vybaveny, musíme vytvořit software využívající pouze standardní vstupně-výstupní porty a nahrazující chybějící hardware efektivním používáním instrukční sady. V současné době je k dispozici kód [4], který implementuje USB rozhraní a funkčnost obvodu (přijímač dálkového ovládání - DO) plně softwarově, ale není modulární. Nelze tudíž oddělit kód USB řadiče od kódu zajišťujícího samotnou funkci přijímače DO. Jakákoliv úprava funkčnosti má za následek pracné přepisování téměř celého programu. Tato práce se snaží vylepšit stávající řešení, a proto dodržuje při návrhu následující kritéria: modularita funkčních celků, zapouzdřenost kódu, přenositelnost řešení uvnitř rodiny AVR, použití levného a dostupného hardware a minimalizace počtu externích elektronických součástek.





### Časování

USB přenos může fungovat ve třech rychlostních režimech - 1,5Mb/s, 12Mb/s a 480Mb/s. Mikrokontroléry AVR mohou dosáhnout maximální rychlosti 20MHz, což znamená, že implementovat lze pouze režim 1,5Mb/s. Rychlejší režimy lze při schopnostech dnešních mikrokontrolérů řešit pouze hardwarově. Časování sběrnice je odvozeno od systémových hodin procesoru a smí se lišit od nominální hodnoty 1,5MHz o 1,5% (15 000 ppm). Mikrokontroléry AVR mohou jako zdroj hodinového signálu používat krystalový oscilátor, generátor nebo vnitřní RC oscilátor. Některé typy AVR neumožňují připojit krystal, protože jejich pouzdro má příliš malý počet vývodů. Na následujícím obrázku je uvedeno schéma miniaturního USB zařízení, které pro svůj běh potřebuje minimum externích součástek (tři rezistory). Obvod ATtiny13 je časován vnitřním programově laditelným RC oscilátorem. Pro komunikaci používá univerzální vstupně-výstupní port B.



Hodnota frekvence systémových hodin musí být celočíselným násobkem frekvence sběrnice 1,5MHz. Mikrokontroléry AVR provádějí většinu instrukcí během jednoho taktu hodin. Při čtení a zápisu na linku (pomocí portu B) se nepoužívá časovač, správné časové poměry se zajišťují vhodným návrhem programu, který spotřebovává pevně daný počet taktů při příjmu/odeslání jednoho bitu dat USB linky. Jakákoliv přerušení, které by nastalo během vykonávání kódu přijímače/vysílače, by mělo za následek zpoždění programu a neúspěch při přenosu dat. Procesor tedy po dobu přenosu vykonává pouze rutiny USB řadiče, ostatní operace (funkce obvodu) musejí být obslouženy v době klidu na lince.

V tabulce jsou shrnuty některé výhodné konfigurace parametrů:

Obvod	ATtiny13	ATtiny2313	ATtiny8	ATtiny13
Frekvence [MHz]	12	18	15	19,5
Zdroj hodin	RC osc.	krystal	krystal	generátor
Nepřesnost hodin [ppm]	7 000	50-100	50-100	50
Počet instrukcí/bit	8	12	10	13
Celková/použitá paměť flash [B]	1024/900	2048/1000	8096/1200	1024/900
Celková/použitá paměť SRAM [B]	64/48	128/32	1024/32	64/32
Vytížení při přenosu 10kB/s	95%	85%	90%	90%
Pouzdro	SOIC 8	SOIC 20	TQFP32	DIL 8
				

### Dekompozice

Softwarovou implementaci USB řadiče lze rozdělit na několik dílčích problémů a ty potom řešit samostatně. Řadič je ve své podstatě série proudových filtrů, které provádějí operace s bitovým datovým tokem za účelem jeho transformace do jiného formátu. Řadič také vyřizuje režijní požadavky sběrnice. Pro přijetí/odeslání dat po sběrnici je nutné zvládnout tyto operace:

- vytvoření (odstranění) kanálového kódování NRZI a bitové výplně (bit stuffingu)
- výpočet a vyhodnocení cyklické redundantní kontroly (CRC)
- serializace a deserializace dat tvořících paket uložený v RAM
- vyřízení inicializačních požadavků hostitele (enumerace)

Ze zvolené frekvence systémových hodin vyplývá i počet taktů, které jsou rutině k dispozici pro příjem/odeslání jednoho bitu dat. Knihovna USB řadiče obsahuje čtyři verze všech rutin pro hodinové taktů 12, 15, 18 a 19,5 MHz. Jednotlivé verze se od sebe liší silou optimalizací a různou spotřebou programové paměti (rychlejší kód spotřebuje více paměti). USB specifikace definuje povinné a volitelné požadavky, které na které musí zařízení odpovídat. Při nedostatku programové paměti se volitelné požadavky neimplementují, avšak na správnou funkci zařízení to nemá vliv. Pro tvorbu kódu je kvůli potřebě přesně a optimálně řídit běh procesoru použit Assembler. Uživatelům jazyka C je k dispozici knihovna opatřená formalismem pro import v C.

### Součinnost řadiče a funkčnosti

Výpočetní čas procesoru musí být rozdělen mezi čtyři úlohy - příjem paketu, odeslání paketu, vyřizování požadavků (enumerace) a zajištění funkčnosti obvodu (převody, jiná komunikace, výpočty, ...). Z důvodu modularity se nesmí zdrojové kódy jednotlivých úloh prolínat. Hlavní smyčka programu patří funkci obvodu a může být kdykoliv hardwarově přerušena příchodem USB paketu. Rutina příjmu paketu má za úkol načíst data z linky a uložit je do SRAM (statická operační paměť). Hlavní smyčka poté musí sama zavolat funkce pro rozbor obsahu paketu, vyřízení požadavku a odeslání odpovědi. Každá z rutin pracujících se sběrnici plně využívá čas procesoru a má za svého běhu zakázána přerušení. Uvedené řízení souběhu programů lze označit jako kooperativní (nepreemptivní) multitasking. Výhodou je spolehlivost rutin přenosu (nemohou selhat z důvodu nedostatku času) a možnost volat funkce zpracování paketu v okamžik vhodný pro hlavní smyčku. Nevýhodou je neodolnost proti selhání hlavní smyčky, která může opomenout zavolání zpracujících rutin a tím způsobit vypršení časového limitu pro zaslání odpovědního paketu USB hostiteli.

### Závěr

Výsledkem práce je softwarové řešení USB řadiče pro mikrokontroléry AVR včetně funkcí pro zpracování paketu a vyřízení enumerace. K dispozici jsou verze pro různé frekvence hodin mikrokontroléru. Modulární kód lze snadno zabudovat do jiných programových celků bez narušení jejich logické struktury. Rychlost přenosu pomocí softwarového řadiče se pohybuje okolo 10kB/s, při nižším vytížení sběrnice je volný výpočetní čas plně k dispozici uživatelskému programu. Řadič je odolný proti chybám díky kontrole CRC. Řešení se vyznačuje především širokou škálou volitelných parametrů (frekvence, zdroje hodin, typy obvodů, pouzdra), dostupností hardwaru a značnými finančními úsporami (až 10x) oproti současným hardwarovým i softwarovým alternativám.

### Zdroje

- [1] USB Implementers Forum. Universal Serial Bus Specification Revision 1.1 [online], c1998. 311 s. [cit. 2007-04-10]. URL: <<http://www.usb.org/developers>>.
- [2] Atmel Corporation. AVR 8-Bit RISC Introduction [online], c2007. [cit. 2007-05-15]. URL: <<http://www.atmel.com/products/avr/overview.asp>>.
- [3] Future Technology Devices International Limited. FTDI Products [online], c2007. [cit. 2007-05-16]. URL: <<http://www.ftdichip.com/FTProducts.htm>>.
- [4] ČEŠKO, Igor. Dálkové ovládání přes USB [online], c2000-2007. [cit. 2007-05-21]. URL: <[http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20\(AVR\).htm](http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20(AVR).htm)>.