

Česká Zemědělská Univerzita v Praze

Technická fakulta



Metody vytváření a ladění aplikací s rozhraním USB

Bakalářská práce

Vedoucí práce: prof. Ing. Zdeněk Bohuslávek, CSc.

Vypracoval: Rudolf VACEK

Obor: Informační a řídicí technika v agropotravinářském komplexu

PRAHA 2012

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra elektrotechniky a automatizace

Technická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Vacek Rudolf

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Methody vytváření a ladění aplikací s rozhraním USB

Anglický název

Methods of developing and debugging applications with USB

Cíle práce

Popsat metody implementace rozhraní USB do aplikací s mikrokontrolery – jednočipovými mikropočítači. Navrhnout metodu implementace do procesorů AVR.

Metodika

1. Zpracování literárních podkladů a konzultace u firem k bodům osnovy 1) a 2).
2. Zpracování přehledu podle bodu osnovy 3) a pokynů z konzultací s vedoucím práce.
3. Koncepční návrh zaměřit na speciální čtecí zařízení.

Osnova práce

1. Popis známých metod řešení komunikace pomocí USB mezi mikrokontrolerem a počítačem.
2. Charakteristika rozhraní USB
3. Přehled dostupných řadičů pro USB v. 2 a jejich stručný popis.
4. Koncepční návrh implementace v mikroprocesorech Atmel AVR.
5. Shrnutí a závěr.

Rozsah textové části

30 stran textu včetně obrázků, grafů a tabulek

Klíčová slova

USB implementace, mikrokontroler, řadiče USB

Doporučené zdroje informací

1. Štohanzel, M., Štrauss, P.: Softwarová realizace USB, dostupné na http://pandatron.cz/?1532&softwarova_realizace_usb
2. Češko, I: [http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20\(AVR\)_eng.htm](http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20(AVR)_eng.htm)
3. USB Implementers Forum. Universal Serial Bus Specification Revision 1.1 [online], c1998. 311 s. [cit. 2007-04-10]. URL: <<http://www.usb.org/developers>>.
4. CEŠKO, Igor. Dálkové ovládání přes USB [online], c2000-2007. <[http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20\(AVR\).htm](http://www.cesko.host.sk/IgorPlugUSB/IgorPlug-USB%20(AVR).htm)>.
5. Coufal, T.: Softwarová implementace USB pro mikrokontroléry AVR <http://hw.cz/teorie-praxe/art1953-softwarova-implementace-usb-pro-mikrokontrolery-avr.html>

Vedoucí práce

Bohuslávek Zdeněk, prof. Ing., CSc.

Termín zadání

listopad 2010

Termín odevzdání

duben 2012


prof. Ing. Jaromír Volf, DrSc.
Vedoucí katedry




prof. Ing. Vladimír Jurča, CSc.
Děkan fakulty

V Praze dne 7.2.2011

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Metody vytváření a ladění aplikací s rozhraním USB“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu citované literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 16. 4. 2012

Poděkování

Na tomto místě chci poděkovat především vedoucímu mé bakalářské práce prof. Ing. Zdeňku Bohuslávskovi, CSc. za ochotu, trpělivost a cenné rady při psaní této práce.

Také bych chtěl poděkovat svému dědovi Ing. Janu Vackovi za gramatické a typografické korekce této práce.

Abstrakt

Práce je soustředěna na mikro kontroléry AVR a sběrnici USB. Propojení těchto dvou prvků nemusí být vždy jednoznačné a existuje poměrně široké spektrum možností. Cílem této práce je jejich popis a zhodnocení. První kapitola popisuje nástroje a periferie mikro kontrolérů AVR od korporace Atmel. Druhá kapitola se věnuje sběrnici USB, problematika je rozsáhlá, přesto bylo téma shrnuto do popisu základních vlastnosti rozhraní USB. Třetí kapitola obecně popisuje řešení problematiky využití mikro kontrolérů využívajících USB a uvádí konkrétní řešení pro softwarovou implementaci. Čtvrtá kapitola představuje konkrétní obvody pro hardwarové řešení. Pátá kapitola ukazuje konkrétní řešení z praxe.

Práce je vytvořena pomocí studia literatury jako literární rešerše.

Oblast mikro kontrolérů a různých rozhraní je velice rychle rozvíjející se oblast, proto již při psaní práce přestávají některé informace být aktuální, přesto je práce užitečným souhrnem v odvětví.

Klíčová slova

AVR, Atmel, USB, IgorPlug, AVR309, V-USB, LUFA, FTDI, Silicon Labs, MICROCHIP

Abstract

The thesis is focused on micro-controllers and AVR USB bus. Linking these two elements is not always clear and there are is a quite wide range of options. The aim of this work is to describe and evaluate these options. The first chapter describes tools and peripherals of AVR micro controllers from Atmel Corporation. The second chapter is devoted to the USB bus, the issue is extensive, but the topic was summarized to describe the basic properties of the USB interface. The third chapter describes the general solution of the use of micro controllers using USB and provides specific solutions for software implementation. The fourth chapter presents specific advanced hardware solutions. The fifth chapter shows a real practical solution.

The work is created through the study of literature as a literature search.

The area of micro controllers and various interfaces is a very fast growing area, so already at writing this work some information cease to be current, but it is a useful summary for work in the area.

Keywords

AVR, Atmel, USB, IgorPlug, AVR309, V-USB, LUFA, FTDI, Silicon Labs, MICROCHIP

Obsah

1	Úvod.....	1
1.1	Korporace Atmel	1
1.2	Architektúra AVR.....	1
1.3	Integrované periferie AVR	2
1.4	Rodiny mikro kontrolérů architektury AVR	4
2	Charakteristika rozhraní USB	5
2.1	Souhrnné vlastnosti.....	5
2.2	Topologie.....	6
2.3	Elektrická vrstva	7
2.4	Mechanická část	8
2.5	Data Flow Model	9
2.6	Protokol	11
2.7	Cyklická redundantní kontrola	13
2.8	USB On-The-Go (USB OTG)	14
2.9	Verze 2.0 a 3.0.....	15
3	Známé metody řešení komunikace pomocí USB mezi mikro kontrolérem a počítačem. 16	
3.1	Integrovaný řadič USB na chipu mikro kontrolérů	16
3.2	Samostatný integrovaný převodník USB a nativního rozhraní mikro kontroléru	17
3.3	Virtuální softwarová implementace USB pomocí mikro kontroléru.....	18
3.3.1	IgorPlug/AVR309 (11).....	18
3.3.2	V-USB (AVR-USB) (13)	21
4	Přehled dostupných řadičů pro USB v. 2.0 a jejich stručný popis	26
4.1	Řadiče firmy FTDI	26
4.2	Řadiče firmy Silicon Labs	29
4.3	Řadiče firmy MICROCHIP	30
5	Koncepční návrh implementace v mikroprocesorech Atmel AVR.....	31
5.1	Váhový počítač	31
5.2	Čtecí zařízení	32
5.3	Manažerský program	34
6	Závěr.....	37
7	Citovaná literatura	38
8	Seznam obrázků	40
9	Seznam tabulek	41

1 Úvod

1.1 Korporace Atmel

Korporace Atmel zaměřená na výrobu polovodičů, byla založena roku 1984 v USA.

Dnes vlastní továrny i mimo USA, v Norsku, Velké Británii, Francii a Německu.

Soustředí se na vývoj a výrobu RISC (redukovaná instrukční sada - Reduced Instruction Set Computer) mikro kontrolérů dále jen MCU, větvě 8051 (Intel r. 1980), využívá licencovaného jádra ARM a vlastní architekturu AVR. Dále pak vyrábí paměti, bezdrátová řešení, dotyková řešení a další specifická zařízení.

Čistý peněžní tok (net cash flow) 517 mil. USD k prosinci 2011. (1)



Obrázek 1 - Logo Korporace Atmel (2)

1.2 Architektura AVR

Jádro AVR je výsledkem práce dvou studentů, Alf-Egila Boga a Vegarda Wollana, v Norském technickém institutu (Norges Tekniske Høgskole – NTH)

Jde o upravenou 8-bitovou Harvardskou koncepci (fyzické oddělení dat od programu) mikropočítače s redukovanou instrukční sadou RISC. Šlo o jeden z prvních MCU, který měl program zapsán na integrované paměti typu Flash.

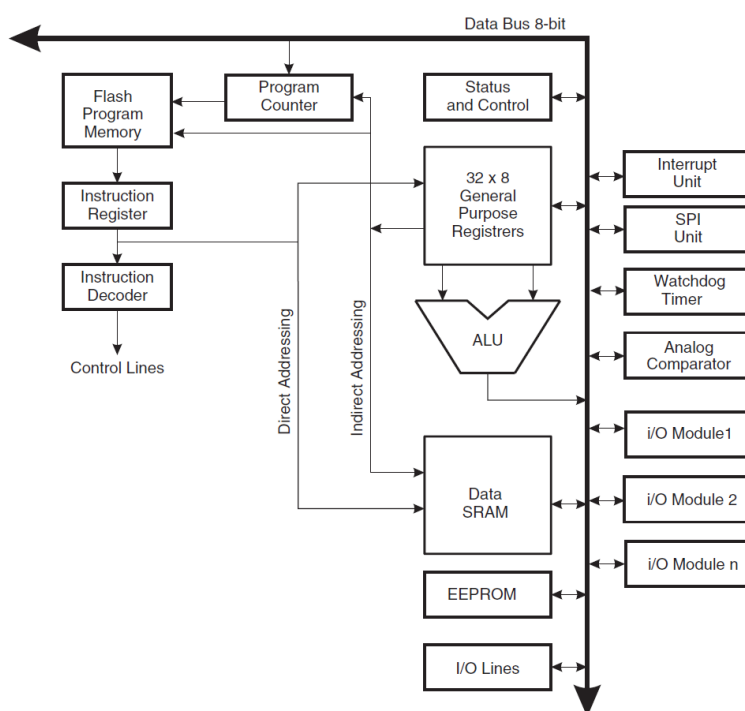
Alf-Egil Bogen a Vegard Wollan se vydali do USA, aby získali finanční a technologickou podporu pro svůj projekt.

Podporu získali od korporace Atmel a v roce 1996 začala výroba prvních MCU rodiny AVR. (3)

1.3 Integrované periferie AVR

Všechny MCU s jádrem AVR mají na základním chipu integrované běžně užívané periferie. Čítače, PWM, A/D převodníky, řídicí obvody sběrnic I2C – USART – SPI – USB, Watchdog timer a analogový komparátor, představující základní prvky komunikace a práce s diskretními prvky a obvody.

Následuje krátký popis periferií (ATmega8(L)):



Obrázek 2 - Blokový diagram Arch. AVR (2)

8-bitový časovač/čítač s vlastní děličkou

(8-bit Timer/Counters with Separate Prescaler)

Základní funkce: jednokanálový čítač, frekvenční generátor, čítač vnějších událostí a 10-bitový dělič vnitřních hodin.

16-bitový časovač/čítač s vlastní děličkou

(16-bit Timer/Counter with Separate Prescaler)

Obvod čítače reálného času s vlastním oscilátorem

(Real Time Counter with Separate Oscillator)

Tři kanály PWM

(Three PWM Channels)

Pulsně šířková modulace (Pulse Width Modulation) slouží jako D/A převodník, pomocí nastavení příslušného registru lze dosáhnout požadované frekvence v závislosti na zvoleném oscilačním kmitočtu hlavních hodin, rozlišení PWM/čítače a hodnotě děliče.

A/D převodník

(6 to 8-channel ADC, 10-bit Accuracy)

Počet kanálů je odvozen od zvoleného pouzdra, vstupy zpracovává vstupní multiplexor.

Základní vlastnosti A/D převodníku: rozlišení 10-bitů, 0,5 LSB vnitřní nelinearita

(least significant bit - nejméně významný bit), ± 2 LSB absolutní přesnost,

čas převodu 13 až 260 μ s, až 15 kSPS (Kilo Samples Per Second) při maximálním

rozlišení, možnost záznamu výsledku zleva, rozsah měřitelného napětí 0 až Vcc,

volitelné referenční napětí 2,56 V, funkce volného běhu nebo měření po jednom vzorku,

přerušování po skončení převodu, funkce probuzení.

Dvou vodičové rozhraní

(Two-wire Serial Interface)

Někdy označováno jako I²C (I-squared-C). Rozhraní je typu Master/Slave, Master

zahajuje a ukončuje relaci, generuje hodinový signál, Slave je adresován masterem.

Výhodou je potřeba pouze dvou vodičů ke komunikaci.

Synchronní / asynchronní sériové rozhraní USART (USRT/UART)

(Programmable Serial USART - Universal Synchronous / Asynchronous Receiver and Transmitter)

Umožňuje synchronní/ asynchronní komunikaci, nejčastěji je spojováno s rozhraním RS232 a RS485.

Sériové periferní rozhraní

(Master/Slave SPI - Serial Peripheral Interface)

Podobné rozdělení jako u I²C, Master/Slave. Také se používá pro nahrání programu MCU do paměti Flash.

Watchdog timer - „hlídací pes“

Hlídá běh MCU v případě potřeby resetuje obvod. Zajišťuje také přechod do různých režimů napájení.

Analogový komparátor

(Analog Comparator)

Porovnává napětí dvou vstupů a v případě rozdílu podá odpovídající výstup.

Interní RC oscilátor

Nastanou-li případy kdy nepotřebujeme rychlý vnější oscilátor, ale bohatě nám dostačuje vnitřní, nepříliš přesný ovšem kalibrovatelný RC oscilátor.

Popsané periferie odpovídají obvodu ATmega8(L), ten byl vybrán, protože obsahuje všechny základní prvky MCU AVR. (2)

1.4 Rodiny mikro kontrolérů architektury AVR

megaAVR

Představuje základní řadu 8-bitové architektury AVR. Pozdější verze byly doplněny o řadič USB a RTC.

Obecné parametry: 4 – 256 KB programovatelné Flash paměti,
28 až 100 vývodů, frekvence oscilátoru až 20 MHz, 1,0 MIPS/MHz.
(Million Instruction Per Second - Milion Instrukcí Za Sekundu)

tinyAVR

Zmenšená verze rodiny megaAVR, použití najde tam, kde není třeba většího počtu vývodů a vystačí menší paměť.

Obecné parametry: 0,5 - 8KB programovatelné Flash paměti,
6 až 32 vývodů, frekvence oscilátoru až 20 MHz, 1,0 MIPS/MHz.

AVR XMEGA

Vývoj megaAVR přinesl mnohá vylepšení, jako například Full-Speed USB řadič s vylepšeným přenosem, který méně zatěžuje procesor. Byla snížena spotřeba energie. Všechny periferie mohou používat DMA (Direct Memory Access - přímý přístup do paměti).

Obecné parametry: 16 – 384 KB programovatelné Flash paměti, 44 až 100 vývodů, frekvence oscilátoru až 32 MHz, 1,0 MIPS/MHz

32-bit AVR UC3

Původním záměrem bylo vytvořit konkurenci nebo alternativu k MCU s jádrem ARM, ale později korporace Atmel koupila licenci a sama začala obvody s jádrem ARM vyrábět a prodávat.

Hlavním rysem je 32-bitová architektura, numerický koprocessor (FPU – floating point unit), Ethernet s MAC adresou, USB včetně podpory USB On-the-Go oproti koncepci Master/Slave, při koncepci On-the-Go může zařízení zastávat obě role.

Značnou nevýhodou rodiny 32-bit AVR je cena, někdy může být až pětinasobkem ceny AVRmega nebo AVR XMEGA, záleží na konkrétním distributorovi.

Dále pak existují doplňkové rodiny pro automobilový průmysl, řízení baterií a dotykové řešení. (4)

2 Charakteristika rozhraní USB

USB neboli Univerzální Sériová Sběrnice (Universal Serial Bus) je široce rozšířené komunikační rozhraní především u PC (osobní počítač - Personal Computer) pro jeho rozšířenost, flexibilitu a schopnost napájení připojeného zařízení.

Všechny MCU AVR používají, nebo mohou být upraveny, aby používaly verzi USB 1.1 tzv. Full-Speed resp. verzi 1.0, která se od 1.1 příliš neliší, proto popisu této, poněkud starší verze bude věnováno nejvíce prostoru. Novější verze se liší jen málo, nejvíce asi přenosovou rychlostí.

Verze USB a používané označení	Max. přenosová rychlost	Počet vodičů (kabeláž je stíněná)
1.0 ~ Low-Speed	1,5 Mbit/s	4
1.1 ~ Full-Speed	12 Mbit/s	4
2.0 ~ Hi-Speed	480 Mbit/s	4
3.0 ~ Super-Speed	5 Gb/s	9

Tabulka 1 - Verze USB

Zpětná kompatibilita mezi verzemi USB je zaručena.

Verze 1.0 nebyla natolik rozšířena, aby bylo významné jí popisovat.

Existuje bezdrátová (wireless) verze USB, ale není příliš rozšířena, proto byla pro účely této práce vynechána.

2.1 Souhrnné vlastnosti

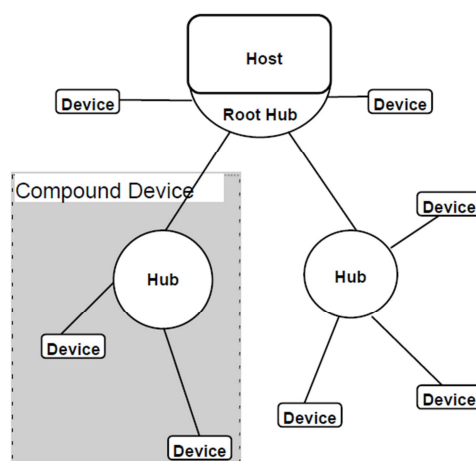
- Každé zařízení USB má svůj jedinečný identifikační kód tzn. Device ID a zároveň jedinečný identifikační kód výrobce Vendor ID, tudíž zařízení dokáže systém sám identifikovat, nastavit a začít používat pomocí příslušného ovladače.
- Nezaměnitelný model kabeláže a konektorů.
- Detaily el. vrstvy jsou pro uživatele irelevantní.
- Možnost dynamického připojení a odpojení zařízení.

- Široký rozsah přenosových rychlostí.
- Maximální délka propojovacího kabelu 5 metrů.
- Podpora synchronního a asynchronního přenosu.
- Podpora až 127 zařízení na jednom kořenovém rozbočovači.
- Žádné zařízení nemůže začít vysílat bez vyzvání.
- Podpora velkého rozsahu velikosti paketů (přenášených bloků).
- Vyhodnocení a oprava chyb přenosu.
- Řízení toku dat a obsluha bufferů (vyrovnávací paměti) je řešeno v protokolu.
- Protokol je jednoduchý na implementaci a integraci.
- Kompatibilní s architekturou a Plug-and-Play „zasuň a hraj“.
- Optimalizováno pro vývoj levných periférií.

(5)

2.2 Topologie

Konkrétní topologie představuje tzv. hvězdicovou topologií tj. jeden hlavní bod (Host), nejčastěji PC, se kterým jsou spojeny ostatní zařízení. Rozbočovač (Hub) je se zařízením (Device) spojen kabelem. Rozbočovač se chová jako opakovač paketů. Všechna zařízení mohou být spojena pouze s jedním kořenovým rozbočovačem (RootHub) tj. v systému může být pouze jeden kořenový rozbočovač. Na jednom kořenovém rozbočovači může být připojeno nejvýše 127 zařízení.



Obrázek 3 - Topologie USB (5)

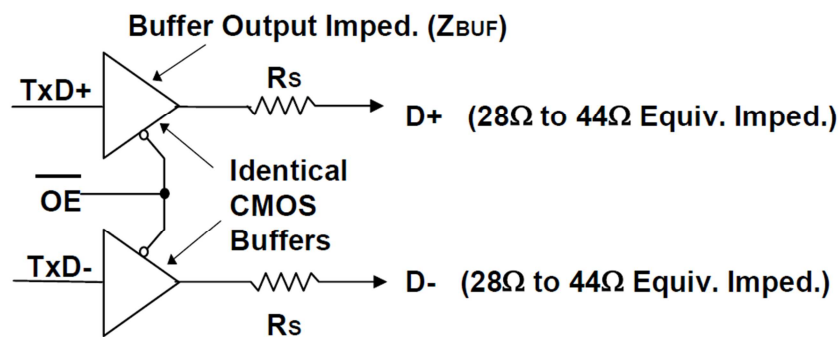
Hostující počítač (Host) je implementací kombinace hardwaru a firmwaru.

(5) (6)

2.3 Elektrická vrstva

USB přenáší signál pomocí rozdílů el. napětí na datových vodičích tzv. diferenciální vedení. Signál je na napěťových úrovních 0 V a 3,3 V (vodiče D+ a D-). (6)

Diferenciální vedení má impedanci 90Ω , proto se doporučuje používat pouze certifikované kabely. Buzení je zajištěno na obou stranách pomocí CMOS hradel a sériově zapojených rezistorů hodnoty 28 až 44Ω , které vytvářejí impedanční přizpůsobení. Když hostitel nebo zařízení přecházejí do režimu vysílání, mohou buďcí hradla být odpojena. (7)



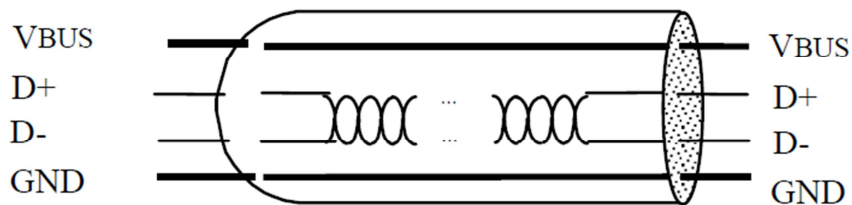
Obrázek 4 - Budící hradla (5)

Aby bylo možné rozeznat Low-speed zařízení od Full-speed je zajištěno pomocí tzv. pull-up rezistorů $1,5 \text{ k}\Omega$, rozdílné el. napětí na datových vodičích D+ a D-. Konkrétně pro Low-speed je $1,5 \text{ k}\Omega$ rezistor zapojen mezi +3,3 V a datový pin D-, pro Full-speed je rezistor zapojen mezi +3,3 V a datový pin D+. (6)

Velkou předností USB je možnost napájení zařízení přímo z rozbočovače. Při připojení může zařízení odebírat až 100 mA el. proudu, pokud je nutné pro jeho provoz více musí požádat o řídicí obvod, zařízení může mít maximální odběr až 500 mA. Ochrana je řešena jednoduše pomocí vratných pojistek tzv. polyswitch nebo komplexně pomocí spínaných tranzistorů typu MOS-FET. (6)

2.4 Mechanická část

Specifikace USB má také jasně definované rozměry a použité materiály prvků, hlavní rysy jsou shrnuty v této kapitole.

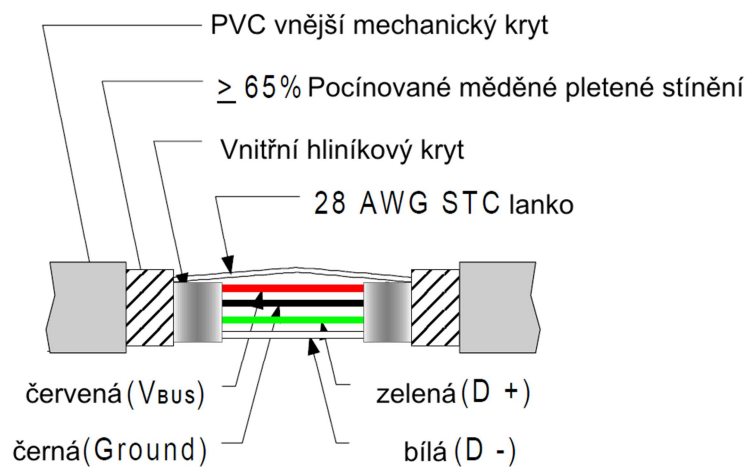


Obrázek 5 - Struktúra kabelu USB (5)

Jak je patrné z obrázku č. 5 datové vodiče jsou křížené (kroucená dvojlinka), pro menší rušení, napájecí vodiče jsou vedeny přímo. Kabel není křížený, ale přímý tj. pořadí napájecích vodičů je na každém konci stejné.

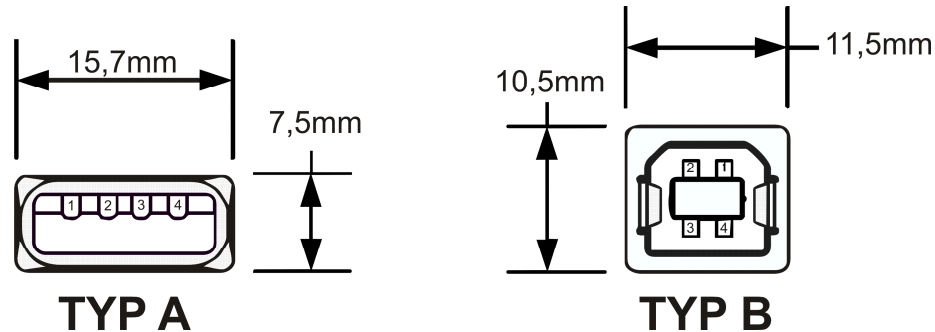
Číslo kontaktu	Název vodiče	Typická barva vodiče
1	VBUS	Červená
2	D-	Bílá
2	D+	Zelená
3	GND	Černá
Kryt	Stínění	Stínící lanko/drát

Tabulka 2 - Zapojení kabelu USB (5)



Obrázek 6 - Použité materiály (5)

Verze USB 1.1 znala pouze dva základní konektory, TYP A a TYP B, dnes již existuje celá řada miniaturních konektorů pro mobilní telefony a MP3 přehrávače a podobná přenosná zařízení, ale TYP A a TYP B jsou stále nejrozšířenější. Dostupnost samičích protikusů pro tištěné spoje je dobrá. (5)



Obrázek 7 - Konektory USB (5)

2.5 Data Flow Model

Kapitola se věnuje popisu principů pohybu a vyměňování paketu přes USB s cílem na

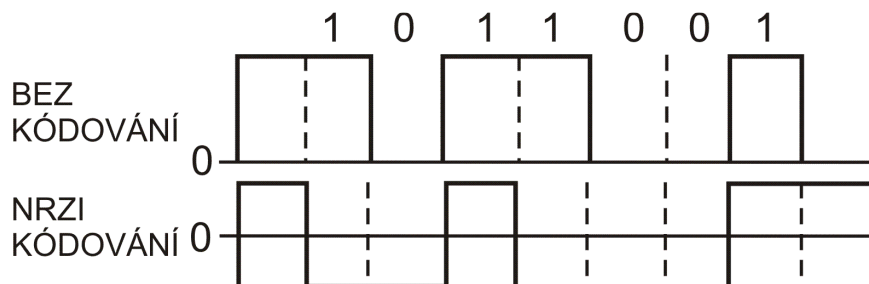
- přenos užitečných dat (pipeline neboli datový kanál)
- zahájení komunikace (pipeline → endpoint 0)
- řízení stavu přenosu
- korigovat chyby přenosu (handshaking)

Na komunikaci nahlížíme jako na rouru (pipeline) neboli datový kanál, která vede od Hostitele k zařízení.

Kódování NRZI

Způsob kódování (Non Return to Zero Inverted - záznam bez návratu k nule) kdy logická 1 znamená změna úrovně signálu, logická 0 beze změny stavu.

Problém nedělají dlouhé posloupnosti logických 0 ovšem dlouhé posloupnosti logických 1 už ano, řešením je vkládání bitů – bit-stuffing.



Obrázek 8 - NRZI kódování

Vkládání bitů – bit-stuffing

Obsahuje-li původní datový tok šest po sobě jdoucích jedniček, přidá vysílač automaticky jednu nulu, aby se tím vynutila změna úrovně.

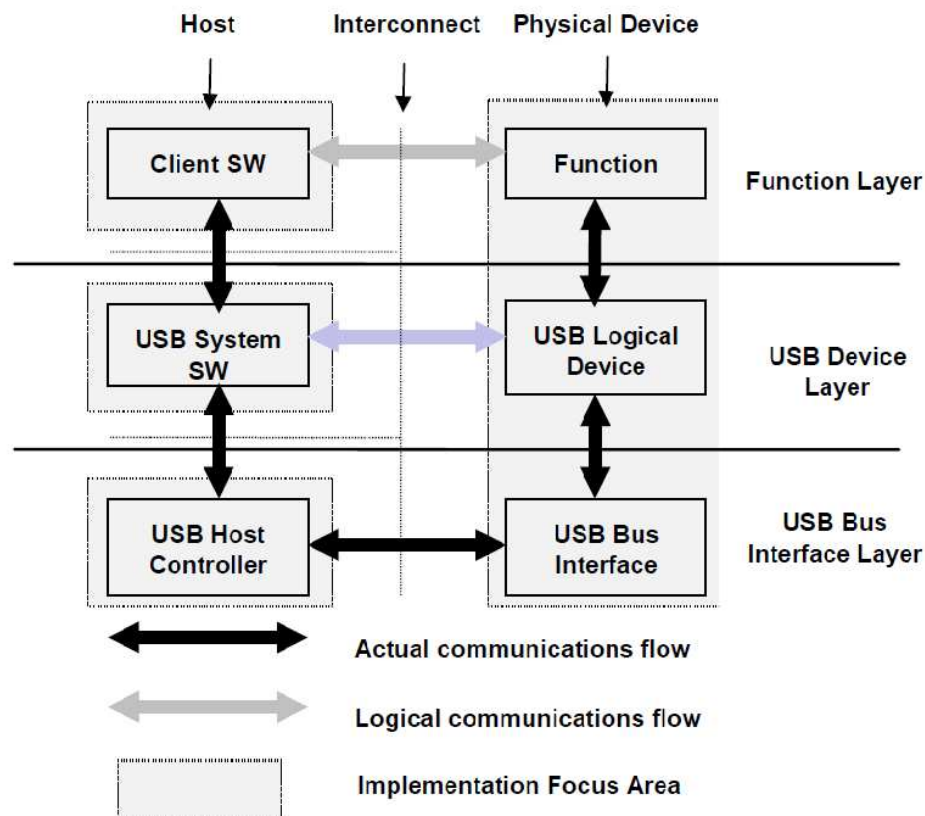
Přijímač tuto nulu z datového toku opět odstraní.

Úrovně komunikace

Host se skládá ze tří částí: klientský software, programové vybavení a Hostitelský kontrolér.

Fyzické zařízení se skládá ze tří částí: funkční, logické zařízení a sběrnice rozhraní.

Komunikace na logické úrovni probíhá na mezi prostředky prvních dvou vrstev: funkční a zařízení (v horizontálním směru), skutečná komunikace probíhá mezi prostředky v mezi vrstvami (ve vertikálním směru) a na vrstvě rozhraní viz obr. č. 8. Každé zařízení obsahuje informace k identifikaci a nástroje pro nastavení. (5)



Obrázek 9 - Úrovně komunikace USB (5)

2.6 Protokol

Řazení bitů v paketu je klasické, podle standardních zvyklostí, od nejméně významného bitu (LSb – least-significant bit) po nejvíce významný bit (MSb – most-significant bit). Takové pořadí se používá proto, že, pokud dojde k chybě na začátku přenosu, chyba v setině nebo tisícině např. hodnoty měření není velká.

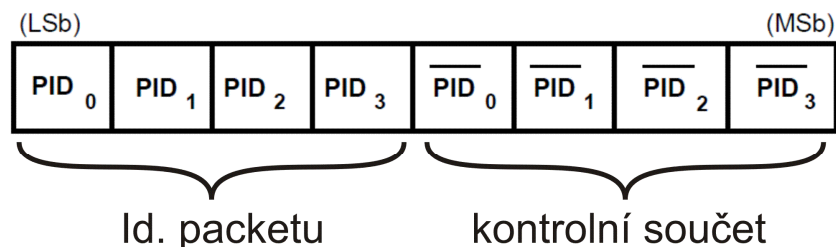
Rozdělujeme 16 typů paketu po čtyřech skupinách

- potvrzovací (handshake) - ACK, NAK, STALL, NYET (No Response Yet)
- pověřovací (token) - IN, OUT, SETUP, SOF
- datové (data) - DATA0, DATA1, DATA2, MDATA
- speciální (special) - PREAMBLE, ERR, Split, Ping

Všechna komunikace s pakety začíná synchronizačním polem (SYNC Field), to objeví se na sběrnici v nečinnosti. Synchronizační pole je kódováno s nejvyšší hustotou hran. Obvykle po synchronizačním poli následuje řetězec „KJKJKJKK“ ve svém vlastním kódování. Používá se pro synchronizaci lokálních hodin v délce osmi bitů. Poslední dva bity ukončují synchronizační pole a případně identifikují začátek PID (identifikační pole).

Synchronizační pole slouží pouze k synchronizaci zařízení a hodin, v diagramu se běžně nezakresluje.

Po identifikačním poli paketu okamžitě nesladuje synchronizační pole, to obsahuje čtyři bity typu paketu a čtyři kontrolní bity. PID indikuje typ paketu, formát paketu a druh způsob detekce chyby přenosu.



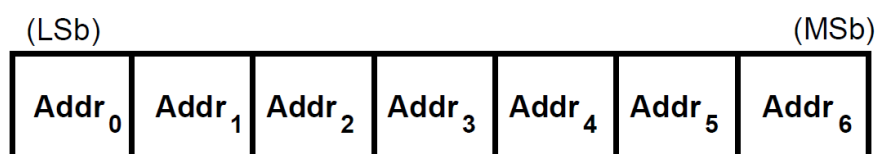
Obrázek 10 - Příslušné bity PID (5)

Typ PID	Název PID	PID [0..3]	Popis
Token (Známka) přenosu	OUT	0001B	Adresa + číslo koncového bodu transakce host - funkce
	IN	1001B	Adresa + číslo koncového bodu transakce funkce - host
	SOF	0101B	Značka začátku rámce a číslo rámce (START-OF-FRAME)
	SETUP	1101B	Adresa + číslo koncového bodu v transakci host - funkce pro instalaci řídicího potrubí (pipe)
Data	DATA0	0011B	Sudý data paket PID
	DATA1	1011B	Lichý data paket PID
Handshake (stvrzení)	ACK	0010B	Potvrzení přijetí paketu bez chyby
	NAK	1010B	Přijímající nemůže přijmout data nebo Vysílající nemůže data odeslat
	STALL	1110B	Koncový bod je pozastaven nebo potvrzení řídicí roury (pipe) není podporováno
Speciální	PRE	1100B	Host upozornil, že zařízení komunikuje na úrovni low-speed

Tabulka 3 - Typy PID překlad z (5)

Adresové pole (ADDR) specifikuje funkci zařízení, přes jeho adresu, to je buď zdroj nebo konec datového paketu, podle hodnoty PID. Adresové pole má délku sedm bitů, může tak adresovat až 127 zařízení, dva na sedmou bez defaultní adresy, která je vyhrazená pro zařízení mezi resetem a zapnutím – endpoint 0.

Výpočet: $2^7 - 1 = 128 - 1 = 127$.



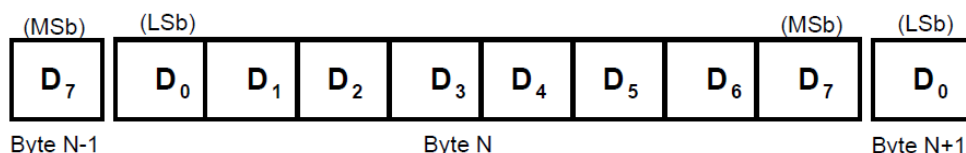
Obrázek 11 - Adresové pole (5)

Pole koncového bodu je definováno pouze u PID typu IN, OUT a SETUP. Všechna zařízení a jejich funkce musí podporovat řídicí datový kanál (pipeline) a defaultní koncový bod nula (endpoint 0).

Low-speed zařízení podporují pouze tři roury na funkci, Full-speed až šestnáct.

Pole čísla rámce je jedenácti-bitové číslo, které Host inkrementuje po každém rámci. Hodnota se přetočí po přesažení 7FFH a posílá se pouze známkou SOF na začátku každého rámce.

Datové pole je v rozsahu nula až 1023 bytů, počet bytů musí celé číslo. Datové pakety mohou být různé délky. (5)



Obrázek 12 - Datové pole (5)

2.7 Cyklická redundantní kontrola

CRC (Cyclic Redundancy Checks) se zde používají k ochraně všech paketů, které nejsou typu PID, PID mají vlastní kontrolní součet.

Výsledek CRC umísťuje odesílatel za odeslaný paket. Existují dvě možnosti jak přepočítat správnost přenosu podle (7):

1. Příjemce přečte paket, který se skládá z dat a z dvoubytového CRC.
Příjemce pro data spočítá CRC a porovná ji s přijatým CRC.
Pokud se obě hodnoty shodují, paket dorazil správně.
Příjemce buď potřebuje vědět dopředu, jak dlouhá jsou data (na USB sběrnici nemožné) a nebo bude počítat CRC s dvoubytovým zpožděním a při konci paketu bude jisté, že výpočet proběhl pouze nad daty a ne nad přijímaným CRC.
2. Příjemce přečte paket a CRC spočítá z celého jeho obsahu.
Cyklická redundantní kontrola obecně má tu vlastnost, že přijímáme-li stejnou posloupnost bitu, jaká byla obsahem CRC registru, pak dostaneme vždy stejný výsledek. CRC spočítané z celého paketu tedy musí vyjít pro každý správně přijatý paket vždy stejně.

Dále pak (7) uvádí dvě metody jak vypočítat CRC, první pochází z (5) ovšem zde je problém v pořadí bitů, proto je popsána druhá, výhodnější metoda.

Algoritmus výpočtu CRC:

1. počáteční hodnota CRC je 0x0000
2. $CRC = (CRC \text{ shr } 1) \text{ or } 0x8000$, přičemž vysunutý bit (LSb) je předmětem dalšího výpočtu
3. pokud je LSb stejný jako vstupní bit (XOR je 0), pak $CRC = CRC \text{ XOR } 0xA001$
4. pro všechny vstupní bity opakovat od bodu 2, jinak skončit
5. budeme-li CRC odesílat, můžeme jej použít bez jakékoliv úpravy
6. byl-li CRC počítán pro kontrolu příchozího paketu, musí se pro bezchybný paket rovnat zbytku 0x4FFE

Výpočet lze realizovat hardwarově pomocí klopných obvodů nebo softwarově pomocí programu.

2.8 USB On-The-Go (USB OTG)

Původně bylo USB míněno čistě jako rozhraní mezi osobním počítačem (PC) a přídatnými periferiemi, ale s rozmachem USB a mobilních zařízení se tato modifikace protokolu ukázala jako nezbytná. Zjednodušuje komunikaci, s OTG není nutné používat PC, ale zařízení lze spojit přímo, např. mobilní telefon s tiskárnou nebo úložným diskem.

Komunikace je zajištěna výměnou rolí zařízení, zařízení si vyměňují role Master – Slave tj. Host – Device, takové chování je označováno jako DRD (Dual-Role Device, Zařízení se dvěma rolemi). Z pohledu pozorovatele se může zdát, že zařízení jsou rovnocenná, na stejné úrovni, peer-to-peer, ovšem hierarchie Master – Slave je zachována. (8)

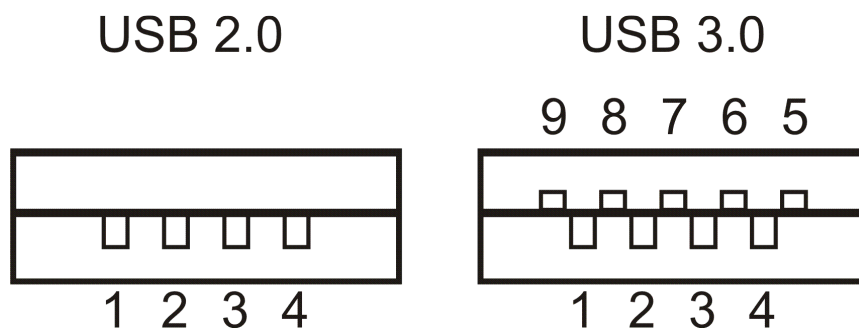
OTG lze nalézt většinou u 32-bitových MCU.

2.9 Verze 2.0 a 3.0

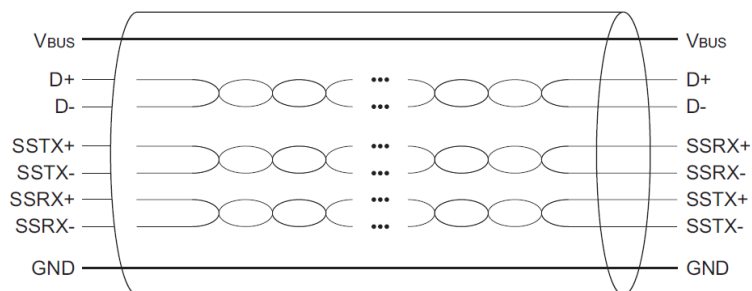
Rozdíly mezi verzí 1.1 a 2.0 jsou opravdu zanedbatelné, největší rozdíl je v přenosové rychlosti (12 Mbit/s vs. 480 Mbit/s), proto pro úplnost je zde srovnání verze 2.0 a 3.0. Tabulka č. 4 ukazuje zásadní rozdíly verzí 2.0 (1.1) a 3.0.

Funkce	USB 2.0	USB 3.0
Přenosová rychlost	480 Mbit/s	5 Gbit/s
Přenos	polo-duplexní, 2 datové vodiče	dual-simplex, 4+2 datové vodiče + signal return
Vodiče	2 datové, 2 napájecí	5 Super-Speed, 2 Hi-Speed, 2 napájecí
Typ přenosu	Synchronní	Asynchronní
Správa napájení	Primitivní (odpojeno, připojeno)	Inteligentní (idle, suspend, sleep)
Detekce připojení	Hardwarové detekce připojení portu, softwarový ovladač přepne zařízení do stavu zapnuto	hardwarová detekce s přechodem do provozního stavu pro datovou komunikaci
Barva výplně konektoru	Bílá	Modrá

Tabulka 4 - Srovnání USB 2.0 a 3.0



Obrázek 13 - USB Konektor TYP A



Obrázek 14 - Kabel USB 3.0 (9)

3 Známé metody řešení komunikace pomocí USB mezi mikro kontrolérem a počítačem

Níže jsou popsány a shrnuty tři metody využití USB pro komunikaci a práci s MCU.

Jsou zde jmenovány konkrétní možnosti realizace a jistě to není absolutní výčet všech, ale jsou zde jmenovány a popsány ty nejznámější.

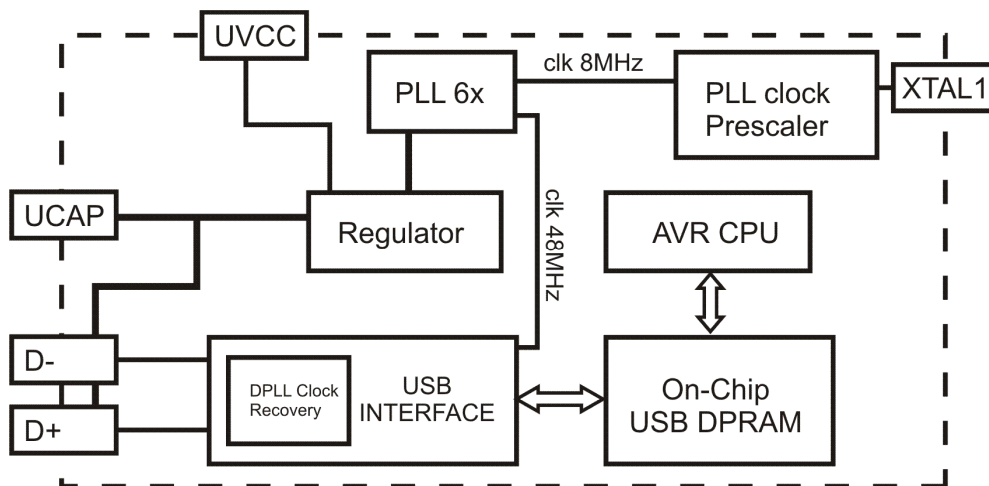
3.1 Integrovaný řadič USB na chipu mikro kontroléru

V pozdějších verzích 8-bitových obvodů AVR začal být řadič USB integrován na chipu, do této doby bylo nutné volit nějaké alternativní řešení, externí řadič nebo virtuální softwarovou emulaci. Je nutné zdůraznit, že Atmel přišel s těmito obvody později než konkurence. Největší předností integrovaného řadiče je, že může dosáhnou až na přenosovou rychlost Hi-speed zařízení, pokud CPU dodává odpovídající množství dat k přenosu a konkrétní MCU takto rychlý přenos podporuje na úrovni hardwaru.

Přehled dostupných obvodů Atmel AVR s řadičem USB je samozřejmě dostupný na internetových stránkách výrobce, včetně kompletní dokumentace obvodů, zde (4)

Nejzákladnější obvod představuje ATmega8U2 vycházející z ATmega8(L). Obvod obsahuje řadič USB 2.0 Full-speed 12 Mbit/s, tento řadič se skládá z regulátoru napájení/napětí, dovoluje datové vývody napájet napětím od 3,0 V do 3,6 V, přičemž napájecí napětí MCU je rozsahu 2,7 V až 5,5 V. Občas se v praxi setkáme s zařízením USB, jehož datová část pracuje na úrovni +5 V, to není v souladu s normou pro USB, a většina zařízení, hlavně základní desky PC, nebudou se zařízením vůbec komunikovat. Dále na blokovém schématu najdeme 176 bytů DP-RAM, systém je schopen pracovat v módu „ping-pong“ což znamená, že pracuje s dvojicí bufferů - vyrovnávací paměť, pro efektivnější vytěžování procesoru. Jednotka USB potřebuje konstantní zdroj hodin (48 MHz), u všech obvodů této řady ATmega je tzv. jednotka PLL (Phase Locked Loop - smyčka fázového závěsu).

Dále můžeme na blokovém schématu vidět datové vývody D+, D-. Vývod pro regulační kondenzátor napájení - UCAP, externí napájecí vývod jednotky USB - UVCC, vstup oscilátoru - XTAL1 a mikroprocesor AVR. (10)



Obrázek 15 - Blokové schéma USB řadiče

Pro práci s integrovaným řadičem je nutno zavést tzv. USB stack (stoh, komín), který řídí celý integrovaný řadič, jde o doplňující knihovnu programu MCU. Existuje mnoho možností, zde jsou dvě nejpoužívanější.

- AVR4900 – Oficiální komerční stack korp. Atmel pod limitovanou licenci.
- LUFA – OpenSource alternativa, velice kvalitně zpracován, podle některých názorů dokonce lépe než oficiální stack.

3.2 Samostatný integrovaný převodník USB a nativního rozhraní mikro kontroléru

Pokud MCU nepodporuje USB, podporuje jiné primitivnější rozhraní, k němu může připojit přídatný převodník mezi USB a nativním (přirozeným) rozhraním MCU viz Obrázek č.16.



Obrázek 16 - Schéma převodu USB/MCU

MCU AVR disponují sběrnicemi USART, SPI a I²C. Tyto sběrnice lze nalézt řadu převodníků v podobě integrovaných obvodů.

Výhodou je snadná a rychlá implementace, stačí fyzicky spojit prvky a upravit program MCU.

Nevýhodou je přidání několika dalších součástek k zařízení a zvýšení celkové ceny výrobku.

3.3 Virtuální softwarová implementace USB pomocí mikro kontroléru

MCU umožňuje v přesný čas uvést na potřebný MCU výstup logický signál, zároveň řadič přerušeni dovoluje zachytávat v přesný čas vstup na pin MCU. Těchto vlastností lze pomocí programu použít k emulaci protokolu USB.

Nevýhodou je, že tento program odebírá výpočetní čas MCU, proto pokud použijeme méně výkonný obvod, nezbyvá na další funkce mnoho výpočetního času.

Za nevýhodu lze také považovat, že softwarové implementace existují pouze jako low-speed zařízení, ale sběrnice, které MCU podporuje, nedosahují přenosové rychlosti 1,5 Mbit/s.

Výhodou je relativně snadná implementace a cena, software pro emulaci USB na MCU AVR je dostupný jako OpenSource (General Public License) nebo za poplatek pod komerční licencí, to už záleží na konkrétně zvoleném řešení.

Existuje mnoho řešení, zde jsou uvedeny tři, která jsou považována za nejznámější.

3.3.1 IgorPlug/AVR309 (11)

Autorem řešení je slovenský občan Igor Češko, jeho práce vznikla z potřeby rozhraní pro program Girder, který slouží pro zpracování infračerveného signálu dálkového ovladače např. od televizoru.

Jeho řešení převzala korporace Atmel pod označením AVR309. V manuálu AVR309 je dokonce původní schéma Igora Češka a jako autor je podepsán on sám. Dále jen AVR309.

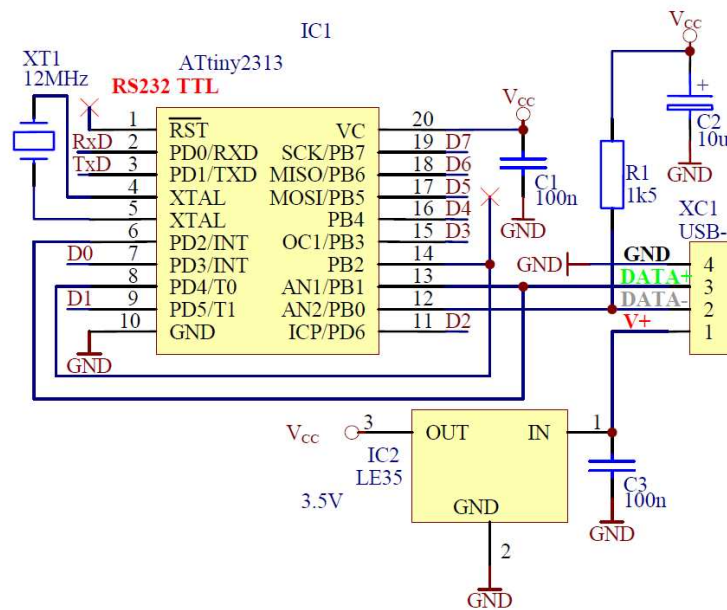
Výhody:

- Jasná a stručná dokumentace
- Snadná dostupnost, i cenová.

Nevýhody:

- Větší kód.
- Problém přesnosti časování a občasná chybovost přenosu.
- Program je psaný ve strojovém kódu, z toho vyplývá horší přehlednost programu a modularita programu.

Hardware AVR309



Obrázek 17 - Schéma zapojení AVR309 (11)

Problém rozdílu napájecího napětí USB (+5 V) a napětí datové části (+3,3 V) vyřešil autor použitím integrovaného polovodičového stabilizátoru (IC2 - LE35).

Jak je ihned patrné ze zapojení rezistoru R1 jde o low-speed zařízení.

Datové USB vodiče musí být na daných vývodech, protože zde jsou potřebné periferie pro komunikaci a program zapisuje data do daných registrů. Obvod disponuje 12 MHz oscilačním krystalem, odpovídajícím potřebnému časování pro USB. Piny 2 a 3 obvodu MCU (IC1) jsou vstupní a výstupní piny sběrnice UART, pro komunikaci na úrovni sběrnice RS232 je zapotřebí ještě převodníku např. MAX232.

Kondenzátory v obvodu mají pouze význam pouze jako filtrační a stabilizační kondenzátory.

Software AVR309

Veškerý příjem a dekódování probíhá na úrovni programu MCU, program nejdříve přijme řadu bitů jednoho paketu do bufferu. Začátek příjmu je identifikován na základě vnějšího přerušení na pin 6 MCU (INT0), je zajištěn příjem synchronizačního paketu. Během příjmu se kontroluje pouze konec paketu (EOP) z důvodu rychlosti přenosu USB a malé rychlosti zpracování MCU. Po úspěšném přijetí je paket teprve dekódován a analyzován. První proběhne kontrola, zda paket byl určen pro zařízení (MCU) podle adresy, dle toho zařízení pozná, že další data jsou mu určena. Dekódování musí proběhnout velice rychle, protože Host čeká na odpověď.

Po úspěšném přijetí napřed odstraní MCU bit stuffing a potom provede NRZI dekódování, tyto změny se provádějí do dalšího bufferu, dochází k vytvoření kopie. Rychlost dekódování už není tak kritická. Během dekódování může být přijat další paket. Zařízení zároveň musí provádět požadovanou akci, např. přeposlání přeložených dat na výstup pro RS232, pokud v tomto okamžiku se Host dotazuje, nemusí mu být odpovězeno, pokud se tak stane, Host pošle dotaz znovu, ovšem potom dochází k prodlevám. Mimo bit stuffingu a NRZI dekódování musí zařízení provádět výpočet a přepočet cyklické redundantní kontroly (CRC).

Po provedení všeho předchozího může teprve zařízení začít vysílat dle USB specifikace. Odesílání probíhá už v reálném čase protože, není tak náročné a na zařízení žádné jiné zařízení nečeká.

V příloze lze nalézt diagram algoritmu práce programu AV309.

Subrutiny pro vysílání jsou v manuálu velice stručně popsány.

Identifikace zařízení (VID a PID)

Každé zařízení v PC, včetně USB zařízení, má identifikační kód prodejce a identifikační kód procesu, (VID - vendor identifier, PID - process identifier). Seznam všech lze nalézt na internetové adrese (12).

AVR309 vlastní kód VID 0x03EB a PID 0x21FF, přičemž VID 0x03EB je vyhrazeno pro korp. Atmel. Proto v cílovém zařízení bychom měli použít svůj vlastní unikátní kód.

Doplňující informace

Na straně PC zajišťuje obsluhu Host s použitím ovladače v podobě dynamické knihovny (DLL), ovladač je podporován pod Windows98/ME/NT/XP.

Z důvodu použití 12 MHz oscilačního krystalu vzniká v přenosu chyba o maximální velikosti 4%. Ovladač zajistí konkrétní přenosovou rychlost, pouze když je chyba menší než 4%.

3.3.2 V-USB (AVR-USB) (13)

V-USB, původně pod názvem AVR-USB, je oproti AVR309 mnohem propracovanější a komplexnější softwarová implementace pocházející od Rakouské skupiny Objective development.

Výhody:

- Volně použitelné společné VID a PID.
- Možnost zakoupit si licenci od 9,90 € do 500 € (Hoby - Professional).
- Mnoho výborně zpracovaných ukázkových příkladů (rozsáhlá komunita).
- Multiplatformní podpora (Linux, Mac OS X, Windows).
- Použití různých oscilačních krystalů (12 MHz, 12.8 MHz, 15 MHz, 16 MHz, 16.5 MHz, 18 MHz a 20 MHz).
- Základní funkce jsou napsány v programovacím jazyku C a jsou bohatě okomentovány.
- Menší kód oproti AVR309.

Nevýhody:

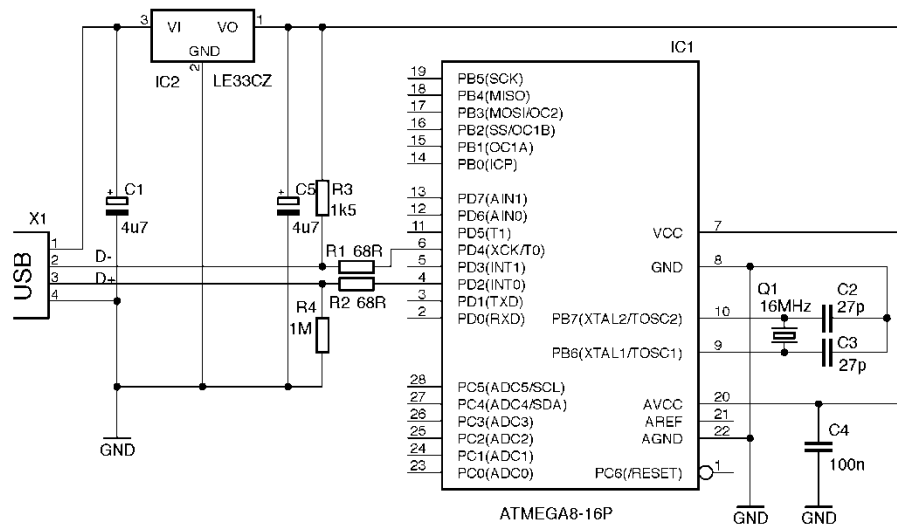
- Žádné?

Dokumentace je opravdu velice obsáhlá na druhou stranu je psána velice jednoduše a upozorňuje na všechny možné chyby.

Zařízení s V-USB se může prezentovat jako několik typů zařízení USB:

- Obecné zařízení, které potřebuje manuálně nainstalovat ovladač např. libusb nebo libusb-win32.
- Zařízení typu HID (Human Interface Devices – „Zařízení s rozhraním pro člověka“) pod podmínkou splnění požadavků na HID. Ovladač instaluje sám Host, s výjimkou Windows kde se stále musí použít libusb-win32. USB standart HID se používá pro klávesnice, myši, joysticky apod.
- Zařízení typu CDC, kde se zařízení hlásí a chová jako virtuální sériový COM port.

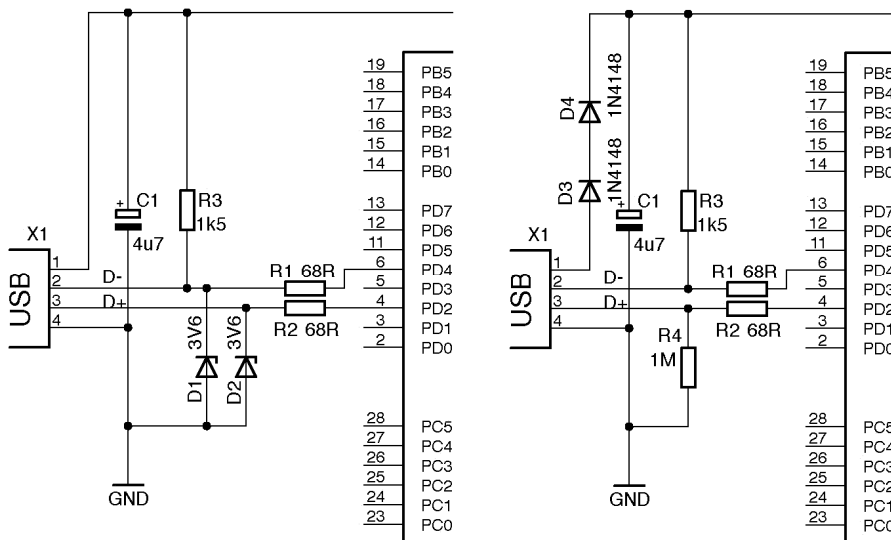
Hardware



Obrázek 18 - Schéma V-USB s LE33CZ (13)

Zapojení ve velice podobné AVR309. Jedinými prvky, které obvod odlišují jsou dva odpory na datových vodičích, které nemusí být použity, pokud nebude zařízení připojeno prostřednictvím kabelu, nýbrž přímo k PC. Dalším odlišným prvkem je použití pull-up rezistoru 1 M Ω (R4) k odstranění vzruchů na kabelu.

Rezistor R3 ukazuje na low-speed zařízení. S výjimkou kondenzátorů C2 a C3 u oscilačního krystalu, jsou všechny kondenzátory určeny jako stabilizační a filtrační prvky.



Obrázek 19 - Alternativní způsoby napájení (13)

Dokumentace nabízí tři způsoby napájení:

1. Použití integrovaného stabilizátoru 3,3 V s nízkým úbytkem napětí (LE33CZ), neefektivnější a zároveň nejdražší varianta. Doporučené použití pull-up rezistoru 1 M Ω (R4) k odstranění vzruchů na kabelu.
2. Stabilizace pomocí 3,6 V Zenerových diod, neefektivní varianta, ale levná.
3. Úbytek napětí pomocí dvou polovodičových diod, neefektivní, ale velice levná varianta. Doporučené použití pull-up rezistoru 1 M Ω (R4) k odstranění vzruchů na kabelu.

Software

Programové rutiny jsou v dokumentaci velice dobře popsány a vysvětleny. Budou zde uvedeny ty základní pro komunikaci s Hostem.

Navázání spojení pomocí endpoint 0

Osm bytů nastavení je popsáno v jazyce C, typu `usbRequest_t`, který deklarován v `usbdrv.h`:

```
typedef struct usbRequest{
    uchar        bmRequestType;
    uchar        bRequest;
    usbWord_t    wValue;
    usbWord_t    wIndex;
    usbWord_t    wLength;
}usbRequest_t;
```

`bmRequestType` – bitová maska obsahující směr toku dat

`bRequest` – identifikace žádostí na zařízení resp. žádostí o jeho funkce

`wValue` – pro žádosti dle USB specifikace

`wIndex` – pro žádosti dle USB specifikace

`wLength` – délka posílaného/přijímaného bloku

Odesílání dat (Zařízení → Host)

Existují dvě možnosti jak, blokově odesílat data:

1. Pokud jsou data ve statickém bufferu RAM, lze okamžitě vrátit blok funkcí `usbFunctionSetup()` viz.
2. Pokud jsou data generována za běhu, lze je posílat po 8 bytech. Řešení lze také použít u zařízení s malou kapacitou RAM.

Příklad aplikace první možnosti:

```
static uchar buffer[64];

usbMsgLen_t usbFunctionSetup(uchar setupData[8])
{
    usbRequest_t *rq = (void *)setupData; // cast to structured data for parsing
    switch(rq->bRequest){
    case VENDOR_RQ_READ_BUFFER:
        usbMsgLen_t len = 64; // we return up to 64 bytes
        if(len > rq->wLength.word) // if the host requests less than we have
            len = rq->wLength.word; // return only the amount requested
        usbMsgPtr = buffer; // tell driver where the buffer starts
        return len; // tell driver how many bytes to send
    }
    return 0; // ignore all unknown requests
}
```

Příklad aplikace druhé možnosti:

```
static uchar currentPosition, bytesRemaining;

usbMsgLen_t usbFunctionSetup(uchar setupData[8])
{
    usbRequest_t *rq = (void *)setupData; // cast to structured data for parsing
    switch(rq->bRequest){
    case VENDOR_RQ_READ_BUFFER:
        currentPosition = 0; // initialize position index
        bytesRemaining = rq->wLength.word; // store the amount of data requested
        return USB_NO_MSG; // tell driver to use
    }
}

usbFunctionRead()
{
    return 0; // ignore all unknown requests
}

uchar usbFunctionRead(uchar *data, uchar len)
{
    uchar i;
    if(len > bytesRemaining) // len is max chunk size
        len = bytesRemaining; // send an incomplete chunk
    bytesRemaining -= len;
    for(i = 0; i < len; i++)
        data[i] = getData(currentPosition); // copy the data to the buffer
    return len; // return real chunk size
}
```

Příjem dat (Host → Zařízení)

Pokud jsou data ve větším formátu, než lze přenést (tedy jak je definováno v `wValue` a `wIndex`) data jsou předána funkci `usbFunctionWrite()`, která je rozdělí na potřebné segmenty. Logická 1 musí být definována pro `USB_CFG_IMPLEMENT_FN_WRITE` v `usbconfig.h`, pokud chceme používat tuto metodu dělení dat.

Příklad aplikace:

```

static uchar buffer[64];
static uchar currentPosition, bytesRemaining;

usbMsgLen_t usbFunctionSetup(uchar setupData[8])
{
    usbRequest_t *rq = (void *)setupData; // cast to structured data for parsing
    switch(rq->bRequest){
    case VENDOR_RQ_WRITE_BUFFER:
        currentPosition = 0; // initialize position index
        bytesRemaining = rq->wLength.word; // store the amount of data requested
        if(bytesRemaining > sizeof(buffer)) // limit to buffer size
            bytesRemaining = sizeof(buffer);
        return USB_NO_MSG; // tell driver to use usbFunctionWrite()
    }
    return 0; // ignore all unknown requests
}

uchar usbFunctionWrite(uchar *data, uchar len)
{
    uchar i;
    if(len > bytesRemaining) // if this is the last incomplete chunk
        len = bytesRemaining; // limit to the amount we can store
    bytesRemaining -= len;
    for(i = 0; i < len; i++)
        buffer[currentPosition++] = data[i];
    return bytesRemaining == 0; // return 1 if we have all data
}

```

Kontinuální odesílání dat

Není možné, aby zařízení data libovolně odesílalo, lze ovšem zajistit aby Host data neustále vyžadoval. Dochází k tzv. streamování dat, kdy se nedostává žádné odezvy nebo potvrzení přijetí dat, musíme však zajistit, aby cílový buffer byl vždy volný.

Příklad aplikace:

```

if(usbInterruptIsReady()){ // only if previous data was sent
    uchar *p;
    uchar len = getInterruptData(&p); // obtain chunk of max 8 bytes
    if(len > 0) // only send if we have data
        usbSetInterrupt(p, len);
}

```

Kontinuální příjem dat

Jde o podobný princip jako při odesílání s tím, že je nutno před použitím nastavit logickou 1 v USB_CFG_IMPLEMENT_FN_WRITEOUT.

Příklad aplikace:

```

void usbFunctionWriteOut(uchar *data, uchar len)
{
    if(usbRxToken == 2){
        processInterruptOut2Data(data, len);
    }else{
        processInterruptOut4Data(data, len);
    }
}

```

Doplňující informace

Na straně Hosta zajišťuje funkci OpenSource ovladač libusb-win32.

4 Přehled dostupných řadičů pro USB v. 2.0 a jejich stručný popis (14)

V této kapitole je popsáno několik základních obvodů, zajišťující převod USB/UART, od nejvýznamnějších výrobců. Datové listy těchto výrobků bývají vyčerpávající a není cílem této práce je přepisovat.

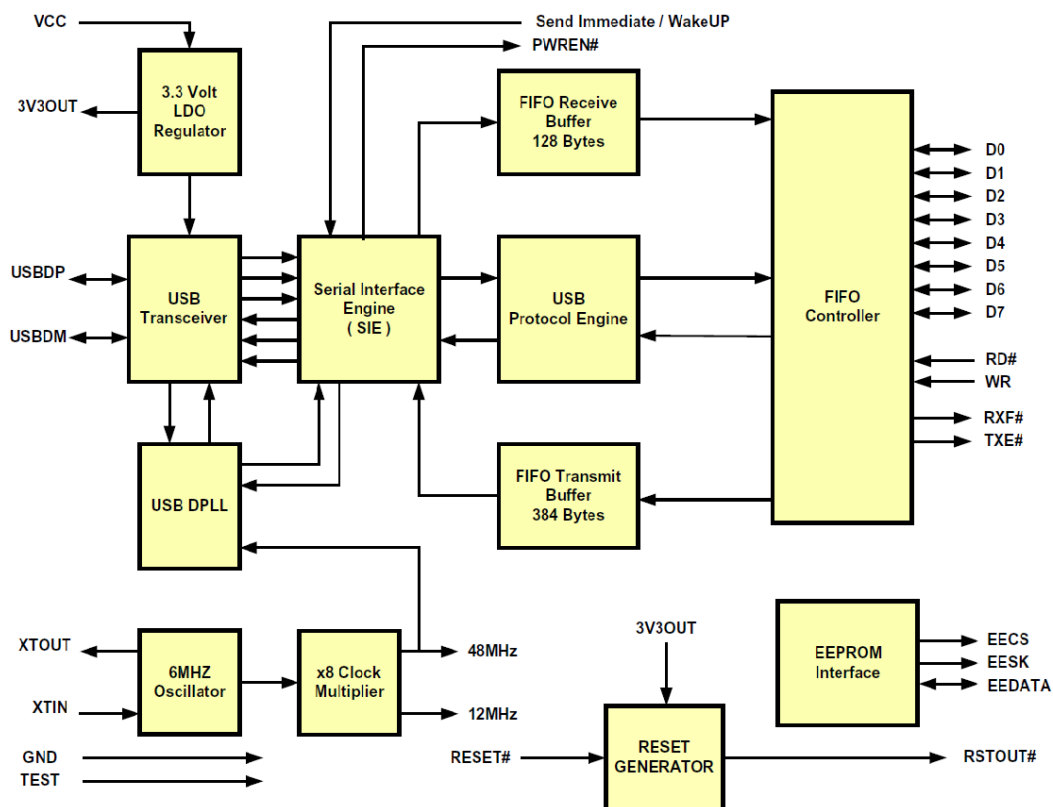
4.1 Řadiče firmy FTDI

Společnost FTDI (Future Technology Devices International Limited) je největším výrobcem zařízení zajišťujících konverzi – převod signálu z úrovně USB na úroveň sběrnic UART, I²C, SPI nebo pouze vstupy/výstupy typu FIFO (First In, First Out).

Nabízí například řešení, kde máme k dispozici Hi-speed USB na jedné straně a na druhé čtyři řadiče UART.

Obvod FT245BL (14)

Představuje vylepšenou verzi FT245BM, vylepšení spočívá v lepší správě napájení, integrovaném RC oscilátoru, možnosti 3,3 V logiky bez stabilizátoru, podpora Bit-Bang modu a mnohém dalším.



Obrázek 20 – Blokové schéma FT245BL (14)

Popis jednotlivých prvků FT245BL dle blokového schématu:

- Stabilizátor 3,3 V zajišťuje retenční napětí pro komunikaci na úrovni 3,3 V, vyžaduje připojení filtračního kondenzátoru na pin 3V3OUT. Pokud je potřeba může zajistit napájení do odběru 5 mA.
- USB Transceiver zajišťuje USB 1.1 / USB 2.0 full-speed fyzické rozhraní USB kabelu. Dva výstupní radiče zajišťují detekci připojení, diferenciální přenos a reset.
- USB DPLL zajistí příchozí NRZI kódování USB dat a poskytuje samostatné hodiny a datové signály do bloku SIE.
- 6MHz oscilátor zajišťuje referenční hodiny pro osminásobný hodinový násobič z externího 6MHz krystalu nebo keramického rezonátoru.
- Osminásobný hodinový multiplikátor má 6MHz vstup z oscilátoru a vytváří 12MHz referenční hodiny pro SIE. Také vytváří 48MHz referenční hodiny pro USB DPLL.
- Jednotka sériového rozhraní (Serial Interface Engine – SIE) provádí převod paralelních dat na sériová data a obráceně pro převod dat USB. V souladu se specifikací USB 2.0, provede bit stuffing / un-stuffing a CRC5 / CRC16 kontrolu toku dat na USB.
- Jednotka USB protokolu řídí tok dat z koncového zařízení kontrolní USB. To se stará o žádosti hostitelského radiče Low-speed USB protokolu, a příkazy pro ovládání parametrů funkcí FIFO.
- FIFO přijímací buffer (First In, First Out – První dovnitř, první ven) (128 bytů) Data odeslaná od USB Hosta do FIFO přes datový USB endpoint jsou uložena ve vyrovnávací paměti FIFO, a potom odstraněna z vyrovnávací paměti přečtením FIFO pomocí RD#.
- Vyrovnávací buffer pro vysílání FIFO (384 bytů) Data zapsaná do FIFO pomocí WR# jsou uložena ve vyrovnávací paměti FIFO. Hostitel odstraní data z FIFO na základě žádosti zasláné pro data v zařízení USB endpointu.
- FIFO radič ovládá přenos dat mezi externími piny FIFO rozhraní a FIFO vysílací a přijímací vyrovnávací paměti.

- **Reset Generátor**
poskytuje spolehlivé zapnutí a nastavení vnitřních obvodů. RESET# vstup a výstup #RSTOUT slouží dalším zařízením pro reset FT245BL, nebo je-li potřeba resetovat pomocí FT245BL jiná zařízení, při resetu musí být #RSTOUT na úrovni log. 0, jinak to vede ven 3,3 V poskytovaným regulátorem. RSTOUT# může sloužit k ovládní 1,5 k Ω pull-up rezistoru na USBDP přímo tam, kde je zapotřebí výčet opoždění USB.
- **EEPROM rozhraní**
(EEPROM - Electrically Erasable Programmable Read-Only Memory)
Ačkoli FT245BL bude fungovat bez volitelného EEPROM, lze pomocí paměti EEPROM přizpůsobit USB VID, PID, sériové číslo, struny a power management na úrovni FT245BL pro aplikace OEM (Original Equipment Manufacturer). EEPROM by měla být 16 bitů konfiguračně např. jako 93LC46B nebo ekvivalentní schopné přenosové rychlostí 1MB/s, VCC = 4,35 V až 5,25 V. EEPROM je programovatelná, na desce plošného spoje přes USB pomocí utility, která je k dispozici na webových stránkách FTDI. To umožňuje, aby paměť na desce plošných spojů byla programována v rámci výrobního procesu a testování.
- Pokud není připojen EEPROM (nebo je EEPROM prázdný), bude FT245BL použit vestavěný výchozí VID, PID popis a výchozí hodnoty power managementu. V tomto případě zařízení nebude mít seriové číslo jako součást USB deskriptoru.

Důležitou doplňkovou funkcí FIFO řadiče je možnost jeho využití jako obecné vstupně/výstupní porty tzv. GPIO (General Purpose Input/Output), této technice se také říká Bit-Bang mode, jde o „hrubé“ použití vstupů a výstupů FIFO řadiče např. pro ovládní řídicích relé nebo spínacích tranzistorů. Můžeme FIFO řadič používat pro logické výstupy a vstupy v širokém spektru aplikací.

Obvod je kompatibilní s USB 2.0 a zpětně kompatibilní s USB 1.1, podporuje Full-Speed komunikaci s přenosovou rychlostí až 12 Mbit/s.

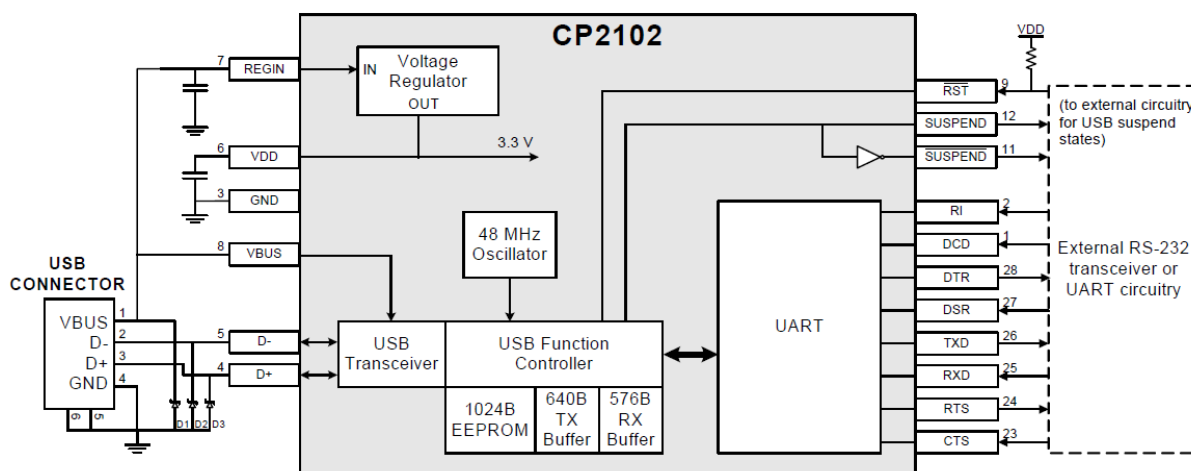
Dále výrobce uvádí v manuálu standardní způsoby napájení přímo z USB nebo ze separátního zdroje, +5 V nebo 3,3 V se stabilizátorem.

4.2 Řadiče firmy Silicon Labs

Silicon Labs můžeme považovat za největší konkurenci FTDI. Jejich obvody jsou považovány za špičkové. Firma Silicon Labs není zaměřena čistě na obvody pro práci s USB, ve své nabídce má mnoho produktů.

Obvod CP2102 (15)

CP2102 představuje převodník rozhraní USB na UART (RS232). Blokové schéma je velice podobné obvodu FTDI s rozdílem FIFO a UART jednotky.



Obrázek 21 - Blokové schéma CP2102 (15)

Charakteristika UART části obvodu:

- Podpora všech Handshake funkcí
- Šířka datového bitu: 5, 6, 7, a 8
- Stop bity: 1, 1.5, a 2
- Parita: lichá, sudá, značka, prostor, bez parity
- Přenosová rychlost UART: 300 bps až 1 Mbits
- 576 Bytů přijímací buffer; 640 byte vysílací buffer

Obecné vlastnosti obvodu:

- Vendor ID 10C4h (VID) Product ID EA60h (PID)
- Vlastí integrovaný regulátor s nízkým úbytkem napětí 3,3 V
- Samostatné napájecí napětí: 3.0 až 3.6 V
- Napájecí napětí z USB: 4.0 až 5.25 V

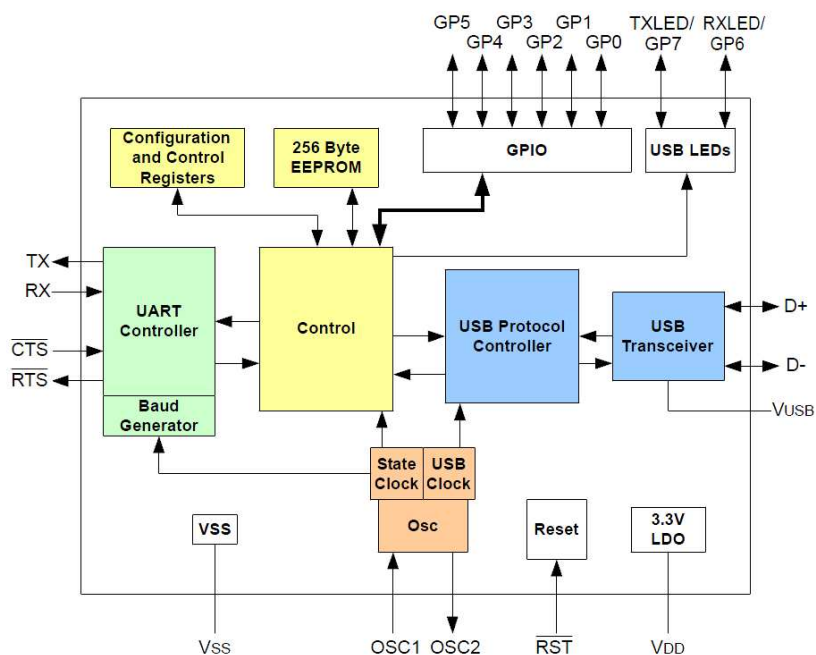
4.3 Řadiče firmy MICROCHIP

MICROCHIP je největší konkurence korp. Atmel v oblasti výroby MCU, ve svém produktovém portfoliu nabízí jako doplněk jeden převodník USB/UART.

Obvod MCP2200 (16)

Představuje obdobu předcházejících obvodů ovšem s vylepšeným využitím GPIO.

Stabilizátor napětí 3,3 V je integrován na chipu, ovšem oscilační krystal se musí doplnit jako diskretní součástka, MCP2200 vyžaduje krystal o rezonanční frekvenci 12 MHz.



Obrázek 22 - Blokové schéma MCP2200 (16)

Výrobce k MCP2200 dodává nástroj pro velice snadnou konfiguraci vnitřní EEPROM, která nese nastavení obvodu. Umožňuje konfigurovat:

- Vendor ID (VID)
- Product ID (PID)
- Přenosovou rychlost UART
- Individuální nastavení vstupně/výstupních vývodů
- Nastavení Tx/Rx LED indikující vysílání a příjem
- Hardwarové řízení komunikace pomocí CTS a RTS
- Invertování signalizace suspendace obvodu
- Řetězec identifikující výrobce a produkt

Vývoj aplikací usnadňuje dynamická knihovna od výrobce pro Microsoft NET Framework 3.5, kterou lze využít s implementací jazyka Visual C++.

V rozsahu 300 b/s až 19200 b/s přenosové rychlosti UART výrobce udává chybovost přenosu 0,00%, v rozsahu 38400 b/s až 921600 b/s chybu 0,16%.

5 Koncepční návrh implementace v mikroprocesorech Atmel AVR

Příkladem praktického využití MCU AVR a USB v zemědělství je např. váha pro tzv. precizní krmení, u precizního krmení se snažíme aby, zvířata dostávala přesně tolik kolik, potřebují jak z pohledu množství, tak z pohledu složení, s minimální chybou.

5.1 Váhový počítač (17)

Pro úplnost je zde krátce popsán váhový počítač, ke kterému se připojuje patrona s recepty.



Obrázek 23 - Váhový počítač Computer Scale RM plus 2 (17)

Základní funkce váhového počítače:

- Řízení nakládky a vykládky podle receptů vkládaných z klávesnice.
- Řízení nakládky a vykládky podle receptů přenesených paměťovou patronou z manažerského programu osobního počítače farmáře. Je umožněn také zpětný přenos dat z váhového počítače, který zahrnuje seznam skutečně realizovaných nakládek a vykládek s časovým údajem, kódem řidiče, stájí, názvem a hmotností komponenty včetně výpočtu odchylky od požadované hmotnosti.
- Nakládka a vykládka zadané hmotnosti z klávesnice váhového počítače s alarmem splnění naložení respektive vyložení hmotnosti.
- Servisní menu pro konfiguraci režimu činnosti váhového počítače včetně kalibrace. Konfigurace umožňuje nastavit datum a hodiny, rozlišení v kg, režim alarmu, řízení jasu displeje, přihlášení / heslo obsluhy, způsob přechodu na další komponentu.

Základní technické parametry:

Rozsah vážení:	0 až 30000 kg
Nastavitelné rozlišení:	1; 2; 5; 10 kg
Počet receptů:	až 20
Počet komponent:	až 20
Počet stájí:	až 20
Přechod komponent v receptu:	automatický nebo manuální
Kapacita paměťové patrony:	512 kB (20 receptur plus uložení reportu pro 1236 událostí).

Zobrazení na maticovém LED displeji – zobrazení čísel i textu

Řízení jasu displeje – automatické senzorem nebo manuální v servisním menu.

Váhový počítač poskytuje komunikaci v českém jazyce a pro každou prováděnou funkci okamžitou radu – pomoc (HELP) stiskem tlačítka „?“.

5.2 Čtecí zařízení



Obrázek 24 - Patrona a čtecí zařízení (17)

Základní funkce (17):

- 1. Vytváření databáze komponent**, obsahující plný název komponenty, kód komponenty – zkrácený název pro zobrazení na váhovém počítači, označení dostupnosti komponenty ve skladu.
- 2. Vytváření databáze stájí**, obsahující plný název stáje, kód stáje- zkrácený název pro zobrazení na váhovém počítači, označení provozu stáje – aktivity.
- 3. Vytváření databáze receptur**, obsahující plný název receptury, kód receptury – zkrácený název pro zobrazení na váhovém počítači, počet kusů zvířat

pro recepturu, označení použití receptury – aktivity, jen aktivní receptury budou zapsány do paměťové patrony.

4. Přenos dat mezi patronou a programem – zápis receptur na patronu, načtení realizovaných nakládek a vykládek (žurnálu), mazání žurnálu v patroně a servisní činnosti kam patří test čtečky a formátování patrony. Pozn.: Formátování patrony umožňuje i váhový počítač.

5. Databáze nakládek a vykládek – historie událostí

Tato funkce slouží k archivaci skutečných pracovních činností (nakládky a vykládky), které byly přeneseny do programu čtením obsahu paměťové patrony v režimu. Přenos dat z otevřené historie událostí je možné realizovat export dat do tabulek ve formátu sešitu Excel.

6. Důležitou funkcí je zálohování databáze – uložení a také opětovné načtení vytvořených databází komponent, stájí a receptur do archivačního souboru v paměti počítače (pevný disk, flash disk atd.).

Technické provedení

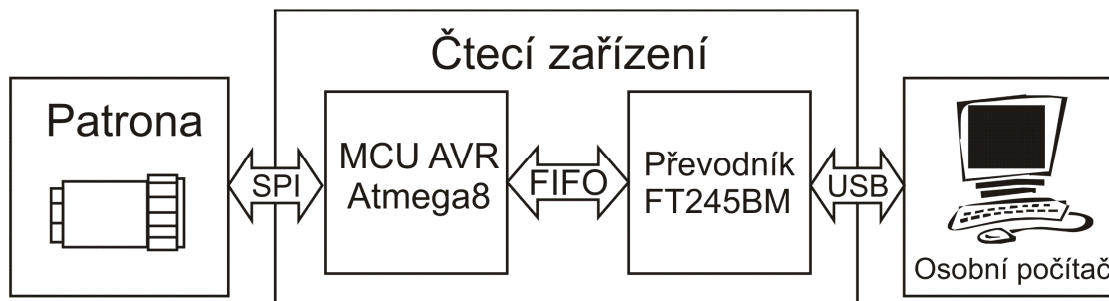
Jak na straně váhového počítače, tak na straně patrony a čtecího zařízení je dbáno na vysokou odolnost, spolehlivost a životnost, použitím konektorů třídy MIL (MILitary – Vojenské), které jsou vyráběny pro splnění nejvyšších požadavků vojenského sektoru. Zaručují vysokou mechanickou odolnost, vysoký stupeň mechanického krytí a vodotěsnosti.



Obrázek 25 - 7 pinový konektor třídy MIL

Komunikace čtecího zařízení

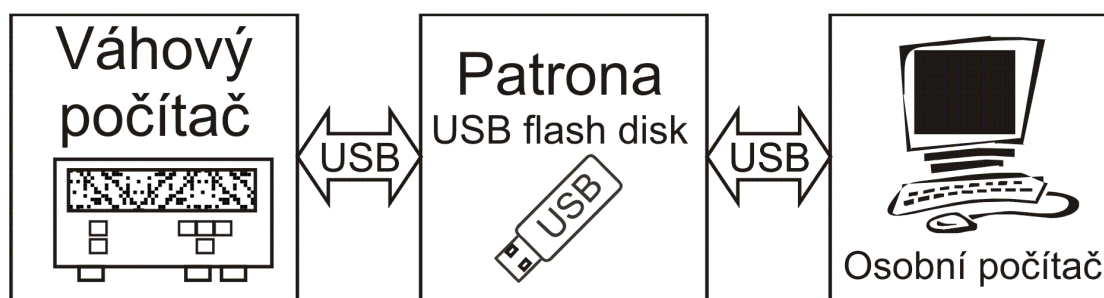
Realizaci čtecího zařízení ukazuje blokové schéma nebo podrobněji schéma zapojení v příloze.



Obrázek 26 - Blokové schéma čtecího zařízení

Jak je patrné z blokového schématu, dochází třikrát k převodu dat, prvně při čtení z EEPROM paměti patrony pomocí sběrnice SPI, podruhé převádí MCU data pro převodník FT245BM způsobem FIFO, třetí a konečný převod se odehrává v převodníku FT245BM na sběrnici USB, v opačném směru k tomu dochází při zápisu receptur z osobního počítače.

Vzhledem k tomu, že dnes jsou již dostupné obvody Atmel s integrovaným řadičem USB na chipu, podporující USB OTG, bylo by možné tento mezi článek v podobě čtecího zařízení vynechat a použít takový MCU ve váhovém počítači a jako patronu použít klasický flash disk. Velice se tak celé zařízení a nebude potřeba několikrát převádět data na různé sběrnice a vše bude pracovat na úrovni USB, viz blokové schéma zjednodušeného modelu.



Obrázek 27 - Blokové schéma zjednodušeného modelu

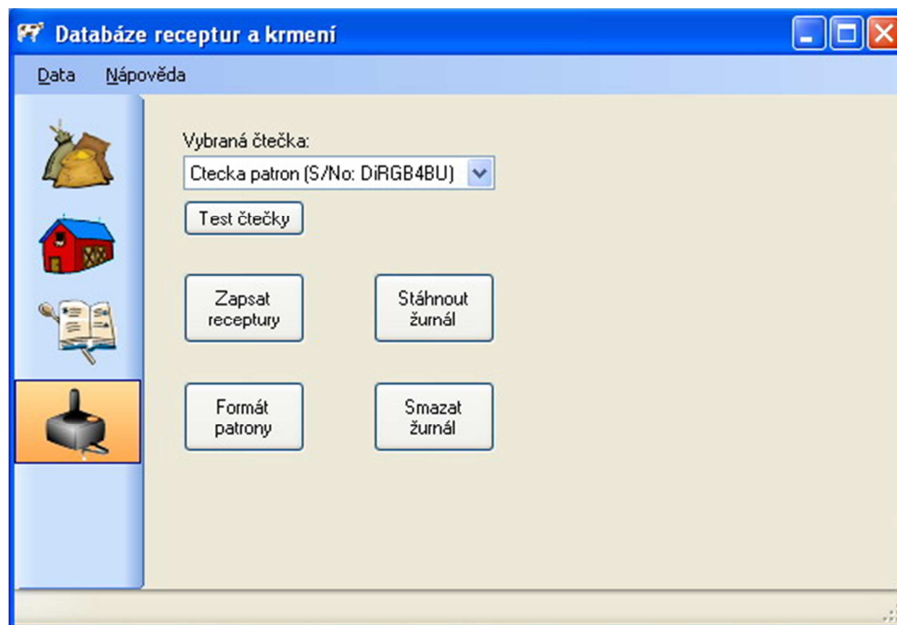
5.3 Manažerský program (17)

Pro práci s patronou slouží manažerský program „Precizní krmení“, který již byl zmíněn výše, zde je několik obrázků z uživatelského rozhraní a jejich krátký popis. Celý program je velice jednoduše vypracován a jeho ovládání je intuitivní.

První obrázek (Obr. č. 28) ukazuje možnosti práce s patronou. Vidíme zde možnost otestování čtečky, pro ověření funkčnosti komunikace.

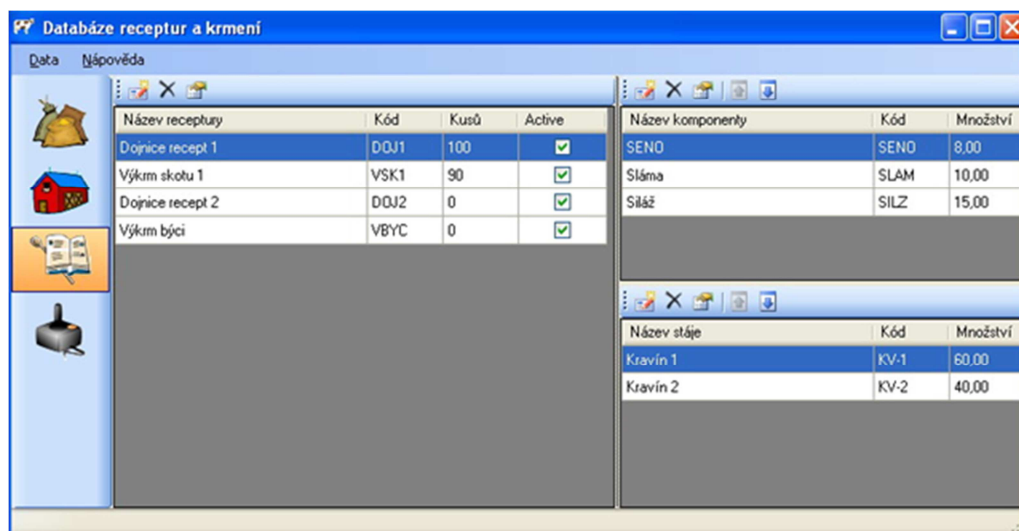
Zápis, stažení a smazání receptury pro konkrétního řidiče krmného vozu.

Dále zde je možnost naformátovat čtečku, tuto možnost nabízí také váhový počítač.



Obrázek 28 - Práce s patronou (17)

Obrázek č. 29 ukazuje možnosti vytváření, upravování a mazání receptů.



Obrázek 29 - Ukázka receptů (17)

Datum a čas	Operace	Operátor	Receptura	Komponenta/stáj	Hmotnost [kg]	Požadováno [kg]	Chyba [%]
20.6.2010 15:12	naloženo	BOBEK	DOJ1	SLAM	356	350	1,7
20.6.2010 15:13	naloženo	BOBEK	DOJ1	SENO	402	400	0,5
20.6.2010 15:13	naloženo	BOBEK	DOJ1	SENZ	1018	1000	1,8
20.6.2010 15:13	naloženo	BOBEK	DOJ1	GPS	214	200	7,0
20.6.2010 15:15	vyloženo	BOBEK	DOJ1	DOJ1	402	398	1,0
20.6.2010 15:21	naloženo	BOBEK	DOJ1	SLAM	352	350	0,6
20.6.2010 15:21	naloženo	BOBEK	DOJ1	SENO	404	400	1,0
20.6.2010 15:21	naloženo	BOBEK	DOJ1	SENZ	1000	1000	0,0
20.6.2010 15:21	naloženo	BOBEK	DOJ1	GPS	200	200	0,0
20.6.2010 15:22	vyloženo	BOBEK	DOJ1	DOJ1	1960	1954	0,3
26.6.2010 15:27	naloženo	BOBEK	DOJ1	SLAM	356	350	1,7
26.6.2010 15:27	naloženo	BOBEK	DOJ1	SENO	406	400	1,5
26.6.2010 15:28	naloženo	BOBEK	DOJ1	SENZ	1004	1000	0,4
26.6.2010 15:28	naloženo	BOBEK	DOJ1	GPS	210	200	5,0
26.6.2010 15:29	vyloženo	BOBEK	DOJ1	DOJ1	1976	1972	0,2

Obrázek 30 - Ukázka nakládek (17)

Obrázek č. 30 ukazuje jednotlivé nakládky řidiče, můžeme zde vidět sloupec Operace informujícím, zda již nakládka proběhla, uživatelské jméno řidiče, kód receptury, kód stáje, požadovanou hmotnost k naložení, skutečně naloženou hmotnost a vypočtenou chybu resp. odchylku.

6 Závěr

Existuje mnoho možností využití mikro kontrolérů, setkáme se s nimi např. v našem mobilním telefonu, mikrovlnné troubě, hudebních přehrávačích, průmyslových řídicích zařízeních a v automobilech. Na druhé straně rozhraní USB je také velice rozšířené a používané.

Představení nástrojů mikro kontrolérů, v první kapitole, by mělo vytvořit základní představu, jaké jsou jejich množnosti. Popis periferií je krátký, předpokládá znalost těchto periferií, měl by osvěžit znalosti odborníka a nastínit jejich smysl tomu kdo není z oboru.

Problematika rozhraní USB je rozsáhlá, samotný popis rozhraní USB daleko přesahuje rámec této práce, přesto jsou zde na několika stranách shrnuty základní vlastnosti tohoto velice rozšířeného rozhraní.

Uvedení hlavních způsobů řešení implementace USB do obvodů s mikro kontroléry AVR a charakteristika jejich výhod resp. nevýhod, je popsána hlavními prvky konkrétních řešení.

Konkrétní řešení lze rozdělit na tři skupiny diskrétní hardwarové řešení, softwarové řešení a smíšené řešení integrované již v mikro kontroléru od výrobce.

V páté kapitole je představeno reálné řešení z praxe, které názorně ukazuje slabá místa problematiky tj. nutnost trojnásobného převodu.

Práce vznikla, aby došlo k přehlednější a vyřešení některých otázek na téma implementace USB v obvodech využívajících mikro kontroléry AVR.

Většina informací a poznatků vychází z manuálů a datových listů výrobců nebo skupin, které se problematikou zabývají a vytvořili nějaké řešení.

Vzhledem k tomu, že vědecko-technický rozvoj probíhá velice rychle v době zadávání této práce, nebylo možné, z důvodů dostupnosti konkrétních mikro kontrolérů, volit třetí řešení kde je řadič USB integrován v mikro kontroléru, ale v průběhu vytváření této práce představila korporace Atmel novou řadu mikro kontrolérů, je nejvýhodnější využít těchto chybějících obvodů z důvodu přiměřeného a dostatečného výpočetního výkonu a nízké ceny oproti ostatním řadám mikro kontrolérů.

Do budoucna by bylo dobré postupně nahrazovat všechna zařízení resp. řešení, která vznikla v důsledku chybějícího segmentu integrovaných obvodů s integrovaným řadičem USB a architekturou AVR, dojde tak ke zlevnění výrobků a k jejich zjednodušení při výrobě a servisu, výrobky se budou skládat z méně komponent, které by mohli způsobit poruchu a hledání poruchy se tak značně zjednoduší.

7 Citovaná literatura

1. **Atmel.** Fact Sheet. *Atmel Corporation*. [Online] [Citace: 5. 1 2012.]
Dostupné z: <http://www.atmel.com/about/corporate/factsheet.aspx>.
2. **Atmel.** Datasheet ATmega8(L). *Atmel Corporation*. [Online] [Citace: 20. 1 2012.]
Dostupné z: www.atmel.com/Images/doc2486.pdf.
3. **avrtvtube.** Youtube.com. *The Story of AVR*. [Online] 6. 12 2008. [Citace: 20. 1 2012.]
Dostupné z: <http://www.youtube.com/watch?v=HrydNwAxbcY>.
4. **Atmel.** AVR 8- and 32-bit Microcontrollers. [Online] [Citace: 7. 3 2012.]
Dostupné z: <http://www.atmel.com/products/microcontrollers/avr/default.aspx>.
5. **USB.org.** Universal Serial Bus 1.1. *Embedded System Design*. [Online] Revision 1.1.
[Citace: 29. 2 2012.] Dostupné z: <http://esd.cs.ucr.edu/webres/usb11.pdf>.
6. **Kainka, Burkhard.** *USB - měření, řízení a regulace pomocí sběrnice USB*. Praha : BEN -
Technická literatura, 2002. 80-7300-073-3.
7. **Smrž, Jan.** *Implementace USB rozhraní AVR mikroradicem*. [Online] 12. 1 2008. [Citace:
25. 1 2012.] Dostupné z: <http://www.smrz.chrudim.cz/abstrakt/smrz-dp.pdf>.
8. **Vojáček, Antonín.** Co se skrývá pod komunikací označenou jako USB OTG? *HW.CZ*.
[Online] HW server s.r.o., 28. 5 2008. [Citace: 14. 3 2012.]
Dostupné z: <http://www.hw.cz/teorie-a-praxe/dokumentace/co-se-skryva-pod-komunikaci-oznacenu-jako-usb-otg.html>.
9. **USB.org.** USB 3.0 Specification. *USB Implementers Forum, Inc.* [Online] 6. 6 2011.
[Citace: 22. 3 2012.]
Dostupné z: http://www.usb.org/developers/docs/usb_30_spec_022412.zip.
10. **Atmel.** Datasheet ATmega8U2. *Atmel Corporation*. [Online] [Citace: 5. 2 2012.]
Dostupné z: <http://www.atmel.com/Images/doc7799.pdf>.
11. **Atmel.** AVR309. [Online] 23. 1 2012. [Citace: 3. 31 2012.]
Dostupné z: www.atmel.com/Images/doc2556.pdf.
12. **PCI Vendor and Device Lists.** [Online] [Citace: 2. 2 2012.]
Dostupné z: <http://www.pcidatabase.com/vendors.php?sort=id>.
13. **Objective Development Software, GmbH.** Overview - V-USB. *V-USB*. [Online]
[Citace: 4. 2 2012.] Dostupné z: <http://vusb.wikidot.com/>.

14. **FTDI**. FT245B. *Future Technology Devices International*. [Online] [Citace: 3. 4 2012.]
Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT245BL.pdf.
15. **Silicon Labs**. CP2102. [Online] [Citace: 4. 4 2012.]
Dostupné z: <http://www.silabs.com/Support%20Documents/TechnicalDocs/cp2102.pdf>.
16. **Microchip**. MCP2200. *MICROCHIP*. [Online] [Citace: 5. 4 2012.]
Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/22228B.pdf>.
17. **Bohuslávka, Zdeněk**. Precizní krmení. 6. 3 2011. ČZU Praha, Technická fakulta, katedra Elektrotechniky a automatizace [Citace: 4. 7 2012.]

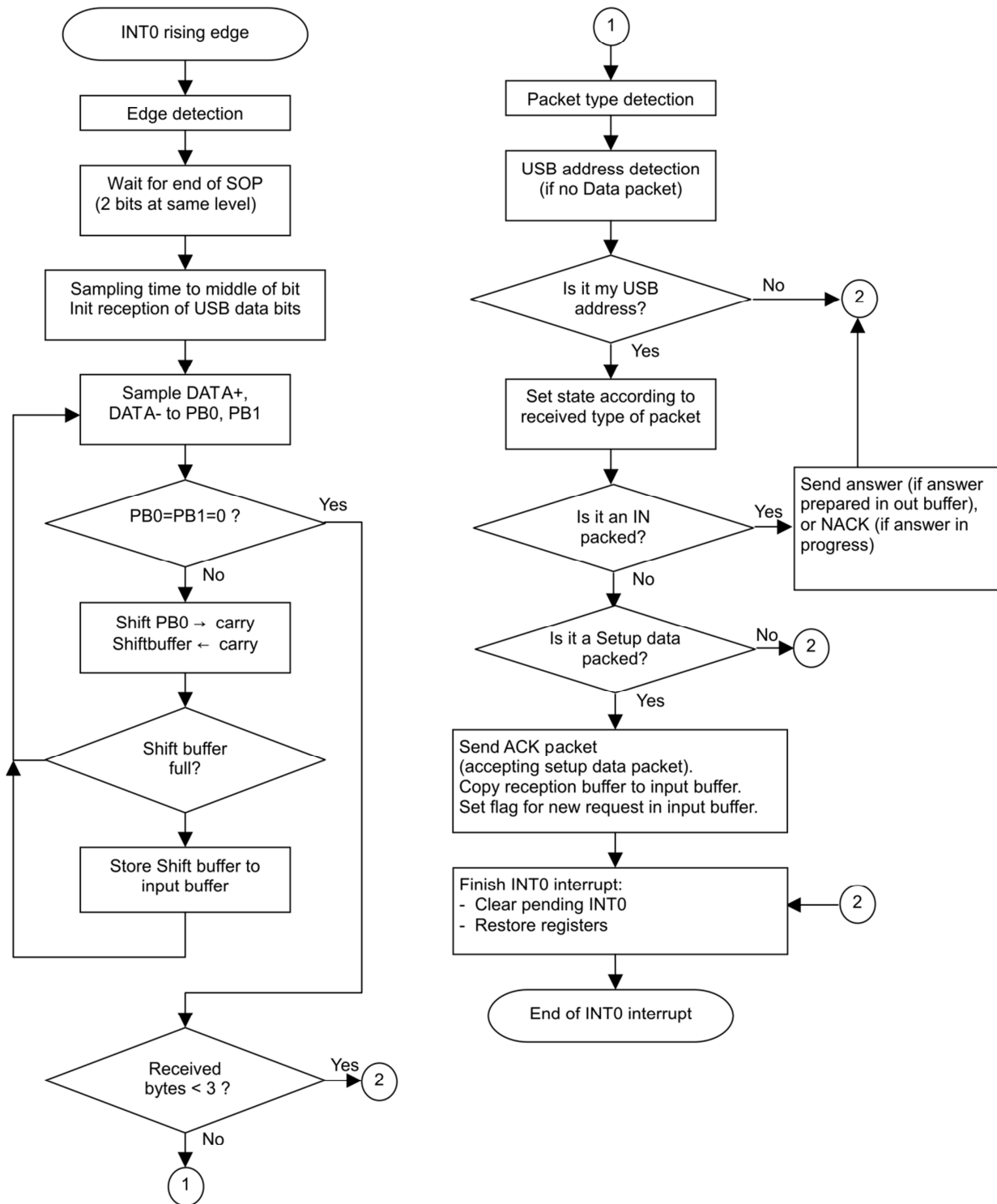
8 Seznam obrázků

Obrázek 1 - Logo Korporace Atmel (2)	1
Obrázek 2 - Blokový diagram Arch. AVR (2)	2
Obrázek 3 - Topologie USB (5)	6
Obrázek 4 - Budící hradla (5).....	7
Obrázek 5 - Struktura kabelu USB (5).....	8
Obrázek 6 - Použité materiály (5)	8
Obrázek 7 - Konektory USB (5)	9
Obrázek 8 - NRZI kódování.....	9
Obrázek 9 - Úrovně komunikace USB (5).....	10
Obrázek 10 - Příslušné bity PID (5).....	11
Obrázek 11 - Adresové pole (5)	12
Obrázek 12 - Datové pole (5).....	13
Obrázek 13 - USB Konektor TYP A.....	15
Obrázek 14 - Kabel USB 3.0 (9).....	15
Obrázek 15 - Blokové schéma USB řadiče	17
Obrázek 16 - Schéma převodu USB/MCU	17
Obrázek 17 - Schéma zapojení AVR309 (11).....	19
Obrázek 18 - Schéma V-USB s LE33CZ (13)	22
Obrázek 19 - Alternativní způsoby napájení (13)	22
Obrázek 20 – Blokové schéma FT245BL (14)	26
Obrázek 21 - Blokové schéma CP2102 (15).....	29
Obrázek 22 - Blokové schéma MCP2200 (16)	30
Obrázek 23 - Váhový počítač Computer Scale RM plus 2 (17).....	31
Obrázek 24 - Patrona a čtecí zařízení (17)	32
Obrázek 25 - 7 pinový konektor třídy MIL.....	33
Obrázek 26 - Blokové schéma čtecího zařízení	34
Obrázek 27 - Blokové schéma zjednodušeného modelu.....	34
Obrázek 28 - Práce s patronou (17).....	35
Obrázek 29 - Ukázka receptů (17)	35
Obrázek 30 - Ukázka nakládek (17).....	36

9 Seznam tabulek

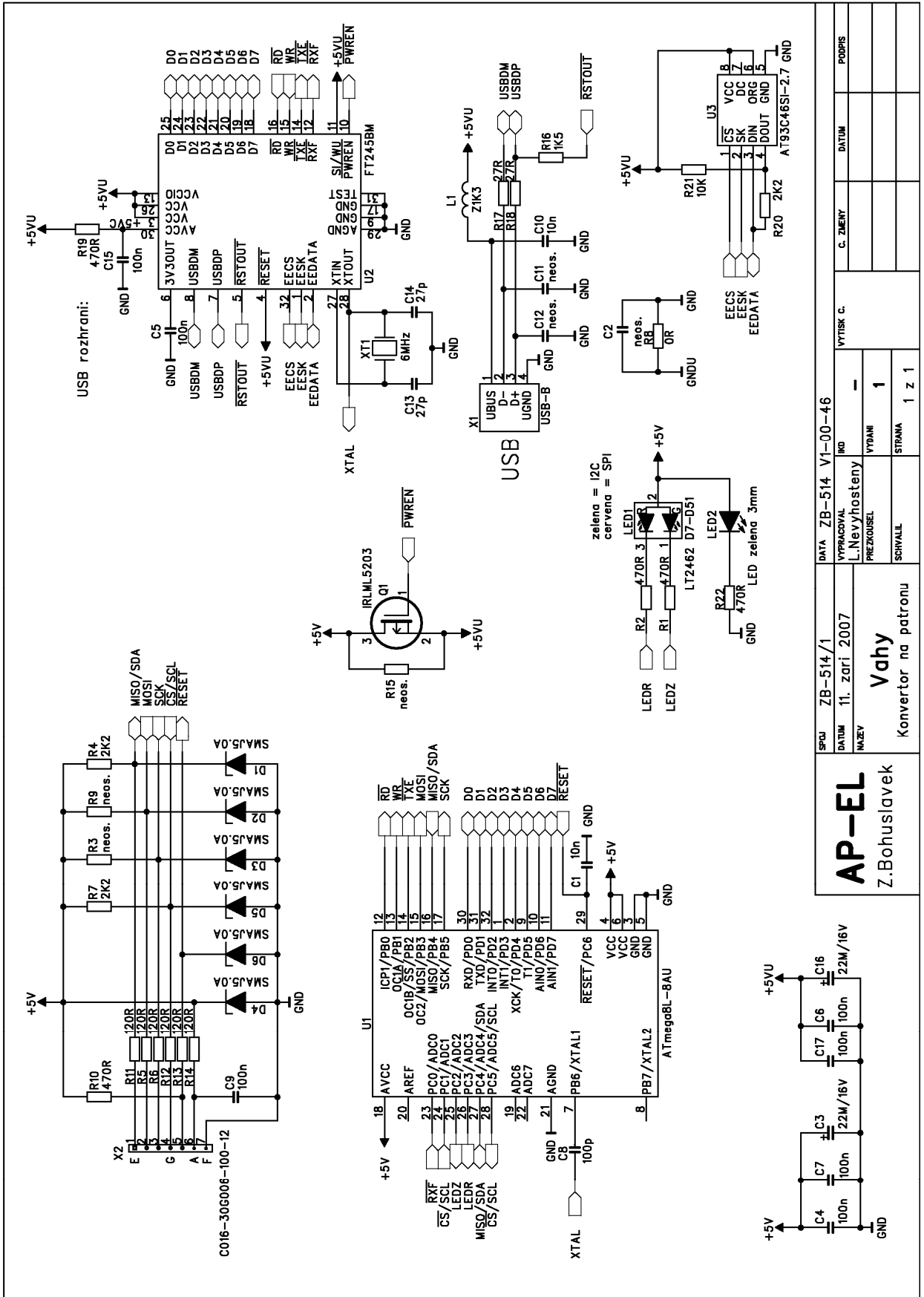
Tabulka 1 - Verze USB	5
Tabulka 2 - Zapojení kabelu USB (5)	8
Tabulka 3 - Typy PID překlad z (5)	12
Tabulka 4 - Srovnání USB 2.0 a 3.0	15

Příloha 1 – Diagram činnosti AVR309:



Príloha 2 – Schéma zapojení čtecího zařízení:

ZB-514 Konvertor na patronu V1-00-46.sch-1 - Tue Sep 11 15:17:34 2007



AP-EL Z.Bohuslavcek		SPIC ZB-514/1		DATA ZB-514 V1-00-46	
DATEM MAZEV	11. zari 2007	VYPRACOVAL L.Nevytosteny	PREZKOUSEL	VYTIISK C.	DATEM
Vahy Konvertor na patronu			VYDANI	1	PODPIS
			STRANA	1 z 1	