

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

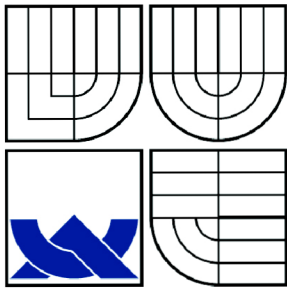
ŘÍZENÍ PERISTALTICKÉ PUMPY PŘES ROZHRANÍ USB

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

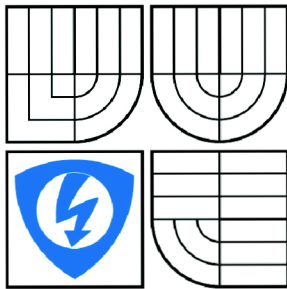
JAROSLAV BALOGH

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

# ŘÍZENÍ PERISTALTICKÉ PUMPY PŘES ROZHRANÍ USB

PERISTALTIC PUMP CONTROL VIA USB INTERFACE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

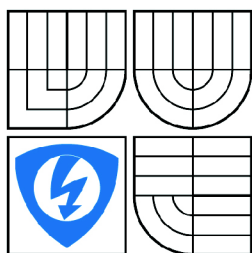
JAROSLAV BALOGH

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR ČECH

BRNO 2009



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Bakalářská práce

bakalářský studijní obor  
Elektronika a sdělovací technika

**Student:** Jaroslav Balogh

**ID:** 78472

**Ročník:** 3

**Akademický rok:** 2008/2009

## NÁZEV TÉMATU:

### Řízení peristaltické pumpy přes rozhraní USB

## POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s možnostmi řízení peristaltické pumpy Rainin Dynamax RP-1. Nastudujte rozhraní USB. Realizujte hardwarové propojení peristaltické pumpy Rainin Dynamax RP-1 s PC. Vytvořte řídicí software v libovolném programovacím prostředí, který bude pracovat na platformě Windows XP.

## DOPORUČENÁ LITERATURA:

[1] Dynamax Model RP-1 Peristaltic Pump User Guide. Emeryville: Rainin Instrument Co., 1996

[2] ŘEHÁK, J. USB - Universal Serial Bus - Popis rozhraní [online]. Praha: HW server s.r.o., 2002 - [cit. 7.5.2008]. Dostupné na [www.hw.cz](http://www.hw.cz).

**Termín zadání:** 9.2.2009

**Termín odevzdání:** 5.6.2009

**Vedoucí práce:** Ing. Petr Čech

**prof. Dr. Ing. Zbyněk Raida**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

## **Anotace:**

Účelem této bakalářské práce je realizovat propojení peristaltické pumpy Rainin Dynamax RP-1 a PC přes rozhraní USB. Dle zjištěných informací byl s pomocí mikrokontroléru FT232R realizován převodník mezi rozhraním USB a RS422, kterým pumpa disponuje. Následně byl vytvořen obslužný program pro vzdálené ovládání rumpy RP-1. Její provoz je tak možné do značné míry automatizovat, popřípadě řídit ze vzdáleného pracoviště.

Práce vznikla v návaznosti na semestrální projekt a na jejím základě je možné vytvořit další aplikace konvertoru USB-RS422, nebo vylepšit tu stávající. Ústav Biomedicínského inženýrství při FEKT, VUT Brno, totiž bude vzdálený přístup k pumpě využívat při analýze DNA.

## **Klíčová slova:**

Peristaltická pumpa, řízení, RS-422, USB, převodník, FT232R, C++, přenos dat.



**Abstract:**

The purpose of this bachelor's thesis is to practical realization of connection between peristaltic pump Rainin Dynamax RP-1 and PC, using USB interface. Integrated circuit device FT232R has been used in USB-RS422 converter, due to modified RS422 interface, which the pump RP-1 uses. There was also developed software for pump's remote controll. RP-1's functions can be driven from remote side or eventually automatized with this software.

This work is based on semestral thesis and in future can be used in another applications with RS422 or with RP-1 pump. Department of Biomedical Engineering, attached to FEEC, Brno University of Technology, will use this remote controll in analysis of DNA.

**Keywords:**

Peristaltic pump, control, interface RS-422, USB interface, data convertor, FT232R, C++, data transmission.

## **Bibliografická citace mé práce:**

BALOGH, J. Řízení peristaltické pumpy přes rozhraní USB: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 58 s.

## **Prohlášení**

Prohlašuji, že svou bakalářskou práci na téma Řízení peristaltické pumpy přes rozhraní USB jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 5. června 2009

.....

podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing. Petru Čechovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 5. června 2009

.....

podpis autora

# Obsah

<b>1 ÚVOD.....</b>	<b>7</b>
<b>2 PERISTALTICKÁ PUMPA RAININ DYNAMAX RP-1 .....</b>	<b>8</b>
2.1 Popis peristaltické pumpy Rainin Dynamax RP-1 .....	8
2.2 Ovládací panel a zadní panel preistaltické pumpy RP-1 .....	10
2.3 Komunikace pumpy RP-1 pomocí RS-422 .....	11
2.3 Externí terminál pumpy RP-1.....	14
<b>3 RS-232 A RS-442.....</b>	<b>15</b>
3.1 Stručný popis rozhraní RS-232.....	15
3.2 Parametry datového přenosu RS-232.....	15
3.2.1 Parita a stop bit.....	15
3.2.2 Asynchronní a synchronní přenos dat .....	16
3.3 Délka vedení s RS-232.....	16
3.4 RS-422.....	17
<b>4 USB.....</b>	<b>18</b>
4.1 Historie a základní popis USB.....	18
4.2 Plug and play .....	19
4.3 Parametry USB .....	20
4.4 Napájení .....	21
4.5 Přenos dat .....	21
<b>5 MOŽNOSTI OVLÁDÁNÍ PUMPY RP-1 POMOCÍ PC.....</b>	<b>23</b>
5.1 Využití integrovaných obvodů firmy FTDI při ovládaní přes RS-422.....	23
5.2 Využití mikrokontroleru při ovládaní přes externí terminál .....	24
5.3 Obslužný program.....	24
<b>6 OBVOD FT232R.....</b>	<b>25</b>
6.1 Charakteristika obvodu FT232R.....	25
6.2 Blokové schéma obvodu FT232R.....	26
6.3 Zapojení obvodu FT232R .....	27

<b>7 PŘEVODNÍK USB – RS422</b> .....	<b>30</b>
7.1 Návrh a popis převodníku USB – RS422 ve vývojovém prostředí Eagle .....	30
7.2 Modifikace výstupu pro pumpu RP-1 .....	34
<b>8 E<sup>2</sup>PROM</b> .....	<b>35</b>
8.1 Paměť E <sup>2</sup> PROM.....	35
8.2 Programování E <sup>2</sup> PROM .....	35
<b>9 PROGRAMOVÉ ŘEŠENÍ</b> .....	<b>36</b>
9.1 Ovladač FTDI .....	36
9.2 Klasické funkce ovladače D2XX.....	37
9.3 Zdrojový kód ovládacího programu pumpy RP-1 .....	39
9.4 Program Rainin Dynamax RP-1 – control .....	42
<b>10 ZÁVĚR</b> .....	<b>44</b>
<b>SEZNAM POUŽITÉ LITERATURY</b> .....	<b>45</b>
<b>SEZNAM OBRÁZKŮ</b> .....	<b>46</b>
<b>SEZNAM TABULEK</b> .....	<b>47</b>
<b>SEZNAM PŘÍLOH</b> .....	<b>48</b>
<b>SEZNAM PŘÍLOH NA CD</b> .....	<b>49</b>
<b>PŘÍLOHY</b> .....	<b>51</b>
Příloha 1: Zdrojový kód programu Rainin Dynamax RP-1 - control.....	52
Příloha 2: Návod k obsluze programu Rainin Dynamax RP-1 - control .....	58

# 1 Úvod

Cílem mé bakalářské práce je prakticky realizovat propojení peristaltické pumpy Rainin Dynamax RP-1 s PC přes rozhraní USB. Práce navazuje na výsledky semestrálního projektu, v kterém se byly prozkoumány možnosti ovládání pumpy RP-1. Byla zvolena realizace řízení pumpy s využitím jejího sériového rozhraní RS422 a převodníku USB - RS422. Jako převodník byl vybrán obvod FT232RL, který plně podporuje obě rozhraní - USB a UART. Ovládací program pumpy, vytvořený v prostředí C++ Builder, poskytne vzdálené obsluze její snadné a plnohodnotné ovládání.

Ústav biomedicínského inženýrství (spadající pod FEKT VUT Brno), který je vlastníkem RP-1, bude využívat vzdálený přístup k této pumpě při analýze DNA. Výsledkem bude efektivnější (možná i automatické) využití pumpy v celém procesu analýzy.

# 2 Peristaltická pumpa Rainin Dynamax RP-1

## 2.1 Popis peristaltické pumpy Rainin Dynamax RP-1

Dynamax RP-1 je peristaltická pumpa speciálně přizpůsobená pro zajištění toku kapaliny. Otočná hlava má deset válečků z nerezavějící oceli. Při otáčení hlavy pumpy pět válečků tlačí trubičku, jednu nebo více, proti kompresnímu rameni, což má za následek tok kapaliny způsobený změnou tlaku uvnitř trubičky. Pět válečků hlavy, které jsou konstantně v kontaktu s trubičkou, zabezpečuje malou pulsaci, hladký tok a vysoký tlak. Model RP-1 je proto ideální k dávkování kritických (velmi malých) množství kapaliny, má možnost gradientní změny průtoku a je také vhodná u různých aplikací průtokových analyzátorů. Obrázek o objemu kapaliny přenesené pumpou si je možno udělat při informaci, že nejmenší rychlost je 0,01 otáčky za minutu. Při použití PVC trubičky o průměru 0,25 mm (nejušší možná) a maximální rychlosti 48 otáček za minutu je dodáno 0,33 mL tekutiny za minutu.

Pumpa RP-1 je poháněna krokovým motorem s velkým kroutivým momentem. Tento motor je ovládán mikrokontrolerem se zpětnou vazbou. Zpětná vazba zabezpečuje větší stabilitu rychlosti i při malých objemech kapaliny. Požadavky pro přesnou práci jsou pracovní prostředí s teplotou mezi 0<sup>0</sup>C až 40<sup>0</sup>C a také tlak prostředí nesmí přesáhnout pět atmosfér. Rychlost otáček je zobrazena na LCD displeji (viz obrázek 2.2) v otáčkách za minutu (rpm – revolutions per minute) a je ovládána tlačídky na předním panelu, stejně jako směr průtoku.

Protože je mnohdy nutné dávkovat více než jednu kapalinu, bylo vyvinuto několik verzí hlavice pumpy: jednonábová a tři vícenábové (2, 4 a 8-nábové). Různé průtoky je možno nastavit použitím trubiček s rozdílným průměrem a to v každém náboji. Tlak, který vyvíjí náboj na trubičku je možno nastavovat individuálně, což je vhodné pro jemné vyladění průtoku a také pro zvýšení životnosti trubiček (ty se méně opotřebují). Trubičky jsou běžně z PVC, silikonu nebo vitonu a jsou dodávány přímo od výrobce. [1]

Pumpa, kterou vlastní UBMI při FEKT VUT Brno, má hlavu ve verzi se dvěma uchyceními pro hadičky. Na obrázku 2.1 je hlava ve verzi se dvěma uchyceními pro hadičky a s deseti ocelovými válečky dobře vidět. Všimnout si můžeme také různé nastavení přítlaku ramen na hadičky a přední část těla pumpy s LCD displejem a ovládacími prvky.



Obrázek 2.1: Peristaltická pumpa Rainin Dynamax RP-1. [fografie z archivu autora]



## 2.2 Ovládací panel a zadní panel preinstaltické pumpy RP-1

Na obrázku 2.2 můžeme vidět rozmístění prvků na ovládacím panelu. LCD displej zobrazuje rychlost motoru v otáčkách za minutu. Pokud je pumpa v provozu znaménko + respektive – indikuje rotaci hlavy ve směru (+) nebo protisměru (-) hodinových ručiček.

Dále je zde šest tlačidel, které slouží na ovládání chodu pumpy a jsou rozděleny do dvou sekcí: rychlost (speed) a tok (flow). Část pro ovládání rychlosti obsahuje 3 tlačítka a to „prime“, „+“ a „-“. Část, která ovládá tok má také tři: stop a dvě pro ovládání směru.

Tlačítko **prime** spouští vysokorychlostní mód. Po stisknutí se rychlost hlavy pumpy (při vypnuté pumpě tlačítko nereaguje) zvýší na 48 otáček za minutu, což je maximum. Maximální průtok trvá do doby, než je stlačeno tlačítko prime nebo jiné startovací tlačítko (+/-). Poté se pumpa vrátí do předchozí rychlosti.

Tlačítko **mínus** (-) snižuje rychlost motoru. Když pumpa běží, zmáčknutí a držení tohoto tlačítka plynule snižuje rychlost motoru. Nastavení rychlosti lze provést i ve stavu nečinnosti pumpy a nastavenou rychlostí se bude hlava otáčet po spuštění pumpy. Nejmenší krok, při nastavování rychlosti je 0.01 otáčky za minutu.

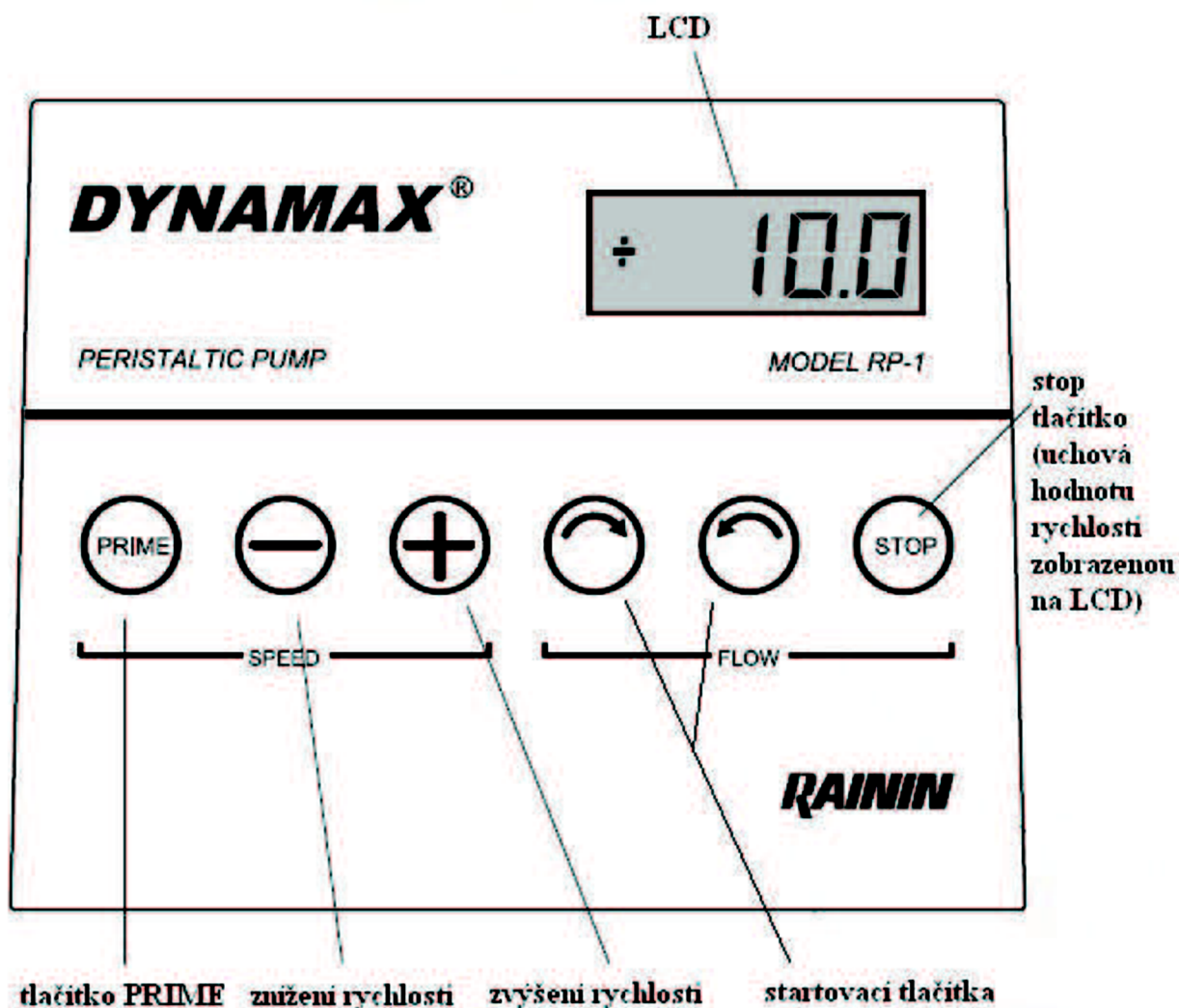
Tlačítko **plus** (+) zvyšuje počet otáček hlavy pumpy RP-1 za minutu a jeho obsluha je stejná jako u mínus.

Při stlačení tlačítka **start (po směru hodinových ručiček)** se začne otáčet hlava pumpy v daném směru a rychlostí, jaká je na LCD displeji. Stlačení tohoto tlačítka, když se hlava otáčí v maximální rychlosti (stlačení tlačítka prime) má za následek její snížení na předešlou hodnotu. Stlačení tohoto tlačítka v momentě, kdy se hlava točí proti směru hodinových ručiček způsobí zastavení pumpy a okamžité otáčení po směru a ve stejné rychlosti.

Tlačítko **start (proti směru hodinových ručiček)** má stejnou funkci s opačným směrem, než druhé start tlačítko.

Tlačítko **Stop** zastaví pumpu a uloží rychlost, která je aktuálně zobrazena na LCD. Tato hodnota je použitá při dalším spuštění pumpy některým ze startovacích tlačidel.

Na zadním panelu (obrázek 2.4) je možné vidět polohu portu RS-422 a také terminálového konektoru. Možnost ovládání pumpy těmito dvěma způsoby je popsána v dalších (pod)kapitolách.



Obrázek 2.2: Ovládací panel peristaltické pumpy Rainin Dynamax RP-1. [1]

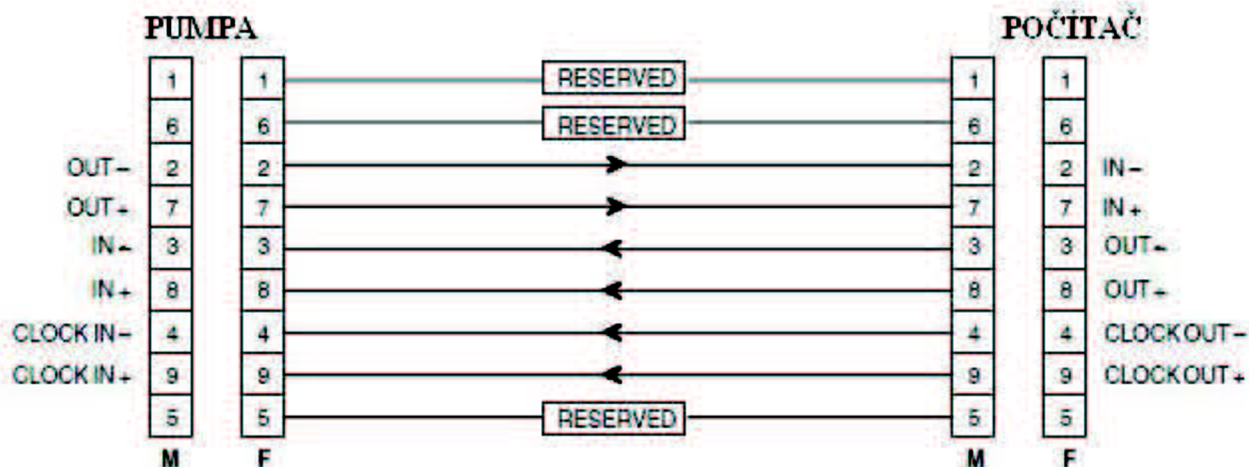
### 2.3 Komunikace pumpy RP-1 pomocí RS-422

Komunikace peristaltické pumpy Rainin RP-1 je možná z kteréhokoli PC se sériovým portem RS-422. Jednotlivé signálové linky RS-422 mezi PC a pumpou RP-1 mají své přesné využití, jak je vidět na obrázku 2.3. Tři linky jsou rezervovány a dalších šest je po párech (úrovně +/-) přiděleno pro signály IN, OUT a hodinový signál. Samozřejmě je možné ji k PC připojit pomocí adaptéru, který bude obsahovat převodník signálu z protokolu RS-422 na jiný, třeba USB nebo RS-232. Tyto protokoly (RS-422) jsou popsány v 3. kapitole.

RS-422 má obousměrný komunikační kanál, který umožňuje PC komunikovat až s 64 podřízenými zařízeními, v našem případě pumpy, s unikátními ID (označené jednotka 0 až jednotka 63). PC komunikuje s konkrétní pumpou v čase, když vyšle unikátní číslo (ID) pumpy.

Pumpa, která rozezná svoje ID se poté napojí na komunikační kanál a všechny ostatní jednotky jsou odpojeny.

Dátová komunikace je asynchronní, osmibitová, parita je sudá. Komunikace se skládá ze start-bitu, osmi datových bitů, jednoho paritního bitu a stop-bitu. Pumpa RP-1 kontroluje sudou paritu a používá ji i při odpovědi. Nastane-li chyba parity (parity error) je pumpa odpojena od RS-422. Opraveno to může být opětovným navázáním spojení a následného znovuodeslání zprávy.



Obrázek 2.3: Signálové linky portu pumpy RP-1.[1]

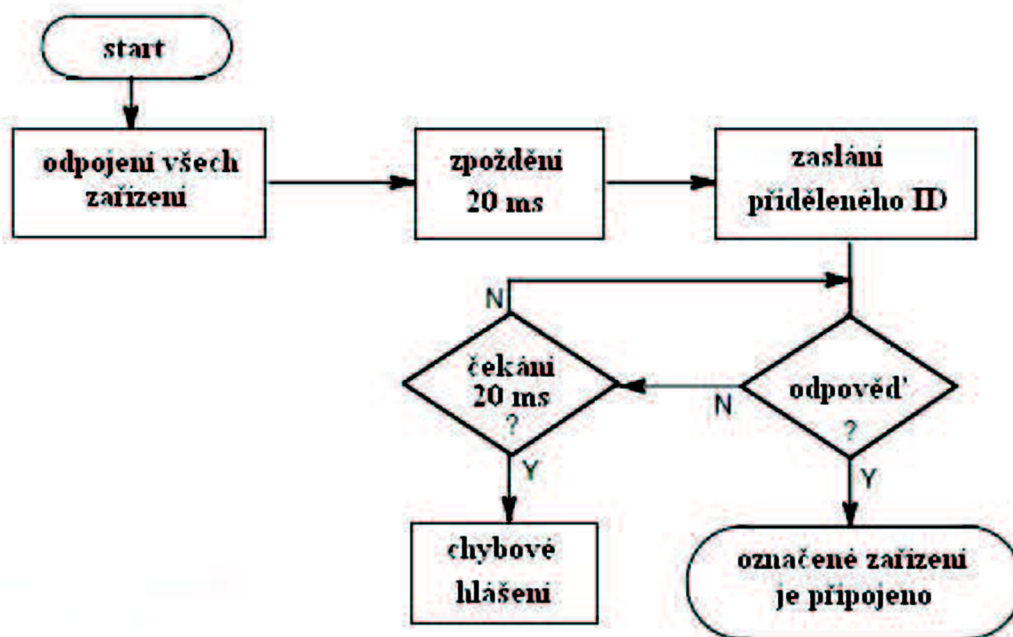
Podporované přenosové rychlosti jsou 600, 1200, 4800, 9600 a 19200 baud. Časovač běží na 16X přenosové rychlosti. Pumpa RP-1 pracuje s rychlostí 600, která pochází z vestavěného časovače. Není-li připojen časovací signal, bude pumpa komunikovat na 19200 baud. [1]

Před navázáním komunikace ke konkrétní pumpě musí nejdříve PC zaslat odhlašovací kód (disconnect code). Následně PC čeká 20 ms aby umožnil kterémukoli aktivnímu zařízení vypnout ovladače. Poté vysílá PC identifikační ID.

Pumpa, která rozená své ID připojí svůj vysílač na komunikační kanál a vyšle odevu (své ID). Ostatní zařízení toto ID nepoznají a odpojí se. Aktivní pumpa vysílá odezvu (ID) do 20 ms, poté co tyto data přijme, aby i PC obdrželo odezvu a nepokládalo zařízení za nedostupné. Pumpa zůstává aktivní do doby, než obdrží příkaz k odpojení (disconnect code), nebo ID jiné pumpy či zařízení. Celý proces je znázorněn na obrázku 2.5.



Obrázek 2.4: Zadní strana těla pumpy RP-1. [fografie z archivu autora]



Obrázek 2.5: Průběh navázání komunikace s pumpou RP-1. [1]

### 2.3 Externí terminál pumpy RP-1.

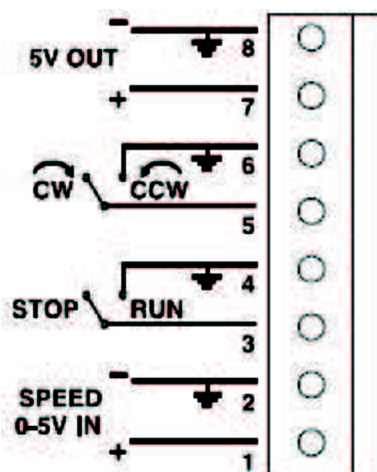
Pumpa RP-1 je vybavena také svorkovnicí, výrobcem označenou jako externí terminál (obrázek 2.4). V podstatě se jedná o vyvedené kontakty ovládacích tlačidel. Pomocí spojení, respektive rozpojení, těchto svorek (pinů) je možno prakticky ovládat pumpu RP-1. Konkrétně se zde nachází 8 pinů. Jejich rozmístění je na obrázku 2.6.

První pár (pin 1 a pin 2) podle připojeného napětí (0-5V) nastavuje rychlost otáčení. Pin 2 je uzeměn.

Druhý pár (pin 3 a pin 4) má funkci tlačítka start/stop. Při spojení těchto kontaktů se pumpa (otáčení hlavy) spustí, při rozpojení se zastaví.

Třetí pár (pin 5 a pin 6) ovládá směr otáčení. Při propojení těchto pinů se hlava otáčí proti směru hodinových ručiček, při rozpojení v opačném směru.

Na poslední pár svorek (pin 7 a pin 8) je vyvedeno napětí 5V a maximální odběr činí 70 mA. Uzeměn je pin 8. Tímto napětím je možno napájet externí zařízení a přímo se nabízí využití k ovládání rychlosti pomocí pinů 1 a 2.



Obrázek 2.6: Uspořádání pinů na svorkovnici (etrení terminál). [1]

## 3 RS-232 a RS-442

### 3.1 Stručný popis rozhraní RS-232

RS-232, na PC také označován jako sériový port, je rozhraní pro přenos informací vytvořené původně pro komunikaci dvou zařízení do vzdálenosti 20 m. Pro větší odolnost proti rušení je informace po propojovacích vodičích přenášena větším napětím, než je standardních 5 V. Přenos informací probíhá asynchronně, pomocí pevně nastavené přenosové rychlosti a synchronizace sestupnou hranou startovacího impulsu. RS-232 používá standardně dva druhy konektorů: 25 respektive 9-pionvý konektor - Cannon 25/Canon 9 (obr. 4.1).

RS-232 používá dvě napěťové úrovně logickou 1 a 0. Logická 1 je někdy označována jako marking state nebo také klidový stav, logická 0 se přezdívá space state. Logická 1 je indikována zápornou úrovní, zatímco logická 0 je přenášena kladnou úrovní výstupních vodičů. Nejběžněji se pro generování napětí používá napěťový zdvojovač z 6 V a invertor. Logické úrovně jsou potom přenášeny napětím +12 V pro logickou 0 a -12 V pro logickou 1.

### 3.2 Parametry datového přenosu RS-232

#### 3.2.1 Parita a stop bit

Parita je nejjednodušší způsob jak bez nároků na výpočetní výkon zabezpečit přenos dat. Ve vysílacím zařízení se sečte počet jedničkových bitů a doplní se paritním bitem tak, aby byla zachována předem dohodnutá podmínka sudého nebo lichého počtu jedničkových bitů.

- sudá parita – počet jedničkových bitů + paritní bit = sudé číslo
- lichá parita – počet jedničkových bitů + paritní bit = liché číslo
- nulová parita (space parity) – paritní bit je vždy v log. 0. Používá se například při komunikaci mezi 7-bitovým a 8-bitovým zařízením, kdy je paritní bit (poslední bit v byte) nahrazen trvalou log. 0. Tím je zachována kompatibilita s 8-bitovým přenosem.
- znaková parita (mark parity) - Paritní bit je nastaven tvrdě na log. 1. Při kompenzaci 7-bitového provozu je třeba tento bit na přijímací straně nulovat, jinak není kompatibilní s ASCII.
- stop bit – Definuje ukončení rámce. Zároveň zajišťuje určitou prodlevu pro přijímač. Právě v době příjmu STOP bitu většina zařízení zpracovává přijatý BYTE.
- zdvojený stop bit – Používá se u pomalejších zařízení pro dobř zpracování přijatého znaku. Jedná se o standard na 110 Bd .



### 3.2.2 Asynchronní a synchronní přenos dat

Synchronní přenos informací znamená, že na nějakém vodiči nebo vodičích se nastaví určitá úroveň, která přenáší informaci a validita informace se potvrdí impulzem, nebo změnou úrovně synchronizačního signálu. Synchronizačním signálem se tedy informace kvantují.

Základní vlastnosti synchronního přenosu :

- Výhodné pro velké objemy dat, přenášené po více vodičích.
- Nutno jednoznačně určit, kdo vysílá synchronizační impulzy.
- Možno použít spojitě proměnnou rychlost přenosu, například podle poměru chybovosti.
- Nutnost synchronizačního vodiče „navíc“ – v podstatě „nepřenáší žádnou informaci“.
- Na straně přijmacího zařízení nepotřebuje nijak složitou elektroniku.

Asynchronní přenos dat přenáší data v určitých sekvencích. Data jsou přenášena přesně danou rychlostí a uvozena startovací sekvencí, na kterou se synchronizují všechna přijímací zařízení. Všechny strany obsahují vlastní přesný oscilátor, díky kterému odečítají data v přesně definovaných intervalech. Po ukončení sekvence je další příjem opět synchronizován startovní sekvencí.

Základní vlastnosti asynchronního přenosu :

- Nevýhodné pro velké objemy dat, ale vhodné pro dlouhá vedení, na nichž by synchronizační vodič činil nezanedbatelné finanční náklady.
- Lze použít pro komunikaci mezi mnoha zařízeními.
- Nutno definovat jednoznačně přenosové rychlosti, změnu rychlosti je třeba ošetřit softwarovou sekvencí, která přiměje počítač změnit hardwarově přenosovou rychlost.
- Celkem složitá a drahá elektronika, nutno použít krystalové oscilátory.
- Až o 20% menší přenosová rychlost užitečných dat při stejné rychlosti komunikace, vzhledem k nutnosti startovacích a paritních bitů.

### 3.3 Délka vedení s RS-232

Standard RS-232 uvádí jako maximální možnou délku vodičů 15 metrů, nebo délku vodiče o kapacitě 2500 pF. To znamená, že při použití kvalitních vodičů lze dodržet standard a při zachování jmenovité kapacity prodloužit vzdálenost až na cca 50 metrů.

Kabel lze prodlužovat při snížení přenosové rychlosti, protože potom bude přenos odolnější vůči velké kapacitě vedení. Uvedené parametry počítají s přenosovou rychlostí 19200 Bd. [7]

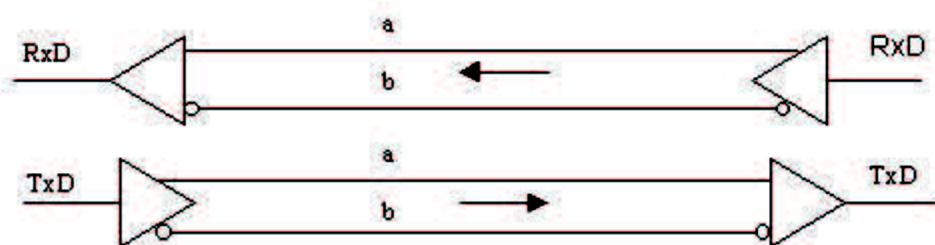
### 3.4 RS-422

Vzhledem k malé odolnosti rozhraní RS-232 vůči rušení bylo vyvinuto několik způsobů jak je možno zvětšit dosah. Prtří mezi ně tzv. proudová smyčka, rozhraní RS-422 a rozhraní RS-485. Tyto linky mohou být vedeny až na vzdálenost 1600m (vodiče s kapacitou do 65pF/m) a lze je větvit. Právě pro možnost obsluhy ze vzdáleného pracoviště je peristaltická pumpa Rainin RP-1 vybavena rozhraním RS-422.

Linka RS-422 používá jeden pár vodičů pro signál RxD a druhý pro signál TxD (viz obrázek 3.1). Z toho vyplývá, že použijeme-li linku RS-422 k prodloužení přenosové vzdálenosti místo “třídrátové” RS-232 (RxD, TxD, GND), nic se nemusí na způsobu komunikace měnit a není tedy třeba ani zásah do software.

Každý ze signálů linky je přenášen po dvojici vodičů, nejlépe v provedení twistový pár. Vodiče označované **a** a **b** jsou vysílačem buzeny v protifázi a přijímač vyhodnocuje jejich napěťový rozdíl. Tímto principem se odstraní součtové (aditivní) rušení.

Zatímco linka RS-232 pracuje s úrovněmi typicky  $-12\text{V}$  a  $+12\text{V}$ , úrovně linky RS-422 jsou menší, typický rozdíl mezi vodiči je  $2\text{V}$ . Aby přijímač mohl pracovat diferenciálně, nesmí být rozdíl mezi zemí vysílače a zemí přijímače větší než  $7\text{V}$ . V opačném případě se vstupy přijímače zahltí a dojde k přerušení komunikace. Proto je nezbytné používat linku RS-422 vždy s galvanickým oddělením, jinak se její výhody ztratí.



Obrázek 3.1: Provedení nevětvené linky RS-422. [7]



# 4 USB

## 4.1 Historie a základní popis USB

USB (z angl. Universal Serial Bus – univerzální sériová sběrnice) vzniklo za spolupráce firem Compaq, Hewlett-Packard, Intel, Lucent, NEC, Microsoft a Philips za účelem vytvoření levného univerzálního rozhraní pro externí zařízení s nižší průchodností dat. Účelem bylo sjednotit způsob připojování těchto periférií. Prvního skutečného rozšíření se USB dočkalo v roce 1998, tři roky po první specifikaci, v počítači Apple iMac. Tento počítač byl vybaven pouze (!) porty USB.

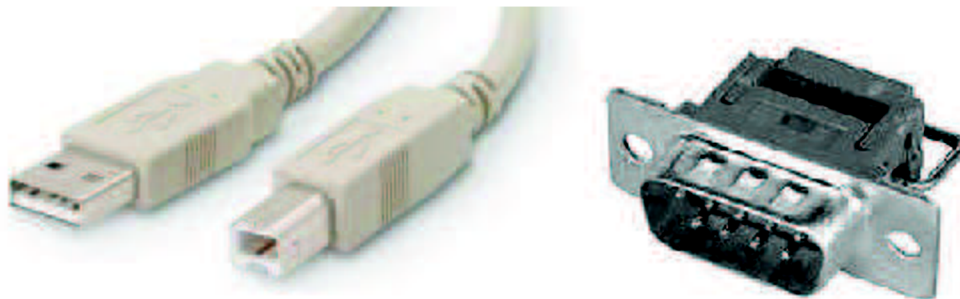
Ve verzi USB 1.1 existují pomalá (low-speed) zařízení s přenosovou rychlostí 1,5 Mb/s a rychlá zařízení (full-speed) s rychlostí 12 Mb/s. USB 1.1 však nebylo schopno konkurovat vysokorychlostním rozhraním např. FireWire (IEEE 1394) od firmy Apple (400 Mb/s; až 63 zařízení). Proto se v roce 1999 začalo uvažovat o druhé generaci USB, která by byla použitelná i pro náročnější zařízení (např.: digitální kamery). Tato nová verze, označovaná jako USB 2.0 (Hi-Speed) přišla v roce 2000 a nabídla maximální rychlost 480 Mbit/s, avšak zachovala zpětnou kompatibilitu s USB 1.1.

USB je v současné době zřejmě nejrozšířenější rozhraní pro připojení periférií ( třeba peristaltická pumpa, tiskárna, myš, klávesnice, joystick, fotoaparát, modem, čtečka paměťových karet a další) k PC, používané ve všech oblastech, od spotřební elektroniky připojitelné k PC až k specializovaným el. zařízením. Vyjma některých průmyslových odvětví USB téměř vytlačilo sériový port RS-232, paralelní port a jiné, třeba PS/2 nebo GamePort. USB rozhraní primárně používá dva typy konektorů. Plochý konektor „typ A“ (obr. 4.1) je dnes obsažen na prakticky každém novém PC v minimálně 2 konektorech (některé základní desky mají integrovanou rovnou USB hub, který obsahuje až 8 portů přímo v PC). Druhý konektor „typ B“ (obr. 4.1) je určen pro periferní zařízení, čímž je zároveň definován standard propojovacího kabelu. Popularitu dokazuje i snaha výrobců a uživatelů speciálních aplikací přecházet na konfiguraci, kde USB slouží jako primární rozhraní a také snaha těchto výrobců o diferenciaci trhu vývojem vlastních verzí konektorů a jiných úprav..

Mezi základní parametry rozhraní USB patří:

- Komunikační rychlost od 1,5 Mbit/s do 480Mbit/s
- Komunikační vzdálenost do 5m
- Možnost připojení více zařízení
- Rozhraní obsahuje 5V napájení

- Lze připojit až 127 zařízení pomocí jednoho typu konektoru.
- USB zajišťuje správné přidělení prostředků (IRQ, DMA, ...).



Obrázek 4.1: Konektory (zleva) USB typ „A“, USB typ „B“ a Cannon 9. [via internet]

## 4.2 Plug and play

Programové vybavení pro ovládání rozhraní USB plně odpovídá standardu plug and play (připojení zařízení za chodu - bez nutnosti restartování počítače a přístupnost tohoto zařízení v řádech sekund) a je již několik let součástí operačních systémů Windows (89,62% uživatelů, listopad 2008), Mac OS X (8,87%), Linux (0,83%) a jejich dalších alternativ. Při fyzickém připojení nového zařízení nejprve hub podle zdvižené datové linky pozná, že se objevilo nové zařízení. Pak proběhnou následující kroky:

1. Hub informuje hostitelský počítač (host) o tom, že bylo připojeno nové zařízení.
2. Hostitelský počítač se dotáže hubu, na který port bylo zařízení připojeno.
3. Hostitelský počítač nyní ví, na který port bylo zařízení připojeno. Vydá příkaz tento port zapnout a provést vynulování (reset) sběrnice.
4. Hub vyrobí nulovací signál (reset) o délce 10 ms. Uvolní pro zařízení napájecí proud 100 mA. Zařízení je nyní připraveno a odpovídá na implicitní (default) adrese.
5. Než zařízení USB obdrží svou vlastní adresu sběrnice, je možno se na ně obracet přes implicitní adresu 0. Hostitel si přečte první bajty deskriptoru zařízení, aby stanovil, jakou délku mohou mít datové pakety.
6. Hostitel přiřadí zařízení jeho adresu na sběrnici.
7. Hostitel si ze zařízení pod novou sběrniceovou adresou načte všechny konfigurační informace.
8. Hostitel přiřadí zařízení jednu z možných konfigurací. Zařízení nyní může odebírat tolik proudu, kolik je uvedeno v jeho deskriptoru zařízení. Tím je připraveno k použití..

Řídící dotazy jsou uloženy do endpointu 0. Po analýze se rozpozná druh dotazu, určité klíčové bajty datového paketu pak definují požadavek na deskriptor zařízení. Mikrokontrolér pak odpoví zápisem deskriptoru do výstupního endpointu 0, odkud si je hostitel přečte.

Po prvním přístupu obdrží zařízení definitivní sběrniceovou adresu, která se musí zapsat do SIE proto, aby mohly být přijímány následující datové pakety směřované na zařízení. [6]

Položka	Význam
VID (Vendor ID)	Číselný identifikátor výrobce (16b číslo přidělováno organizací USB)
PID (Product ID)	Číselný identifikátor výrobku (16b číslo určené výrobcem)
Manufacturer ID	Řetězec identifikující výrobce
Manufacturer	Řetězec popisující výrobce
Product	Řetězec popisující výrobek
Serial Number	Řetězec sériového čísla (umožňuje připojit několik stejných výrobků)
Počet konfigurace	Počet konfiguračních deskriptorů, které se například liší odběrem

Tabulka 4.1: *Popis (deskriptor) zařízení. [6]*

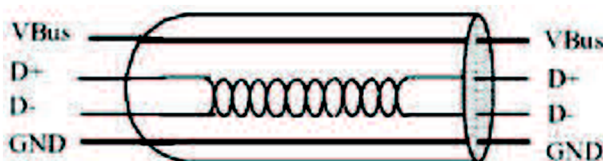
### 4.3 Parametry USB

Jsou definovány dvě verze fyzické vrstvy USB 1.1. Pro USB verze 2.0 byla doplněna nejrychlejší vrstva :

- High Speed - 480Mbps/s
- Full Speed - 12Mbps/s
- Low Speed - 1.5Mbps/s

Všechny verze propojení mohou být použity a provozovány současně pro připojení různých typů periférií k jednomu počítači. Uvedené verze se od sebe liší jak provedením kabelu, tak elektrickými parametry rozhraní připojeného zařízení.

Při rychlostech "Full Speed" a "High Speed" je log. 1 přenášena diferencially uvedením D+ přes 2.8V s připojeným 15Kohm odporem na GND a D- je pod 0.3V s 1.5Kohm odporem připojeným na 3.6V. Log. 0 je řešena opačně =D- přes 2.8V a D+ pod 0.3V se stejnými hodnotami odporů. Stínění USB kabelu musí být připojeno k pinu GND na straně "host". Žádné zařízení již nepřipojuje stínění k pinu GND. Uspořádání vodičů v kabelu USB je zobrazeno na obrázku 4.2. [6]



Obrázek 4.2: *Uspořádání vodičů v kabelu USB. [6]*

## 4.4 Napájení

Rozlišuje se mezi zařízeními s vlastním napájecím zdrojem (self-powered) a zařízeními, která jsou napájena přes sběrnici USB (bus-powered). V mnoha případech je možno volit oba způsoby. Zařízení pak má například konektor pro napájecí zdroj, který je možno volitelně propojit s externím napájecím zdrojem. Podle specifikace USB je proudový odběr ze sběrnice automaticky omezen. Je-li tudíž odebírán větší proud než přípustný, napájení by mělo být odpojeno. Připojeným zařízením USB zároveň poskytuje i stejnosměrné napájecí napětí 5 V. Připojené zařízení tak může po sběrnici odebírat proud až 100 mA, v případě potřeby může zařízení požádat o větší proud, maximálně však o 500 mA. U osobních počítačů občas bývají napájecí vodiče sběrnice vyvedeny přímo ze zdroje počítače a USB zařízení připojené k počítači tak může odebírat i mnohem vyšší proud. Tohoto triku zneužívají například některé externí USB pevné disky, jejichž odběr je vyšší než požadovaných 500 mA a které po připojení k jinému počítači nemusí fungovat.

HUB dodává pomocí napájecích pinů do USB zařízení napětí 4.75 V až 5.25 V. Maximální pokles napětí ze HUBu je 0.35 V. [6]

- Každý HUB musí být schopen poslat konfigurační data na napětí 4.4V, ale jen "low-power" funkce musejí fungovat.
- HUB napájený po sběrnici : Odběr max 100 mA při zapnutí a 500 mA průběžně.
- HUB napájený ext. zdrojem : Odběr max 100 mA, musí dodávat 500 mA na každý port.
- Zařízení "Low power" : Odběr max 100 mA.
- Běžná zařízení : Odběr max 100 mA.
- "Spící" zařízení : Max 0.5 mA

## 4.5 Přenos dat

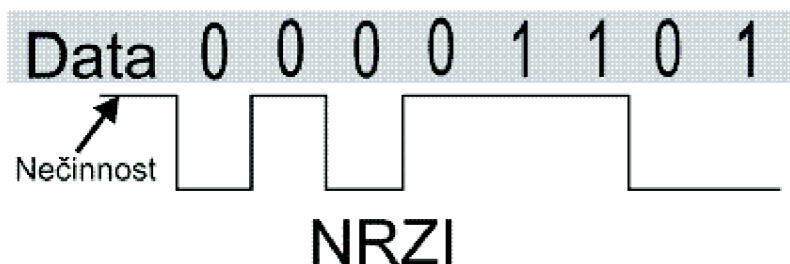
Pro veškerou komunikaci mezi počítačem a funkční jednotkou jsou k dispozici tři typy paketů. Každá výměna dat začíná tím, že počítač vyšle tzv. token packet obsahující popis typu a směru výměny dat, adresu USB zařízení a číslo koncové jednotky (endpoint number). Pak zařízení, které má vysílat data, vyšle datový paket nebo indikuje, že žádná data nejsou k dispozici. Příjemčí strana nakonec vyšle tzv. handshake packet, kterým informuje, zda přenos proběhl úspěšně.

Existují zde dva typy přenosového modelu: Prvním typem je tok dat (stream) využívající izochronní přenos dat v reálném čase. Tok dat nemá přesně definovanou strukturu. Druhým je zpráva (message), který využívá asynchronní přenos. Zpráva má přesnou strukturu :

- Řídící zpráva určená pro konfigurování poprvé aktivovaného zařízení;
- Zpráva obsahující větší objem dat (např. pro tiskárnu nebo plotter), jež je většinou segmentována do více částí;

- Zpráva s přerušením (obvykle několik bajtů), kterou spontánně vysílá zařízení, aby předalo zprávu o svém stavu (např. změna polohy myši).

Pro kódování je ve všech případech použit kód NRZI (not-return-to-zero recording). Zabezpečení přenosu zajišťuje CRC (Cyclic Redundancy Check).



Obrázek 4.3: Kód NRZI. [via internet]

# 5 Možnosti ovládání pumpy RP-1 pomocí PC

## 5.1 Využití integrovaných obvodů firmy FTDI při ovládání přes RS-422

Pomocí integrovaných obvodů firmy FTDI se lze skrz USB snadno připojit do širokého spektra aplikací. Z hlediska PC se pak přípravek s FTDI, který je připojen na USB, jeví jako standardní (ovšem velmi rychlý) sériový port. Z hlediska zařízení se jeví buď jako sériový port (typ '232) nebo jako osmibitová obousměrná sběrnice (typ '245).

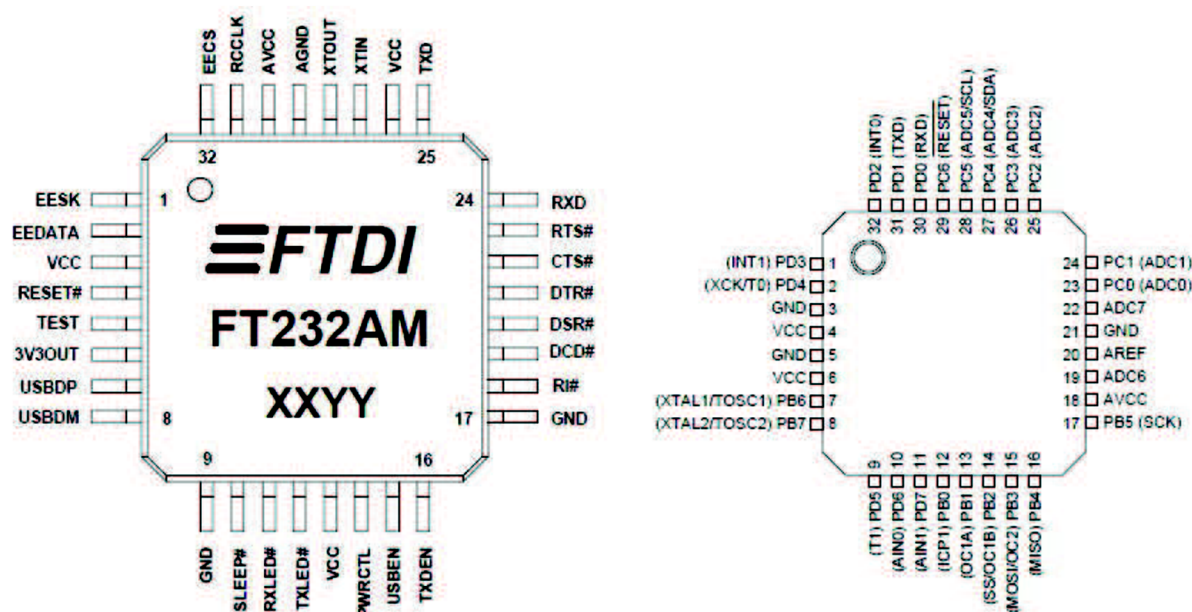
Není-li mezi RS-232 a RS-422 z pohledu způsobu komunikace žádný rozdíl, je možné použít integrovaný obvod FT8U232AM firmy FTDI (na obrázku 5.1 je schéma rozmístění pinů) jako konvertor USB - UART s přenosovou rychlostí 300 Bd až 920 kBd pro RS-232/RS-422. K dispozici je i plné hardwarové řízení přenosu - signály RTS, CTS, DTR, DSR, DCD a RI, a navíc signál TXDEN pro spolupráci s konvertory úrovní pro RS-485. V obvodu je zabudována dvouportová vyrovnávací paměť o velikosti 384 B ve směru od PC k aplikaci a 128 B ve směru k PC. [6]

Společnými vlastnostmi konvertorů firmy FTDI je podpora protokolu USB 1.1, možnost připojení externí EEPROM obsahující uživatelské sériové číslo nebo identifikační řetězec, možnost napájení 4,4 V až 5,25 V přímo z USB (zabudovaný 3,3 V regulátor), integrovaný násobič kmitočtu 6 MHz - 48 MHz pro časování USB operací. Proudová spotřeba je max. 50 mA při normálním provozu a max. 250 mA v režimu USB Suspend. Oba obvody se vyrábějí v kompaktním pouzdře MQFP (velikost 7×7 mm) s 32 vývody o rozteči 0.8 mm. Uživatel snadno přistupuje ke koncovému UARTu nebo FIFO portu prostřednictvím ovladačů VCP (Virtual COM Port) dodávaných pro platformy Windows 98/ME/NT4.0/2000/XP, Apple OS8/OS9 a Linux, jakoby obsluhoval standardní COM porty (např. pomocí Win API). K dispozici jsou také DLL knihovny s ovladači včetně příkladů pro Visual C++, Visual Basic, Borland C++ Builder a Delphi. Ovladače jsou volně dostupné na Internetu. Použití uvedených integrovaných obvodů tedy není zatíženo žádnými dalšími skrytými náklady (nákup návrhového systému, placení licenčních poplatků apod.).

Použití FT8U232AM tedy odstraňuje problémy při programování, hlavně složité nastavování registrů mikrokontrolerů, které nejsou přímo uspořádány pro komunikaci mezi USB a RS-422. K propojení PC a RP-1 by tedy stačilo doporučené zapojení FT8U232AM a obslužný program.

## 5.2 Využití mikrokontroleru při ovládní přes externí terminál

Další možností je naprogramovat běžný mikrokontroler, například ATmega8 (na obrázku 5.1 je schéma rozmístění pinů), aby podle vstupů z obslužného softwaru řídil peristaltickou pumpu RP-1 pomocí externího terminálu (podkapitola 2.3). Spojování a rozpojování kontaktů svorkovnice pinů by zabezpečoval tranzistor, nebo relé, změnu rychlosti potom využití funkce pulsně šířkové modulace. Napájení potřebné pro obsluhu terminálu je na jeho svorkovnici vyvedeno přímo z pumpy. Výhodou je fakt, že podobné aplikace se běžně prakticky programují ve freeware programech a také ceny těchto mikrokontrolerů a obslužných obvodů jsou v řádech desítek korun.



Obrázek 5.1: Schéma rozmístění pinů na FT8U232AM (vlevo) a ATmega8 (vpravo) v pouzdech typu MLF. [9] a [10]

## 5.3 Obslužný program

Obslužný program, použitelný na platformě MS Windows XP je možné naprogramovat v několika programovacích jazycích, například C++, Matlab nebo Labview. Tyto se krom jiného liší mírou uživatelského komfortu a také preferencí mezi textovým nebo grafickým programovacím jazykem.

Po spuštění by aplikace měla mít podobné funkce jako má ovládací panel pumpy RP-1 a je možné vytvořit jednoduché funkce, zabezpečující předprogramovatelná spuštění/vypínání pumpy a změny rychlosti průtoku tekutiny. Další možnosti jsou ve vytvoření přesných záznamů činnosti pumpy a další archivace dat.

# 6 Obvod FT232R

## 6.1 Charakteristika obvodu FT232R

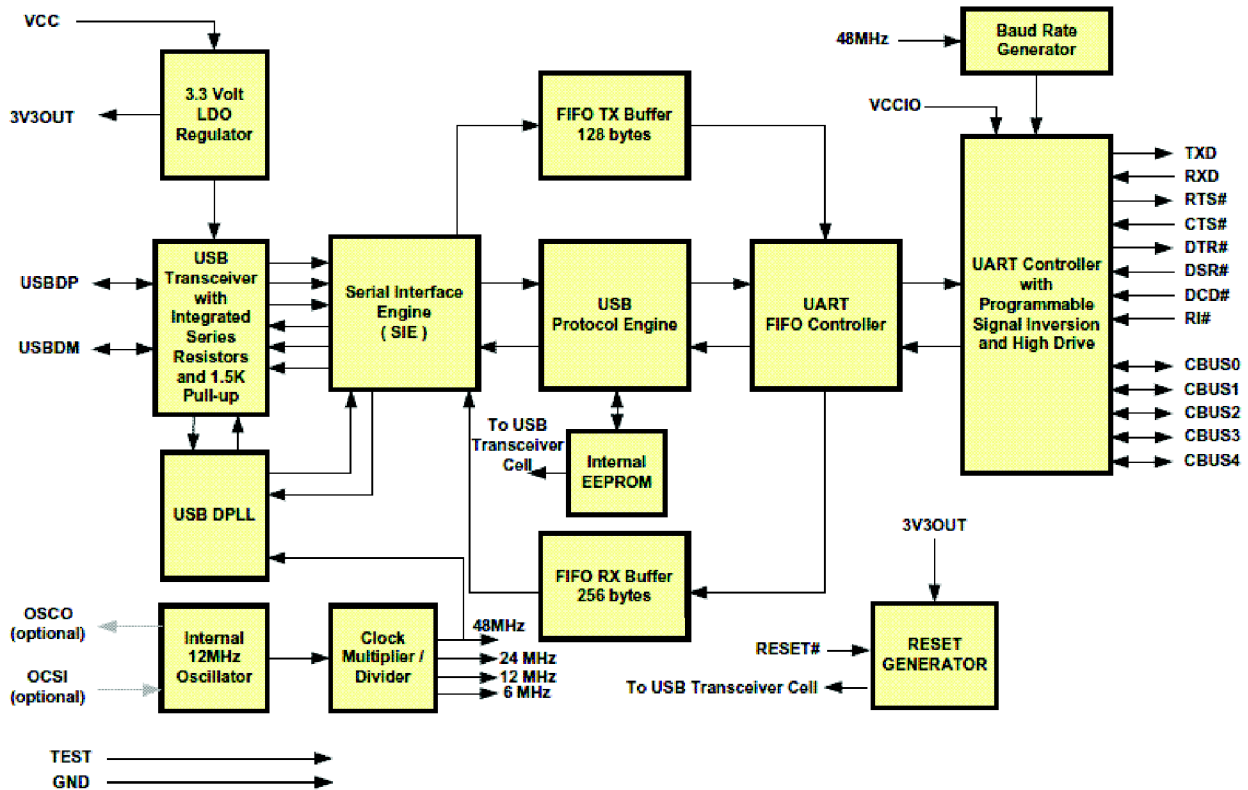
FT232R je jednočipový převodník mezi sběrnicemi USB a UART. Obvody FT232R jsou již čtvrtou generací, vylepšením verzí FT232AM a FT232BM, které přidávají nové funkce (konfigurovatelné řídicí signály) a integrují v sobě více součástek. Ve verzi FT232R (proti verzi FT232BM) je integrována například paměť E<sup>2</sup>PROM, pro uložení identifikace USB zařízení v systému (VID (Vendor ID) a PID (Product ID)), dále obvod obsahuje USB rezistory a integrovaný zdroj hodin (není nutné připojení krystalu). [4] a [2]

Hardwarové vlastnosti FT232R jsou:

- jednočipový převodník mezi sběrnicemi USB – UART
- plný handshake a rozhraní signálu modemu
- podpora 7/8 bitového přenosu, 1/2 stopbitů a parity (sudá, lichá, značená, mezerová, bez parity)
- přenosová rychlost nastavitelná v širokých mezích
- přijmací buffer hloubky 256 B, vysílací buffer hloubky 126 B (zajištění vysoké propustnosti dat)
- integrovaný konvertor úrovní UART a řídicích signálů pro 5 až 1,8 voltovou logiku
- integrovaný 3,3 voltový regulátor pro USB obvody
- integrovaný Power-on Reset
- kompatibilita se standardy USB 1.1 a USB 2.0



## 6.2 Blokové schéma obvodu FT232R



Obrázek 6.1: Zjednodušené blokové schéma obvodu FT232R. [4]

Popis vývodů FT232RL a jejich funkcí

- RXD, TXD – vstup a výstup UART
- DTR#, RTS#, RI#, DSR#, DCD#, CTS# - signály modemu UART
- VCC, GND, AGND – napájecí napětí (včetně analogové země pro oscilátor)
- VCCIO – napájecí napětí pro napěťový převodník (rozsah 1.8 až 5 V)
- V3VOUT – výstup vestavěného stabilizátoru 3,3 V
- USBDM, USBDP – vývody pro připojení na datové linky USB
- RESET# - resetovací vstup (obvod má vlastní reset, pokud není třeba provádět vnější reset, zapojuje se na VCCIO)
- TEST – určen pro testování funkce výrobcem (připájí se na zem)
- OSCI, OSCO – vstup a výstup oscilátoru (lze však využít vestavěný generátor hodin)
- CBUS0 až CBUS4 – konfigurovatelné vstupně/výstupní vývody řídicí sběrnice (pro význam signálů viz tabulka 6.1)

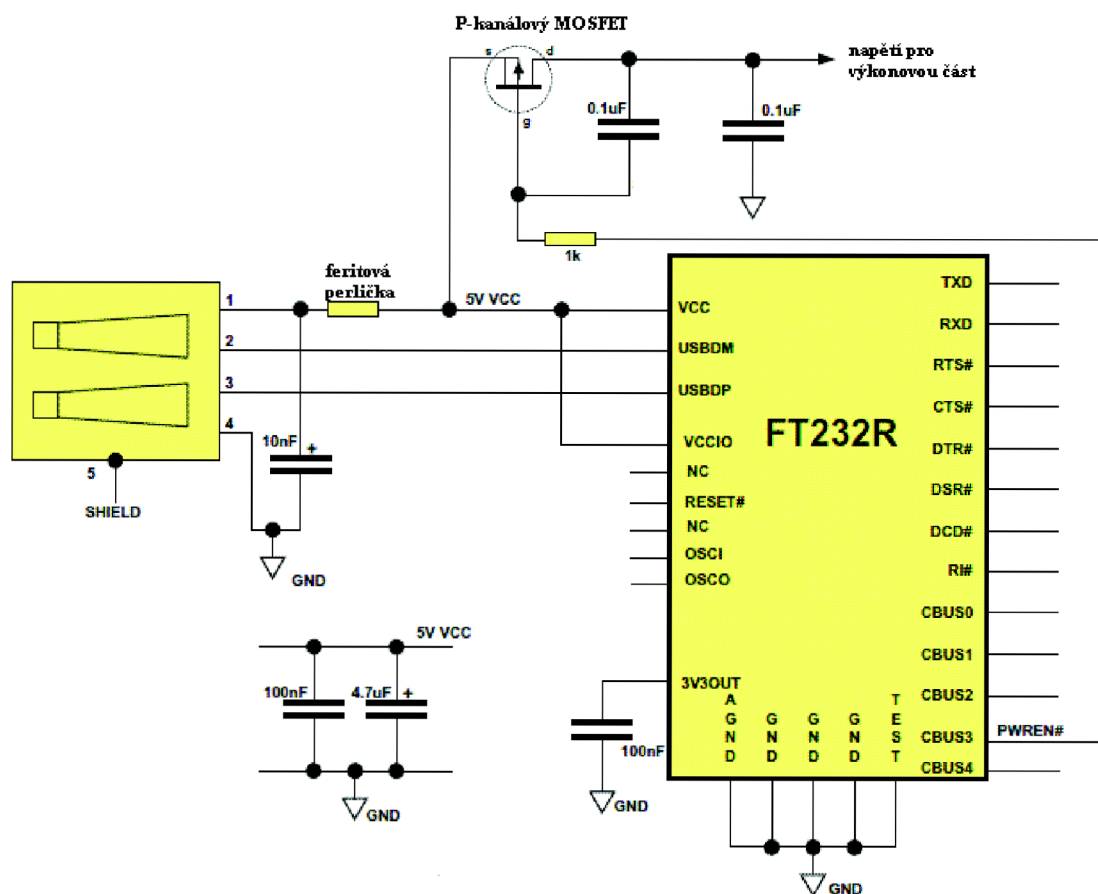
Signál	Dostupný pomocí vývodu	Popis
TXDEN	CBUS0 až CBUS4	Povolení vysílání dat pro RS485.
PVREN#	CBUS0 až CBUS4	Přechází do log. 0 po tom, co je zařízení konfigurováno přes USB. Je držení v log. 1 v průběhu režimu USB suspend. Tento signál se může používat pro řízení P-kanálového MOSFET (musí být přispůsoben logickým úrovním) spínače. Pro tento účel musí být povolen pull-down.
TXLED#	CBUS0 až CBUS4	Signál pro indikační LED odesílání dat.
RXLED#	CBUS0 až CBUS4	Signál pro indikační LED příjmu dat.
TX&RXLED#	CBUS0 až CBUS4	Signál pro indikační LED odesílání/příjmu dat (sdružení funkce do jednoho vývodu).
SLEEP#	CBUS0 až CBUS4	Přechází do log. 0 v průběhu režimu USB suspend. Obvykle se používá pro vypnutí vnějšího převodníku napět'ových úrovní z TTL na RS-232 C
CLK48	CBUS0 až CBUS4	Hodinový výstup 48 MHz.
CLK24	CBUS0 až CBUS4	Hodinový výstup 24 MHz.
CLK12	CBUS0 až CBUS4	Hodinový výstup 12 MHz.
CLK6	CBUS0 až CBUS4	Hodinový výstup 6 MHz.
CbitBangI/O	CBUS0 až CBUS3	V režimu bit bang lze všechny 4 vývody CBUS sběrnice používat jako obecné vstupně/výstupní (konfigurováním v interní E <sup>2</sup> PROM)
BitBangWRn	CBUS0 až CBUS3	Strobovací signál zápisu WR# pro synchronní a asynchronní režim bit bang (výstup).
BitBangRDn	CBUS0 až CBUS3	Strobovací signál čtení RD# pro synchronní a asynchronní režim bit bang (výstup).

Tabulka 6.1: *Dostupné signály sběrnice CBUS.* [3]

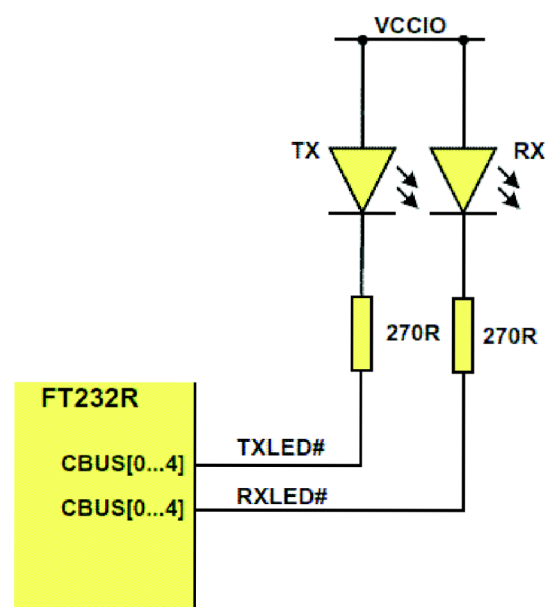
### 6.3 Zapojení obvodu FT232R

Je možné zapojit obvod FT232R v mnoha variantách, například zapojení s napájením aplikace z USB sběrnice, 2 varianty zapojení s napájením aplikace z vlastního zdroje, zapojení pro 3,3 V logiku, různá zapojení indikačních LED, zapojení pro režim BitBang (možnost přímého řízení všech 8 datových linek, datové výstupy FT232R lze poté (nastavení v E<sup>2</sup>PROM) chápat jako 8bitovou vstupně/výstupní sběrnici) nebo zapojení s možností řízení spotřeby vnějších obvodů. V této kapitole jsou vybraná zapojení obvodu s napájením z USB a řízením odběru, zapojení indikačních LED a zapojení UART rozhraní RS422.

Při zapojení s napájením aplikace z USB a řízením odběru externích obvodů (obrázek 6.2) je obvod FT232R napájen přímo z USB sběrnice a další obvody jsou připojeny k napájení až za P-kanálovým MOSFETem. Ten je řízen obvodem FT232R a zabezpečuje nepřekročení maximálního odběru z USB. Je ovšem nutné (v E<sup>2</sup>PROM) povolit volbu pull-down pro režim suspend, konfigurovat jeden z vývodů sběrnice CBUS jako PWREN# a použít jej pro ovládání spínacího tranzistoru a řízená logika musí mít vlastní resetovací obvod, který automaticky zajistí reset při náběhu/poklesu napětí.

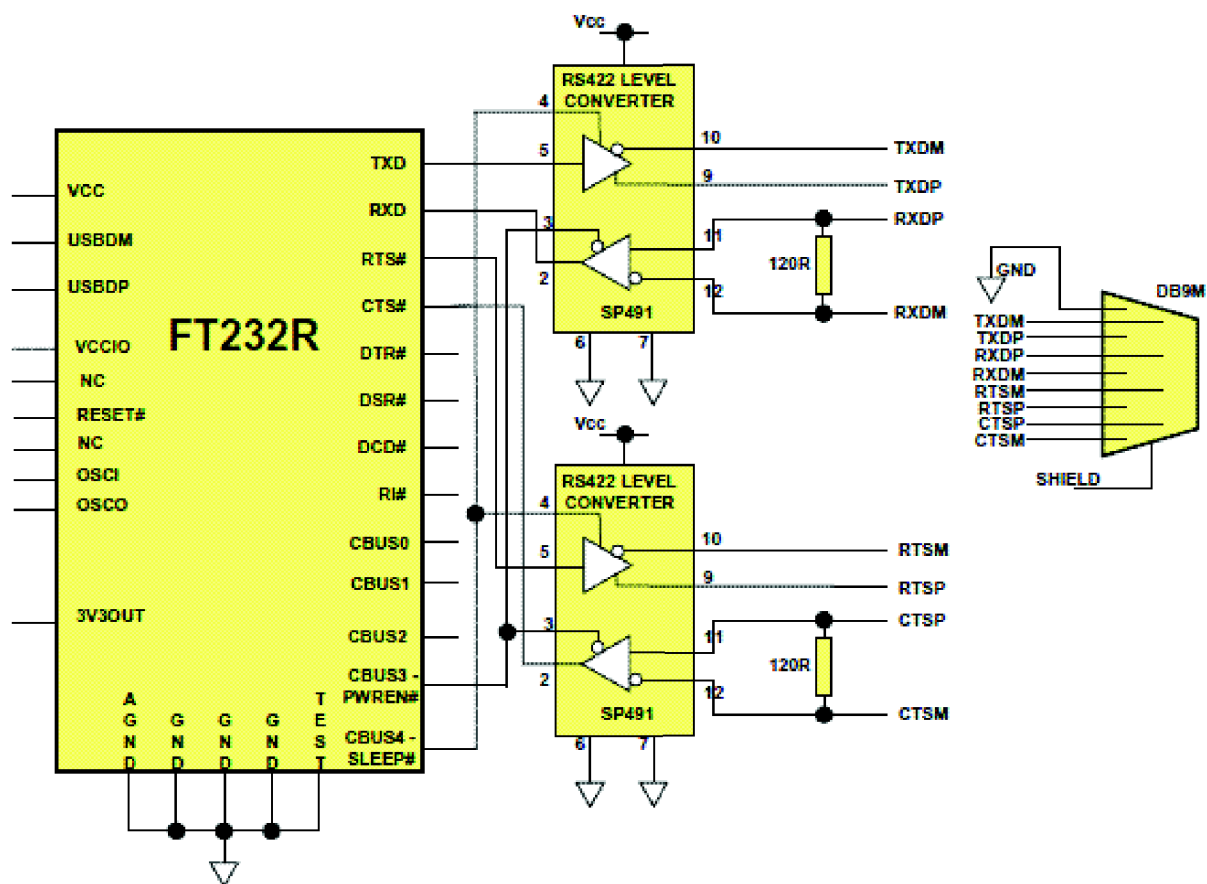


Obrázek 6.2: Napájení aplikace s vyšším odběrem z USB sběrnice. [4]



Obrázek 6.3: Zapojení indikačních LED. [4]

Při použití LED, které indikují příjem a vysílání dat je možné použít sběrnici CBUS a její signály TXLED# a RXLED#. V případě příliš vysoké spotřeby (jedna klasická LED odebírá přibližně 20mA) je možné použít pouze jednu LED a signál TX&RXLED#, který signalizuje jakoukoli komunikaci.



Obrázek 6.4: Zapojení převodníku USB-UART (RS-422). [4]

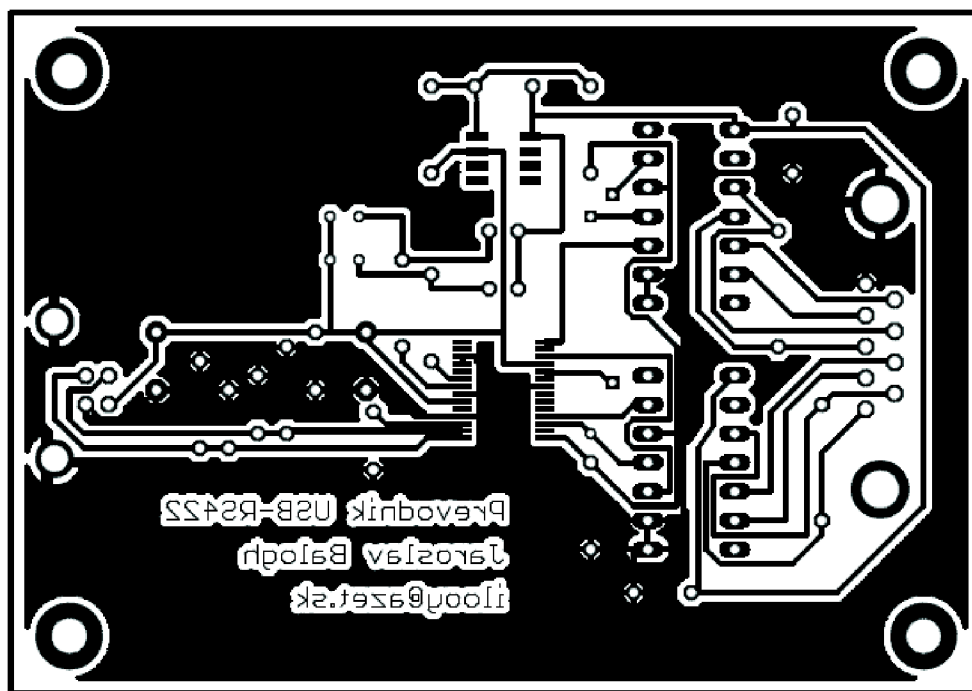
Na zapojení převodníku USB-UART (RS-422) vidíme použití dvou budičů SP491 (ekvivalent použitého LTC491CN), kde je řídicími signály PWREN# a SLEEP# řízena komunikace budičů. Vysílač i přijmač jsou aktivní, když zařízení pracuje a jsou nefunkční, když je zařízení v módu USB suspend. Také je výrobcem při napájení aplikace z USB doporučeno použití P-kanálového MOSFETu (viz obrázek 6.2), aby byla dodržena maximální spotřeba 500μA v režimu stand-by. [4]

# 7 Převodník USB – RS422

## 7.1 Návrh a popis převodníku USB – RS422 ve vývojovém prostředí Eagle

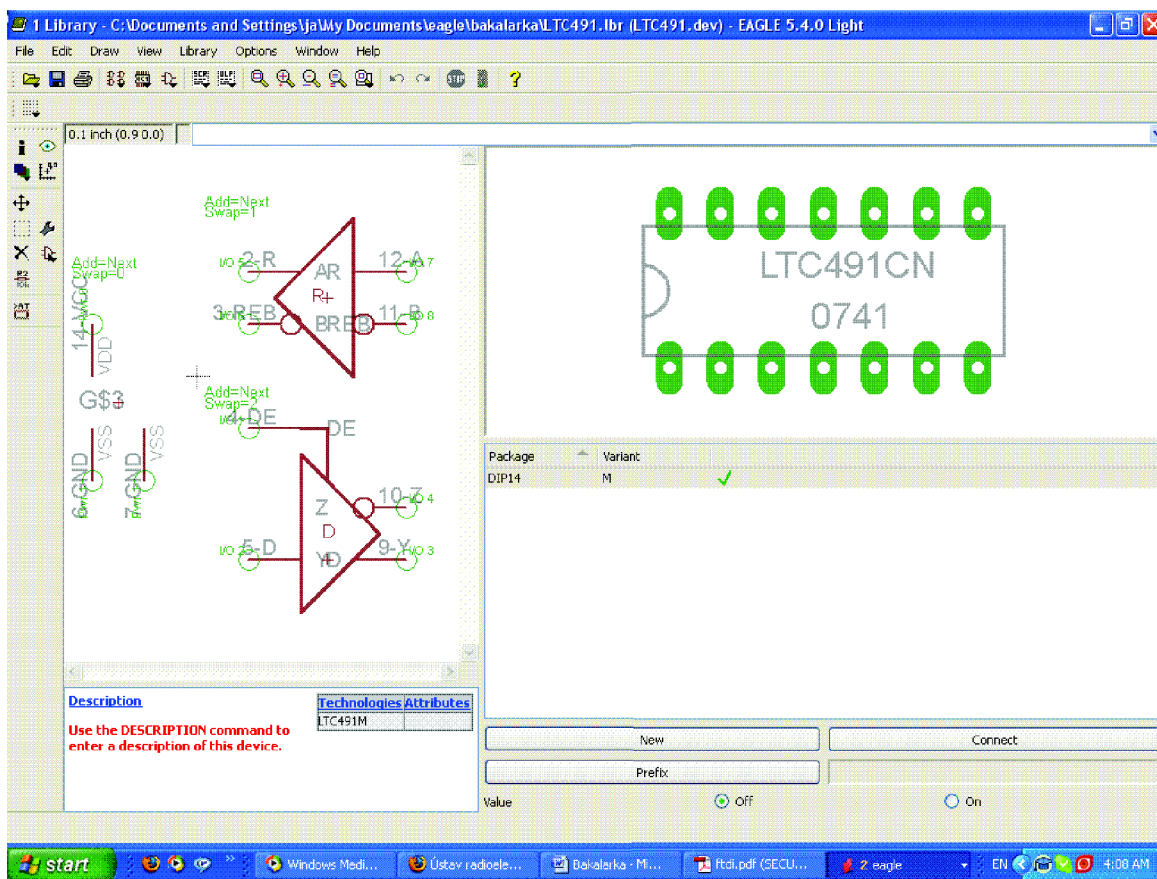
Jako realizovaná verze převodníku bylo zvoleno zapojení obvodu FT232R s napájením z USB, s řízením odběru externí logiky, dvojicí indikačních LED. Návrh schématu zapojení i desky plošného spoje (dále DPS) byl realizován pomocí freeware verze programu Eagle 5.4.0.

Před samotným návrhem byl proveden průzkum trhu, za účelem zjistit dostupnost součástek (hlavně budičů a obvodů FT232RL) a informace o pouzdech. Obvod FT232R se vyrábí ve dvou pouzdech SSOP-28 (označení FT232RL) a QFN-32 (označení FT232RQ). Dostupnější a levnější byla verze FT232RL, která má rozměry 5,3x10,2 mm, šířku kontaktu 0,3 mm a mezeru mezi kontakty 0,35 mm. Model této součástky je standardní součástí knihoven programu Eagle 5.4.0. Jako ekvivalent budiče SP491 byl v ČR (resp. Praze) dostupný pouze LTC491CN. Ten je dodáván v pouzdu DIL-14 ovšem jeho knihovna v programu Eagle 5.4.0 chybí a proto bylo nutné ji vytvořit (editor knihoven součástek je součástí programu). Použity byly dále konektory USB B (90°) a Cannon 9-pin 90°, několik filtračních kondenzátorů a pull-up rezistorů. V konečné podobě je DPS jednovrstvá a bylo nutné použít dvě drátové propojky.



Obrázek 7.1: Výkres plošných spojů desky převodníku USB-RS422.

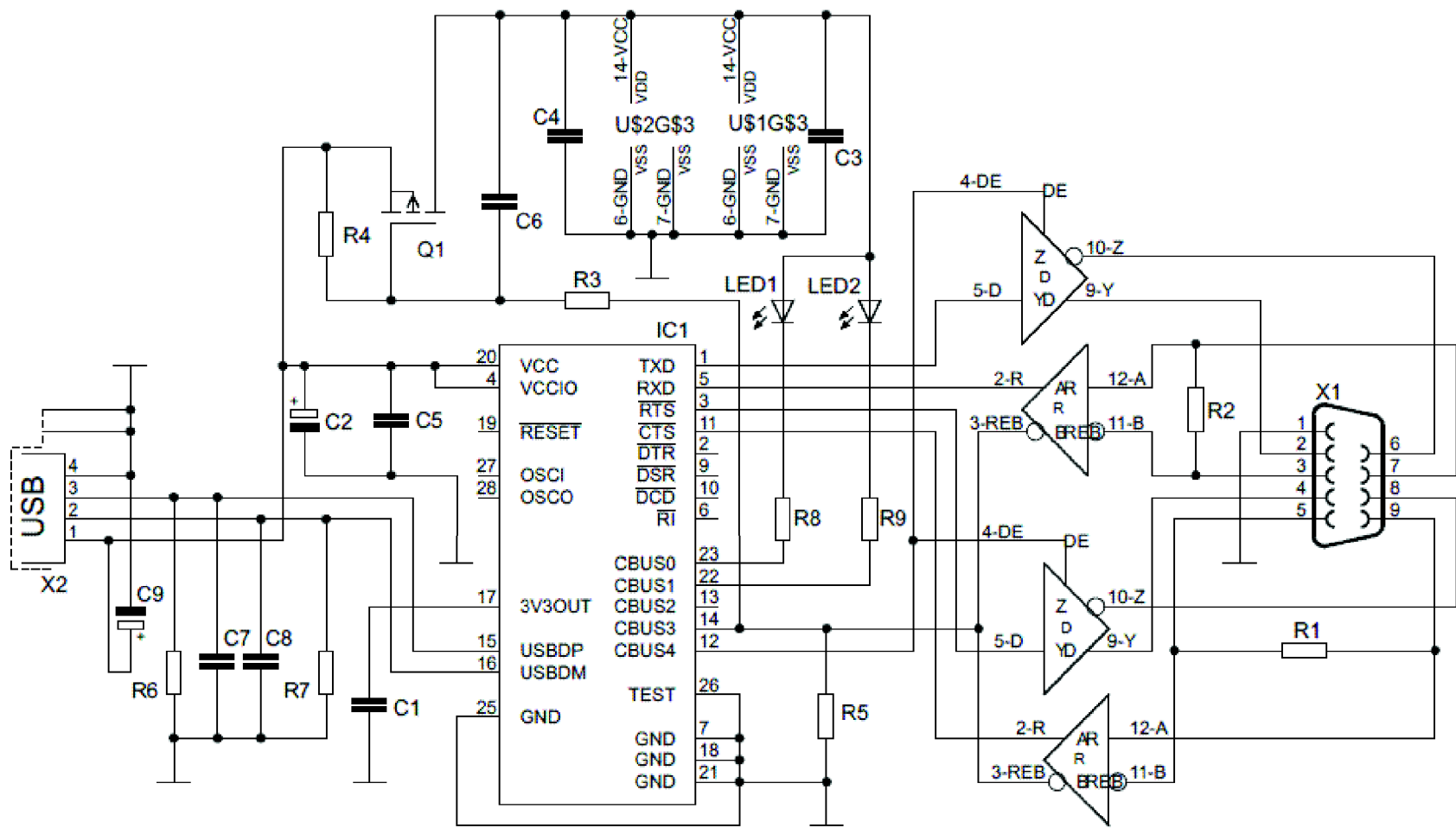




Obrázek 7.2: Tvorba knihovny LTC491CN v Eagle 5.4.0.

Hlavní součástí převodníku jsou integrované obvody FT232RL a LTC491CN. Tyto jsou při svých vstupech opatřeny 100nF kondenzátory na odstranění rušení. Obvod FT232RL má dále před napájením 4.7 $\mu$ F a 10nF kondenzátory na zabránění rušení z USB. Datové spoje z LTC491 jsou opatřeny pull-up rezistory, pro lepší tvarování signálů. U linek z USB je uváděno několik hodnot, rezistory mezi výstupy LTC491CN výrobce udává na přesné hodnotě 120 $\Omega$  a dokládá to (v datasheetu) průběhy výrazně zkreslených signálů při použití jiných hodnot odporů. Kondenzátory C7, C8 a rezistory R7, R8 osazeny nakonec nebyly, ale v případě rušení nebo skreslení datových datových signálů z USB je možno tento přidáním součástek vyfiltrovat. Další ochranné kondenzátory a rezistory jsou použity při zapojení P-kanálového MOSFETu (použit IRF7314PBF), aby bylo dosaženo co nejjemnějšího vypínání a zapínání periférií. Jsou použity konektory USB B (samice) a Cannon 9-pin (samec), aby byl dodržen obvyklý standard (USB typ B na připojovaném zařízení a Cannon 9-pin jako typický konektor sériového rozhraní).

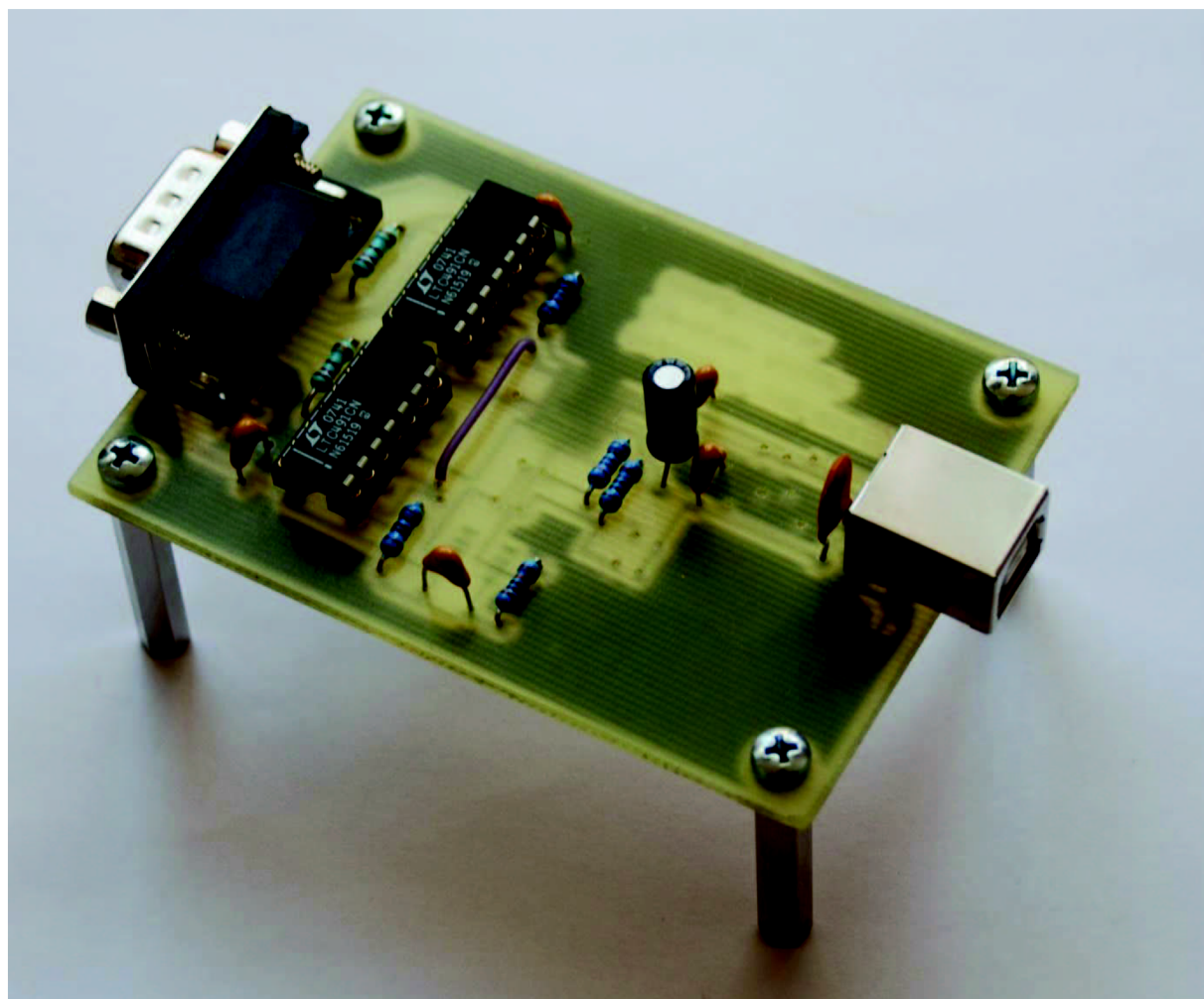
Celkově má hotový převodník trochu větší rozměry, než by bylo možné provést, ale jelikož se zejména kvůli budičům nevejde do pouzdra konektoru, je nutné jej umístit do ochranné krabičky a nejmenší dostupný rozměr na českém trhu je 10x7x4 cm. Délka spojů by však zůstala nezměněná i při menším provedení DPS vzhledem k velkému množství pasivních součástek. Na funkci tak větší rozměr DPS nemá vliv.



Obrázek 7.3: Schéma zapojení převodníku USB-RS422.

množství	hodnota	zařízení	součástky
2	120 $\Omega$	R-EU_0207/10	R1, R2
1	1k $\Omega$	R-EU_0204/7	R3
1	100k $\Omega$	R-EU_0204/7	R4
3	10k $\Omega$	R-EU_0204/7	R5, R6, R7
2	270 $\Omega$	R-EU_0204/7	R8, R9
5	100nF	C-EU050-024X044	C1, C3, C4, C5, C6
1	4.7 $\mu$ F	CPOL-EUE5-5	C2
2	47pF	C-EU050-024X044	C7, C8
1	10nF	CPOL-EUE5-5	C9
2	červená, zelená	LED3MM	LED1, LED2
1	IRF7314PBF	IRF7314PBF_SO8	Q1 (P-kanálový MOSFET)
1		F09HP	X1 (konektor Cannon 9-pin)
1	PN61729-S	PN61729-S	X2 (konektor USB B)
1	FT232RL	FT232RL	IC1
2	LTC491M	LTC491M	U\$1, U\$2

Tabulka 7.1: Seznam použitých součástek.



Obrázek 7.4: Převodník USB-RS422. [fotografie z archivu autora]



## 7.2 Modifikace výstupu pro pumpu RP-1

Vzhledem k tomu, že pumpa RP-1 využije pouze datové vstupy a výstupy rozhraní RS422, je nutné vybavit převodník také konektorem přizpůsobeným pro účely pumpy. Signály TXDM, TXDP, RXDM a RXDP standardizovaného konektoru RS422 je nutno připojit dle tabulky 7.2 ke konektoru pro pumpu RP-1. Toto propojení vychází z obrázku 2.3, který je výrobcem dodáván jako rozložení pinů.

Z důvodu uchování možnosti využít převodník USB-RS422 i k jiným účelům, než je řízení pumpy RP-1 a nevyhnutné nutnosti použití drátových propojek, byl k převodníku připojen další konektor typu Cannon 9-pin, který byl zapojen dle specifických požadavků pumpy. Toto zapojení také umožňuje použití standardního kabelu (cannon 9-pin – cannon 9-pin) bez rizika ztráty dalšího převodníku nebo nutnosti použít upravený kabel (s křížením vodičů). Na hotovém převodníku USB-RS422 jsou tak 3 konektory:

- USB typ B – pro připojení převodníku k PC
- Cannon 9-pin (samec) – pro připojení standardního rozhraní RS422 a možností plného využití funkcí RS422 (řízení toku dat a dalších)
- Cannon 9-pin (samec) – pro připojení pumpy RP-1

Všechny konektory jsou také popsány na krytu zařízení (plastová konstrukční krabička), aby nedošlo k chybnému připojení.

rozhraní <b>RS422</b> – strana převodníku (PC)		směr dat	rozhraní pumpy <b>RP-1</b> – strana pumpy	
číslo pinu na konektoru Cannon 9-pin	signál dle značení RS422		signál dle značení RP-1	číslo pinu na konektoru Cannon 9-pin
9	TXDM	=>	In -	3
4	TXDP	=>	In +	8
3	RXDM	<=	Out -	2
8	RXDP	<=	Out +	7

Tabulka 7.2: Propojení pinů standardu RS422 a pumpy RP-1. [1]

# 8 E<sup>2</sup>PROM

## 8.1 Paměť E<sup>2</sup>PROM

Obvod FT232RL má v sobě zabudovanou paměť E<sup>2</sup>PROM (=EEPROM), ve které jsou uloženy informace od výrobce, obsahující VID a PID, což je důležité při identifikaci zařízení v PC. Na základě těchto údajů je PC schopen komunikovat s daným zařízením a je možné jej jednoznačně identifikovat v případě připojení více podobných zařízení (v takovém případě je nutné ID přepsat, aby bylo každé jedinečné). Další možností, kterou paměť EEPROM poskytuje je zvýšení maximálního odběru proudu z USB. Od výrobce je nastavena maximální hodnota odběru pro zařízení 100mA a je možné ji zvýšit až na hodnotu 500mA. Nastavení popisu zařízení (zobrazuje se ve vyskakovacím okně při připojení zařízení k PC), povolení Pull-down vstupně/výstupních pinů v režimu USB Suspend a invertování řídicích signálů modemu UART jsou také možnosti konfigurace EEPROM.

## 8.2 Programování E<sup>2</sup>PROM

Výrobce FTDI doporučuje k programování paměti EEPROM použít program MProg. Nejnovější je verze MProg 3.5. Ovládání je intuitivní a vychází ze standardu Windows (například tlačítko lupa pro vyhledání zařízení a podobně). Tento program je schopen zjistit počet připojených zařízení a také informaci, která zařízení jsou naprogramována a která ne. Je také možno vytvořit profil, který obsahuje veškerá nastavení a je tedy možné stejně naprogramovat více zařízení bez nutnosti opakovaného nastavování jednotlivých parametrů. Program je univerzální a respektuje tak specifika všech obvodů firmy FTDI včetně FR232R (výběr se provádí v záložkách na pravé straně). Při programování EEPROM je nutné dodržet následující postup:

- vyhledání připojených zařízení
- upravení nastavení – změny parametrů, invertování signálů, povolení specifických stavů
- uložení nastavení do profilu
- vlastní naprogramování (ikona blesku)
- restart zařízení

Restart je nejlépe proveden odpojením obvodu FT232R od PC. Bez restartu zařízení nepracuje správně, zejména řídicí signály sběrnice CBUS zanáší značnou chybu. Například signál SLEEP# bez restartu místo úrovně log. 1 (5V) s vysokou frekvencí měnil úroveň na celém rozsahu (0-5V).

[14]

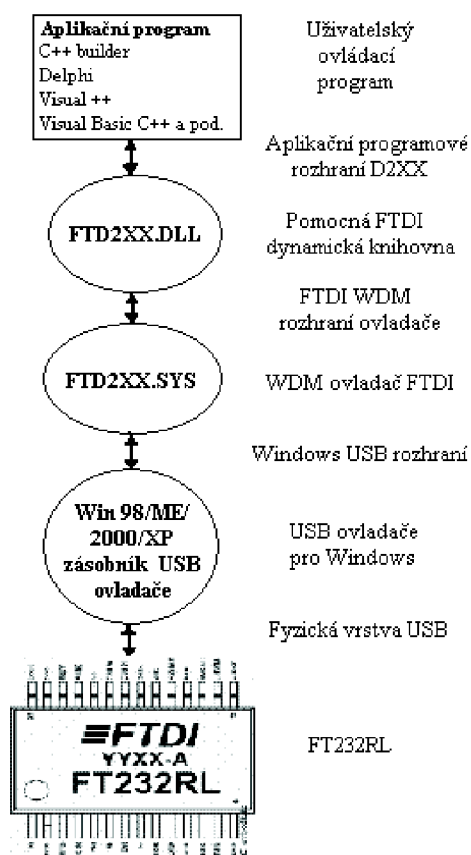
# 9 Programové řešení

## 9.1 Ovladač FTDI

Pro ovládání zařízení na bázi obvodu FT232RL lze použít ovladač CDM (aktuálně ve verzi CDM 2.04.16), který slučuje funkce jeho předchůdců VCP a D2XX.

VCP ovladače umožňují ovládat zařízení jako virtuální sériový port (zařízení je sice připojeno k USB, ale chápe se jako by bylo připojeno k sériovému portu).

Rozhraní D2XX je přímý ovladač pro Windows, který umožňuje aplikačnímu programu komunikovat s FT232RL pomocí kódu uloženého v dynamické knihovně (DLL). D2XX obsahuje WDM ovladač (FTD2XX.SYS), který komunikuje se zařízením přes Windows USB zásobník a DLL (FTD2XX.DLL), která je rozhraním pro aplikační programy vytvářené ve vývojových prostředích Visual C++, C++ Builder, Delphi, Visual Basic a dalších. [12]



Obrázek 9.1: *Architektura ovladače D2XX.* [2]

## Skupiny funkcí ovladače D2XX

- **Klasické rozhraní** seskupuje původní D2XX funkce, které jsou pro zpětnou kompatibilitu zachovány. Jedná se o funkce, které poskytují snadný přístup k USB zařízením.
- **Rozhraní E<sup>2</sup>PROM** dovoluje aplikaci číst/zapisovat do konfigurační E<sup>2</sup>PROM včetně možnosti použití volného prostoru pro uložení aplikačně specifických údajů.
- **FT232RL rozšíření** dovoluje plně využívat schopnosti 4. generace (režim BitBang, izochronní režim).
- **FT\_Win32 API** definuje funkce odpovídající původním Win32 API voláním pro práci se sériovým portem. Dávají tedy možnost snadno a rychle přejít od aplikace vyvinuté pro sériový port na aplikaci s obvodem FT232RL připojeným na USB sběrnici. Klasické rozhraní a FT\_Win32 API představují dva alternativní přístupy. Nedoporučuje se obě skupiny funkcí míchat.

## 9.2 Klasické funkce ovladače D2XX

Většina funkcí, které zajišťují informace o zařízení nebo s těmito informacemi pracují mají ve svém parametru hodnotu `dwFlags`. Tato hodnota určuje formát zjišťované informace. Její hodnoty jsou shrnuty v tabulce 9.1.

<code>dwFlags</code>	Význam
<code>FT_LIST_NUMBER_ONLY</code>	Zjistí počet připojených zařízení; <b>pvArg1</b> je interpretován jako ukazatel na proměnnou typu <b>DWORD</b> , kam se uloží počet připojených zařízení
<code>FT_OPEN_BY_SERIAL_NUMBER</code>	Zjistí sériové číslo zařízení
<code>FT_OPEN_BY_DESCRIPTION</code>	Zjistí popis výrobku
<code>FT_LIST_BY_INDEX</code>	Lze použít pro zjištění informačních řetězců pro jedno zařízení; <b>pvArg1</b> musí obsahovat pořadové číslo (index) zařízení (čísluje se od 0), <b>pvArg2</b> je interpretován jako adresa řetězce pro příjem informačních dat
<code>FT_LIST_ALL</code>	Lze použít pro zjištění informačních řetězců všech připojených zařízení; <b>pvArg1</b> ukazatel na pole adres řetězců pro načtení informačních dat (poslední adresa musí být <b>NULL</b> ), <b>pvArg2</b> počet položek pole odkázaného <b>pvArg1</b>

Tabulka 9.1: Možné hodnoty `dwFlags` a jejich význam. [2]

Před tím, než se začne se zařízením pracovat, musí se získat jeho informace a handle. K tomu slouží funkce `FT:ListDevice` pro zjištění aktuálních připojených zařízení a `FT_OpenEx`, která otevře spojení se zařízením a zjistí jeho handle.

### **FT\_ListDevices** (*PVOID pvArg1, PVOID pvArg2, DWORD dwFlags*)

Parametry pvArg1 a pvArg2 jsou závislé na dwFlags. V demonstračním programu obsahuje pvArg1 pořadové číslo zařízení, které je připojeno a pvArg2 obsahuje adresu řetězce, kam se uloží popis zařízení. Dle tohoto popisu následně otevírá připojení k zařízení. Parametr dwFlags určuje formát zjišťované informace.

### **FT\_OpenEx** (*PVOID pvAgr1, DWORD dwFlags, FT\_HANDLE \*ftHandle*)

Parametr pvArg1 určuje název zařízení, které chceme připojit, ale jeho význam závisí na hodnotě dwFlags. Parametr dwFlags určuje, z čeho se bude zařízení rozpoznávat dle jeho popisu uloženého v E<sup>2</sup>PROM paměti. Parametr ftHandle určuje adresu proměnné, do které se má uložit handle otevřeného zařízení.

Pro nastavení rychlosti a charakteristiky komunikace (parita, počet datových bitů a pod.) se používají funkce FT\_SetBaudRate a FT\_SetDataCharacteristics.

### **FT\_SetBaudRate** (*FT\_HANDLE ftHandle, DWORD dwBaudRate*)

Funkce FT\_SetBaudRate nastavuje přenosovou rychlost. Parametr ftHandle určuje handle zařízení, na kterém se má přenosová rychlost nastavit a dwBaudRate reprezentuje přenosovou rychlost v Baudech.

### **FT\_STATUS FT\_SetDataCharacteristics** (*FT\_HANDLE ftHandle, UCHAR uWordLength, UCHAR uStopBits, UCHAR uParity*)

Parametry funkce FT\_SetDataCharacteristic jsou parametr ftHandle, který reprezentuje handle daného zařízení, uWordLength značí délku znaku (FT\_BITS\_8 nebo FT\_BITS\_7), parametr uStopBits určuje počet stop-bitů (FT\_STOP\_BITS\_1 nebo FT\_STOP\_BITS\_2) a parametr uParity definuje používanou paritu možnosti jsou: FT\_PARITY\_NONE, FT\_PARITY\_ODD, FT\_PARITY\_EVEN, FT\_PARITY\_MARK nebo FT\_PARITY\_SPACE (nepoužitá, lichá, sudá, značená, mezerová)

Po navázání komunikace můžeme přistoupit k zápisu a čtení dat. Určeny jsou k tomu funkce FT\_Read a FT\_Write

### **FT\_Write** (*FT\_HANDLE ftHandle, LPVOID lpBuffer, DWORD dwBytes, LPDWORD lpdwBytesWritten*)

První parametr je handle zařízení. Druhý parametr je adresa bufferu zapisovaných dat. Proměnná dwBytes reprezentuje počet bajtů, které budeme posílat. lpdwBytesWritten je adresa proměnné typu DWORD pro příjem skutečného počtu zapsaných dat.

**FT\_Read** (*FT\_HANDLE* *ftHandle*, *LPVOID* *IpBuffer*, *DWORD* *dwBytes*, *LPDWORD* *lpdwBytesReturned*)

Tato funkce má obdobnou syntaxi jako funkce pro zápis. První parametr udává handle zařízení, druhý adresu bufferu pro příjem dat. *DwBytes* reprezentuje číslo, které udává kolik chceme načíst bajtů. Poslední parametr udává adresu proměnné typu *DWORD* pro příjem skutečného počtu přečtených bajtů.

**FT\_Close** (*FT\_HANDLE* *ftHandle*)

*FT\_Close* je funkce, která po uknočení komunikace slouží k uzavření spojení s FT232RL. Jediným parametrem je handle zařízení, s kterým chceme uknočit spojení.

Všechny zmíněné funkce vrací návratovou hodnotu typu *FT\_Status*. Jednotlivé významy jsou uvedeny v tabulce 9.2. [2] a [12]

Hodnota	Význam
FT_OK	operace úspěšně proběhla
FT_INVALID_HANDLE	neplatný handle zařízení
FT_DEVICE_NOT_FOUND	zařízení nenalezeno
FT_DEVICE_NOT_OPENED	zařízení neotevřeno
FT_IO_ERROR	Vstupně/výstupní operace selhala
FT_INSUFFICIENT_RESOURCES	nedostatečné systémové zdroje
FT_INVALID_PARAMETER	neplatný parametr operace
FT_INVALID_BAUD_RATE	neplatná přenosová rychlost
FT_DEVICE_NOT_OPENED_FOR_ERASE	není otevřeno pro smazání
FT_DEVICE_NOT_OPENED_FOR_WRITE	není otevřeno pro zápis
FT_FAILED_TO_WRITE_DEVICE	selhání zápisu
FT_EEPROM_READ_FAILED	selhání při čtení E <sup>2</sup> PROM
FT_EEPROM_WRITE_FAILED	selhání při zápisu do E <sup>2</sup> PROM
FT_EEPROM_ERASE_FAILED	selhání při mazání E <sup>2</sup> PROM
FT_EEPROM_NOT_PRESENT	E <sup>2</sup> PROM není připojena
FT_EEPROM_NOT_PROGRAMMED	E <sup>2</sup> PROM není naprogramována
FT_INVALID_ARGS	neplatný argument
FT_OTHER_ERRORS	jiná chyba

Tabulka 9.2: Možné hodnoty *FT\_STATUS*. [2] a [12]

### 9.3 Zdrojový kód ovládacího programu pumpy RP-1

K vytvoření programu k ovládání pumpy RP-1 bylo použito vývojové prostředí Borland C++ Builder 6.0, které nabízí použití programovacího jazyka C++ a také vytvoření grafického rozhraní ovládacího programu. Výsledkem je spustitelný .exe spouboř, který po spuštění otevře kontrolní panel s ovládacími tlačídky, které reprezentují stejné funkce, jaké poskytuje pumpa na svém předním panelu s výjimkou tlačítka STOP (tento příkaz rozhraní pro vzdálené ovládání pumpa nezná).

Pumpa Rainin Dynamax RP-1 má ve svém manuálu [1] (appendix B - strany 21-26) specifikovány požadavky na parametry přenosu. Konkrétně vyžaduje, aby komunikace probíhala podle standardů rozhraní RS422. K tomuto je použit převodník USB-RS422, popsáný v sedmé kapitole. Dle obrázku 2.3, na kterém je zapojení linek v portu pumpy je zřejmé, že zapojení neodpovídá standardu RS422. Pumpa pouze využívá datových vstupů a výstupů rozhraní RS422, nepoužije ovšem řídicí signály. Není nutné se v tomto případě zabývat řízením toku dat, které probíhá pomocí signálů modemu RTS# a CTS#. Softwarově je toto ošetřeno použitím příkazu `ftSetFlowControll` s parametrem NONE (žádná kontrola toku dat).

Rozpoznání převodníku, navázání komunikace a nastavení parametrů komunikace (přenosová rychlost 19200 Baud, 8 datových bitů, sudá parita a 1 stop-bit) je provedeno prostým použitím funkcí z podkapitoly 8.2 a proto je uvedeno pouze v příloze. Samotný průběh signálu potom vypadá následovně (pro signál TXDP): start bit (0), binární reprezentace znaku ASCII tabulky (8 bitů), paritní bit (součet 1 v datových bitech a parite musí být sudý = sudá parita) stop-bit (1).

Peristaltická pumpa reaguje a je řízena pomocí znaků ASCII tabulky. Jako první je nutné zaslat 255. znak, který je popsán jako disconnection code. Ten odpojí všechna připojená zařízení. Následuje číslo (ID) pumpy (výrobcem nastaveno na hodnotu 30) s nastaveným nejdůležitějším bitem na 1 – k číslu 30 se přičte 128, spolu teda 158. Praktická část příslušného zdrojového kódu je níže. Pro úplnost: proměnná *a* je deklarována jako znak (char) a přiřazením čísla se do něj uloží znak, který toto číslo (v dekadickém tvaru) reprezentuje v ASCII tabulce. Například dekadické číslo 10 (0a hexadecimálně, znak: kruh ve čtverečku) reprezentuje příkaz Line Feed (LF). [1] a [2]

Ukázka zdrojového kódu pro připojení pumpy RP-1:

```
a=255;
ftStatus = FT_Write(ftHandle,&a,velikost,&p); //zašle příslušný znak (odpojení) pumpě RP-1
sleep (20); // timeout 20 milisekund
a=158;
ftStatus = FT_Write(ftHandle,&a,velikost,&p); // zašle ID pumpy - připojí konkrétní pampu
```

První příkaz, který pumpa obdrží by měl být Lock (znak L - 76 desítkově), tedy příkaz na uzamčení komunikace. Vyřadí tím z funkce tlačítka na pumpě (kromě STOP) a pumpa je tak připravená přijímat další příkazy – je dálkově ovládána (Remote Control). Příkazy které provádí jakoukoli změnu jsou buffrované a je nutné dodržet rámeček LF-příkaz-CR, kde LF je Line Feed (10 dec.), příkaz je několik znaků reprezentovaných desítkovými čísly ASCII tabulky a CR je Carriage Return.

Ukázka zaslání příkazu Lock (znak L):

```

a=10; // Line Feed
ftStatus = FT_Write(ftHandle, &a, velikost, &p);
a=76; // 76. Znak ASCII je L – reprezentace příkazu Lock
ftStatus = FT_Write(ftHandle, &a, velikost, &p);
a=13; // Carrige Return
ftStatus = FT_Write(ftHandle, &a, velikost, &p);

```

Do pumpy je možné posílat dva druhy příkazů: buffered a immediate. Bafrovaný příkaz (viz příklad výše) se skládá z Line Feed (LF), samotného příkazu a Carrige Return (CR) a slouží ke změně stavu (rychlost, směr otáčení) pumpy. Pumpa jako odpověď na bafrovaný příkaz vrací LF, dále odpověď, ve které je aktuální nastavení měněného parametru a CR. Okamžitý příkaz (immediate) se zasílá pouze jako jeden znak (tedy bez LF a CR) a slouží pouze jako příkaz k odeslání dané informace o pumpě. Tabulka příkazů, které je možno pumpě zaslat, je níže.

příkaz	typ příkazu B/I	Funkce (změna stavu/vracené informace)	pumpa vrací informaci
<b>R</b> (Rnnnn)	B	nastavení nové rychlosti nnnn jsou čísla reprezentující stovky otáček za minutu R1250 nastaví rychlost 12.5 otáček za minutu	Rnnnn
<b>!</b> (!nn)	B	zaslání nového ID pumpy nn jsou čísla reprezentující nové ID	!nn
<b>J</b> jF nebo jB	B	nastavení směru otáčení jF (forward) nastaví otáčení po směru hodinových ručiček, jB protisměr	jF/jB
<b>L</b>	B	uzamkne klávesnici pumpy – povolení vzdáleného ovládání	L
<b>U</b>	B	otevře klávesnici pumpy – vypnutí vzdáleného ovládání	U
<b>R</b>	I	čtení displeje – „d“ reprezentující směr otáčení, XX.XX je nastavená rychlost, „c“ ovládání (klávesnice/vzdálené RS422/externí port ) a „a“ podmínka autostartu	dXX.XXca
<b>V</b>	I	stav analogového vstupu – „vvv“ reprezentují čísla 0 až 255 a ty reprezentují 0-5V na vstupu	vvv
<b>%</b>	I	identifikace pumpy – „x“ značí verzi	RP1V1.x
<b>?</b>	I	Stav požadavku – „c“ značí jak je pumpa ovládána, „e“ error status, „d“ směr otáčení a „s“ jestli se hlava otáčí nebo stojí	ceds

Tabulka 9.3: Seznam příkazů pro pumpu RP-1. [1]

Pumpa RP-1 nevyžaduje načítání zasláných odpovědí na příkazy, ale je vhodné tyto funkce využít pro zjištění momentálního stavu pumpy. K tomu je použito funkcí FT\_GetQueueStatus a FT\_Read. Funkce GetQueueStatus zjistí, kolik bajtů je v přijímacím zásobníku (buffer) obvodu FT232R a funkce FT\_Read následně tento počet znaků přečte a uloží do proměnné. Potom je lehká cesta k překódování dat (desítková hodnota) do ASCII znaku a podat informace uživateli. [1] a [2]



Ukázka načítání informací o pumpě zaskáním příkazu R (immediate):

```
FT_Purge(ftHandle,FT_PURGE_RX|FT_PURGE_TX); // vyprázdnění zásobníku
for (int i=1;i<=10;i++) //cyklus pro 10 zopakování
{
    a=82; //ASCII reprezentace znaku R
    ftStatus = FT_Write(ftHandle,&a,velikost,&p); // zaslání znaku R
    ftStatus = FT_GetQueueStatus(ftHandle,&RxBytes); // zjištění počtu bitů v zásobníku
    ftStatus = FT_Read(ftHandle,&InData,RxBytes,&w); // načtení zásobníku
    informace[i]=InData; // uložení informace do proměnné
    delay(); // fce. spoždění (nutné mezi zasláním znaku)
}
```

Pumpa po zaslání všech možných okamžitých (immediate) příkazů vrací zpět požadovanou proměnnou tak, že pošle její první znak a čeká na zaslání ASCII znaku ACK (06 hex.) nebo opětovné zaslání příkazu a poté vrací další znak. Proto je nutné použít cyklus (například odpověď na příkaz R je 10 znaků, proto cyklus proběhne 10-krát).

Tlačítko stop, které na pumpě slouží k jejímu zastavení zůstává aktivní i při vzdáleném ovládní (jako „panic-stop“ – stop v případě nouze) a pro pumpu RP-1 neexistuje příkaz se stejnou funkcí. Tlačítko stop v programu proto reprezentuje zaslání příkazu nastavujícího nulovou rychlost.

Další, podstatnější, problém nastal při zvyšování rychlosti o více než 20 otáček za minutu. Pumpa se při zaslání příkazu s výrazným zvýšením otáček zhruba v polovině případů z neznámého důvodu zasekla. Mohlo by tak dojít k poškození krokového motoru pumpy (ten při zaseknutí vydával nepříjemný zvuk). Tento problém byl v programu vyřešen postupným zvyšováním rychlosti - vždy o deset otáček. Změna rychlosti proto není okamžitá a její navýšení je delší o 25 ms při zvýšení o každých 10 otáček. Uživatel tak tento problém nemusí odhalit a ani řešit.

## 9.4 Program Rainin Dynamax RP-1 – control

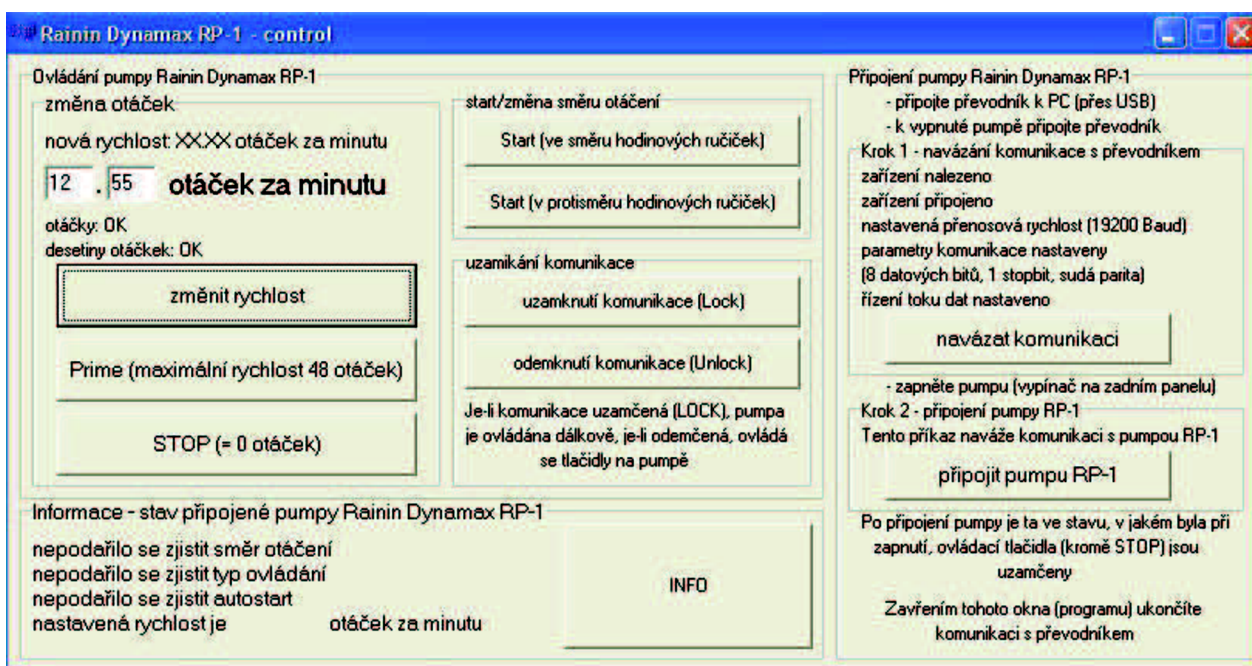
Program **Rainin Dynamax RP-1 – control** je konečná verze programu k ovládní peristaltické pumpy RP-1. Před použitím tohoto programu je nutné nainstalovat ovladač pro převodník (obvod FT232RL) - program CDM 2.04.16 nebo jeho novější verzi. Po spuštění programu Rainin Dynamax RP-1 – control je nejprve nutné připojit převodník a navázat komunikaci s pumpou (íkonky na pravé straně okna programu – viz obrázek 9.2).

Po připojení pumpy je možné používat stejné funkce, které je možno nastavit přímo na pumpě, tedy změnu rychlosti, změnu směru otáčení (tato funkce slouží v programu i na pumpě také ke startu otáčení), funkci PRIME (maximální otáčky 48 ot. Za minutu) a funkci STOP.

Nově vytvořená je funkce INFO, která i vzdálenému uživateli dodá informace o stavu pumpy. Další nové funkce jsou uzamčení a odemčení komunikace. Když se komunikace odemče, je možné pumpu ovládat její klávesnicí (uživatel programu má stále možnost použít INFO). Po zamčení komunikace se klávesnice pumpy stane neaktivní (krom bezpečnostního STOP tlačidla) a vzdálená obsluha má opět plnou kontrolu nad pumpou. V tomto případě není nutné opět navazovat komunikaci s pumpou.

Ukončení komunikace je provedeno v momentě zavření okna programu. Po opětovném spuštění programu Rainin Dynamax RP-1 – control je nutné komunikaci opět navázat.

Podrobný návod k obsluze programu je v příloze a také na CD, spolu s dalším softwarem (CDM, MProg), návody a technickou dokumentací.



Obrázek 9.2: Okno programu Rainin Dynamax RP-1 – control.

## 10 Závěr

Na základě informací o rozhraní USB, peristaltické pumpě **Rainin Dynamax RP-1** a možnosti, jak tyto propojit, bylo zvoleno technicky vyspělejší řešení v podobě využití modifikovaného rozhraní RS422, kterým pumpa disponuje.

Prakticky byl realizován převodník USB – RS422 s využitím specializovaného obvodu FT232R. Standardní výstup převodníku - rozhraní RS422 byl doplněn o další, modifikovaný pro pumpu RP-1. Zařízení si tak ponechalo svojí schopnost převodu dat USB do plnohodnotného formátu RS422 (i s řízením toku dat) a také speciální funkci pro ovládání pumpy. Dále byl v jazyce C++ vytvořen program pracující v prostředí Windows XP pro vzdálené ovládání pumpy RP-1. Tento program je schopen ovládat pumpu i bez přítomnosti obsluhy přímo u RP-1 a to má za následek její efektivnější využití.

Uplatnění najde tato práce při procesu analyzování DNA na Ústavu Biomedicínského Inženýrství při FEKT, VUT Brno. Je možné na ní také navázat, například dalším vylepšením ovládacího programu pro specifické požadavky pumpy Rainin Dynamax RP-1 i jiných zařízení, pracujících s rozhraním RS422.

# Seznam použité literatury

- [1] RAININ INSTRUMENT CO., Emmerville. Dynamax Model RP-1 Peristaltic Pump User Guide. 1996. 32 s.
- [2] MATOUŠEK, D. USB prakticky s obvody FTDI – 1. díl. Praha: Nakladatelství BEN – Technická Literatura, 2003. 270 s. Nakladatelství BEN – Technická Literatura. ISBN 80-7300-103-9.
- [3] MATOUŠEK, D. Praktické aplikace obvodu FT232RL – konvertor USB-UART. ARadio, 2009, ročník XIV, číslo 1., str. 3-39. ISSN 1211-3557
- [4] FTDI. Glasgow. FT232R datasheet. 2009. 29 s.
- [5] LINEAR TECHNOLOGY. Milpitas (CA). LTC491 datasheet. 1992. 12 s.
- [6] ŘEHÁK, J. USB - Universal Serial Bus - Popis rozhraní [online]. Praha: HW server s.r.o., 2002 - [cit. 7.5.2008]. Dostupné na WWW <<http://hw.cz/Teorie-a-praxe/Dokumentace/ART327-USB---Universal-Serial-Bus---Popis-rozhrani.html>>.
- [7] OLMR, V. HW server představuje - Sériová linka RS-232 [online]. Praha: HW server s.r.o., 2002 - [cit. 12.12.2005]. Dostupné na WWW <<http://hw.cz/rs-232#rs422>>.
- [8] SMITH, J. Peristaltic pump [online]. Florida: Wikimedia Foundation Inc., 2003 – [cit. 6.3.2008]. Dostupné na WWW <[http://en.wikipedia.org/wiki/Peristaltic\\_pump](http://en.wikipedia.org/wiki/Peristaltic_pump)>.
- [9] FTDI. Glasgow. FT232AM datasheet. 2007. 28 s.
- [10] ATMEL CORPORATION. San Jose. ATmega8 datasheet rev.5/08. 2008. 24 s.
- [11] INTERNATIONAL RECTIFIER. Kansas. IRF7314PbF datasheet. 2004. 7 s.
- [12] FTDI. Glasgow. D2XX Programmer's Guide. 2006. 129 s.
- [13] FTDI. Glasgow. USB to RS422 UART Serial Converter PCB datasheet. 2009. 15 s.
- [14] FTDI. Glasgow. MProg 3.0 Manual. 2006. 52 s.

# Seznam obrázků

Obrázek 2.1: Peristaltická pumpa Rainin Dynamax RP-1.	8
Obrázek 2.2: Ovládací panel preistaltické pumpy Rainin Dynamax RP-1.	10
Obrázek 2.3: Signálové linky portu pumpy RP-1.	11
Obrázek 2.4: Zadní strana těla pumpy RP-1.	12
Obrázek 2.5: Průběh navázání komunikace s pumpou RP-1.	12
Obrázek 2.6: Uspořádání pinů na svorkovnici (etrení terminál).	13
Obrázek 3.1: Provedení nevětvené linky RS-422.	16
Obrázek 4.1: Konektory USB typ „A“, USB typ „B“ a Cannon 9.	18
Obrázek 4.2: Uspořádání vodičů v kabelu USB.	19
Obrázek 4.3: Kód NRZI.	22
Obrázek 5.1: Schéma rozmístění pinů na FT232AM a ATmega8 v pouzdrech typu MLF.	24
Obrázek 6.1: Zjednodušené blokové schéma obvodu FT232R.	26
Obrázek 6.2: Napájení aplikace s vyšším odběrem z USB sběrnice.	28
Obrázek 6.3: Zapojení indikačních LED.	28
Obrázek 6.4: Zapojení převodníku USB-UART (RS-422).	29
Obrázek 7.1: Výkres plošných spojů desky převodníku USB-RS422.	30
Obrázek 7.2: Tvorba knihovny LTC491CN v Eagle 5.4.0.	31
Obrázek 7.3: Schéma zapojení převodníku USB-RS422.	32
Obrázek 7.4: Převodník USB-RS422.	33
Obrázek 9.1: Architektura ovladače D2XX.	36
Obrázek 9.2: Okno programu Rainin Dynamax RP-1 – control.	43

# Seznam tabulek

Tabulka 4.1: Popis (deskriptor) zařízení.	20
Tabulka 6.1: Dostupné signály sběrnice CBUS.	27
Tabulka 7.1: Seznam použitých součástek.	33
Tabulka 7.2: Propojení pinů standardu RS422 a pumpy RP-1.	34
Tabulka 9.1: Možné hodnoty dwFlags a jejich význam.	37
Tabulka 9.2: Možné hodnoty FT_STATUS.	39
Tabulka 9.3: Seznam příkazů pro pumpu RP-1.	41

# Seznam příloh

Příloha 1: Zdrojový kód programu k ovládání pumpy RP-1	52
Příloha 2: Návod k obsluze ovládacího programu	58

# Seznam příloh na CD

- bakalářská práce (ve formátu pdf)

## Složka *Dokumenty a datasheety*:

- datasheet FT232R, datasheet LTC491, datasheet IRF7314PBF, manuál pumpy Rainin Dynamax RP-1, datasheet převodníku USB-RS422
- návod na programování D2XX
- návody na naprogramování ovladače pumpy RP-1 (zaslané firmou Rainin)
- návod na obsluhu programu Mprog 3.0
- návod na instalaci ovladačů firmy FTDI (D2XX je jeho součástí)
- záznamy komunikace s firmou FTDI a Rainin

## Složka *D2XX*:

- instalační soubor programu CDM 2.04.16 (freeware)
- návod k instalaci CDM, návod k programování pomocí D2XX
- ukázky programů v D2XX
- záznam komunikace s firmou FTDI

## Složka *Rainin*:

- manuál k pumpě Rainin Dynamax RP-1
- návod na ovládací program pumpy (2x)
- záznam komunikace s firmou Rainin

## Složka *MProg*:

- instalační soubor k programu Mprog 3.5 (freeware)
- návod k použití programu Mprog 3.0

## Složka *Eagle*:

- knihovna součástky LTC491CN
- převodník USB-RS422 v1.0 (osazovací plán, seznam součástek, výkres plošných spojů)
- převodník USB-RS422 v2.0 (osazovací plán, seznam součástek, výkres plošných spojů, soubory eaglu .brd a .sch – schéma zapojení a návrh DPS)
- instalační soubor programu Eagle 5.04 (freeware)



Složka *Program*:

- spustitelný program Rainin Dynamax RP-1 - control ke vzdálené obsluze pumpy RP-1
- soubor readme.txt s popisem programu a návodem k obsluze
- projekt programu v C++ (pro případnou editaci)
- soubor zdrojovy\_kod.txt s kompletním zdrojovým kódem programu

# **Přílohy**

## Příloha 1: Zdrojový kód programu Rainin Dynamax RP-1 - control

```
#include <vc1.h> //deklarace hlavičkových souborů
#include <math.h>
#pragma hdrstop
#include <cstdlib>
#include <windows.h>
#include <iostream>
#include "Ftd2xx.h"
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
using namespace std;
int pom; //deklarace proměnných
char a;
AnsiString text;
FT_HANDLE ftHandle;
FT_STATUS ftStatus;
char popis[64];
unsigned char data[3];
DWORD p;
char OutData[255];
DWORD velikost;
DWORD w;
bool tim=0;
char rychlost[2];
char smerotaceni;
char ovladani;
char autostart;
char zmenarychlosti[4];

TForm1 *Form1;

void delay() //funkce pro timeout
{
Sleep(25); //25 ms timeout
}

void zaslani(char b, char c, char d, char e) //fce. zaslani prikazu zmeny rychlosti
{
FT_Purge(ftHandle,FT_PURGE_RX | FT_PURGE_TX); //vyčištění zásobníku
velikost = 1;
a=10; // LF
ftStatus = FT_Write(ftHandle,&a,velikost,&p);
a=82; //R
ftStatus = FT_Write(ftHandle,&a,velikost,&p);
ftStatus = FT_Write(ftHandle,&b,velikost,&p);
ftStatus = FT_Write(ftHandle,&c,velikost,&p);
ftStatus = FT_Write(ftHandle,&d,velikost,&p);
ftStatus = FT_Write(ftHandle,&e,velikost,&p);
a=13; //CR
ftStatus = FT_Write(ftHandle,&a,velikost,&p);
delay();
}
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
int binarni;
velikost = 1;
return ;
}
}
```

```

//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    velikost = 1;
    a=10; //LF
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=76;
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=13; //CR
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    delay();
    Form1->Button9->Click();
}
//-----
void __fastcall TForm1::close(TObject *Sender, TCloseAction &Action)
{
    ftStatus = FT_Close(ftHandle);
}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    a=10;
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=85; //Unlock
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=13;
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    delay();
    Form1->Button9->Click();
}
//-----
void __fastcall TForm1::Button4Click(TObject *Sender)
{
    a=10; // LF
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=106; //j
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=70; //F (ve smeru hodin)
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=13; //CR
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    delay();
    Form1->Button9->Click();
    Form1->Button9->Click();
}
//-----

void __fastcall TForm1::Button5Click(TObject *Sender)
{
    ftStatus=FT_ListDevices(0,popis,FT_LIST_BY_INDEX|FT_OPEN_BY_DESCRIPTION);
    if(ftStatus!=FT_OK)
    {
        Label2->Caption= "zařizení nebylo nalezeno";
        system("PAUSE");
        return ;
    }else
    {
        Label2->Caption= "zařizení nalezeno";
    }
    ftStatus=FT_OpenEx(popis,FT_OPEN_BY_DESCRIPTION,&ftHandle);
    if(ftStatus!=FT_OK)
    {
        Label3->Caption= "zařizení se nepodařilo připojit";
        system("PAUSE");
        return ;
    }else
    {
        Label3->Caption= "zařizení připojeno";
    }
    ftStatus=FT_SetBaudRate (ftHandle, FT_BAUD_19200); //pumpa ma 19200 baud
    if(ftStatus!=FT_OK)
    {
        Label4->Caption= "přenosová rychlost nenastavena";
        system("PAUSE");
    }
}

```

```

        return ;
    } else
        Label4->Caption= "nastavená přenosová rychlost (19200 Baud)";
ftStatus=FT_SetDataCharacteristics (ftHandle, FT_BITS_8, FT_STOP_BITS_1, FT_PARITY_EVEN);
if(ftStatus!=FT_OK)
    {
        Label5->Caption= "nepodařilo se nastavit parametry komunikace";
        system("PAUSE");
        return ;
    } else
        Label5->Caption= "parametry komunikace nastaveny";
ftStatus=FT_SetFlowControl (ftHandle, FT_FLOW_NONE,0,0);
if(ftStatus!=FT_OK)
    {
        Label6->Caption= "řízení toku dat nenastaveno";
        system("PAUSE");
        return ;
    } else
        Label6->Caption= "řízení toku dat nastaveno";
        return ;
    }
}
//-----
void __fastcall TForm1::Button6Click(TObject *Sender)
{
    DWORD RxBytes=0;
    char InData[5]="0";
    velikost = 1;
    a=255; //odpojí všechna zařízení (disconnection code)
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=158; // ID pumpy - připojí konkrétní pumpu
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    ftStatus = FT_GetQueueStatus(ftHandle,&RxBytes);
    ftStatus = FT_Read(ftHandle,&InData,RxBytes,&p);
    Label1->Caption =RxBytes;
    a=10; //LF
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=76; //Lock - uzamkne komunikaci
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=13; //CR
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    delay();
    Form1->Button9->Click();
}
//-----
void __fastcall TForm1::Button7Click(TObject *Sender)
{
    a=10; // LF
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=106; //j
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=66; // B - protisměr (Backward)
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    a=13; //CR
    ftStatus = FT_Write(ftHandle,&a,velikost,&p);
    delay();
    Form1->Button9->Click();
    Form1->Button9->Click();
}
//-----
void __fastcall TForm1::Button8Click(TObject *Sender)
{
    zaslani (49,48,48,48); //R1000 //funkce PRIME
    zaslani (50,48,48,48); //R2000
    zaslani (51,48,48,48); //R3000
    zaslani (52,48,48,48); //R4000
    zaslani (53,56,48,48); //R4800
    Form1->Button9->Click();
}
//-----
void __fastcall TForm1::Button9Click(TObject *Sender)

```

```

{
DWORD RxBytes=0;
char InData;
char informace[10]="0";
velikost = 1;
FT_Purge(ftHandle,FT_PURGE_RX | FT_PURGE_TX); // vyprazdneni zasobniku
for (int i=1;i<=10;i++)
{
a=82; // R
ftStatus = FT_Write(ftHandle,&a,velikost,&p);
ftStatus = FT_GetQueueStatus(ftHandle,&RxBytes);
ftStatus = FT_Read(ftHandle,&InData,RxBytes,&w);
Memo1->Lines->Add(InData);
informace[i]=InData;
delay();
}
FT_Purge(ftHandle,FT_PURGE_RX | FT_PURGE_TX); // vyprazdneni zasobniku
smerotaceni=informace[2];
//Memo1->Lines->Add(smerotaceni);
switch (smerotaceni)
{
case '+': Label8->Caption="pumpa se točí VE SMĚRU hodinových ručiček";
break;
case '-': Label8->Caption="pumpa se točí V PROTISMĚRU hodinových ručiček";
break;
case ' ': Label8->Caption="pumpa se NETOČÍ, směr otáčení není zadán";
break;
default: Label8->Caption="nepodařilo se zjistit směr otáčení";
break;
}
ovladani=informace[8];
Memo1->Lines->Add(ovladani);
switch (ovladani)
{
case 'K': Label9->Caption="pumpa se ovládá KLÁVESNICÍ (stiskem tlačítka na pumpě)";
break;
case 'R': Label9->Caption="pumpa se ovládá VZDÁLENĚ (tímto programem)";
break;
default: Label9->Caption="nepodařilo se zjistit typ ovládání";
break;
}
autostart=informace[10];
Memo1->Lines->Add(autostart);
switch (autostart)
{
case '*': Label10->Caption="je nastaven autostart ( displej bliká)";
break;
case ' ': Label10->Caption="autostart není nastaven";
break;
default: Label10->Caption="nepodařilo se zjistit autostart";
break;
}
rychlost[0]=informace[3];
rychlost[1]=informace[4];
rychlost[2]='.';
rychlost[3]=informace[6];
rychlost[4]=informace[7];
Memo1->Lines->Add(rychlost);
Label11->Caption=rychlost;
velikost = 1;
a=10; //LF
ftStatus = FT_Write(ftHandle,&a,velikost,&p);
a=76;
ftStatus = FT_Write(ftHandle,&a,velikost,&p);
a=13; //CR
ftStatus = FT_Write(ftHandle,&a,velikost,&p);
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{

```

```

zaslani (48,48,48,48);           //příkaz STOP
Form1->Button9->Click();
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    tim=1;
}
//-----
void __fastcall TForm1::Button13Click(TObject *Sender)
{
    char w,x,y,z;           //načtení zadané rychlosti
    bool uspech1;
    bool uspech2;
    int A;
    int B;
    uspech1=TryStrToInt(Edit1->Text,A); //konverze znaku na číslo
    uspech2=TryStrToInt(Edit2->Text,B); //konverze znaku na číslo
    if (uspech1)
    {
        if (uspech2)
        {
            if (A>48)           //podmínka: je-li číslo moc velké, tak...
            {
                Label12->Caption="otáčky: chci kladné číslo od 0 do 48";
                ShowMessage ("Chyba při zadávání otáček!");
            }
            else           //postupne zvyšování otacek
            {
                Label12->Caption="otáčky: OK";
                if (A<10)           //ošetření: zadáno 1 číslo (jednotky otáček)
                {
                    w=48;
                    x=A+48;
                }
                else           //ošetření: zadány 2 čísla (desítky otáček)
                {
                    int C=A/10;
                    w=C+48;           //přepočítání čísla na označení znaku ASCII
                    int D=A-(C*10);
                    x=D+48;           //přepočítání čísla na označení znaku ASCII
                }
            }
            if (B>99)           //podmínka: je-li číslo moc velké, tak...
            {
                Label13->Caption="otáčky: chci 2 desetinná místa (číslo od 0 do 99)";
                ShowMessage ("Chyba při zadávání otáček!");
            }
            else           //je-li vhodné
            {
                Label13->Caption="desetiny otáček: OK";
                int B_length=Edit2->Text.Length();
                if (B_length<2)           //ošetření: zadáno 1 číslo (jednotky otáček)
                {
                    y=B+48;
                    z=48;
                }
                else           //ošetření: zadány 2 čísla (desítky otáček)
                {
                    int C=B/10;
                    y=C+48;           //přepočítání čísla na označení znaku ASCII
                    int D=B-(C*10);
                    z=D+48;           //přepočítání čísla na označení znaku ASCII
                    if (A>10)           //postupne zvyšování rychlosti (po 10ti otáčkách)
                    {
                        zaslani (49,48,48,48);
                    }
                }
                if (A>20)
                {
                    zaslani (50,48,48,48);
                }
            }
        }
    }
}

```

```

if (A>30)
{
zaslani (51,48,48,48);
}
if (A>40)
{
zaslani (52,48,48,48);
}
zaslani(w, x, y, z); //zaslani nove rychlosti
}
}
}
}
else
{
Label13->Caption="desetiny otáček: zadej mi číslo, né znak";
ShowMessage ("Chyba při zadávání otáček!");
}
}
else
{
Label12->Caption="otáčky: zadej mi číslo, né znak";
ShowMessage ("Chyba při zadávání otáček!");
}
Form1->Button9->Click();
}

```



## **Příloha 2: Návod k obsluze programu Rainin Dynamax RP-1 - control**

1. nainstalujte ovladač pro převodník FT232RL – program **CDM 2.04.16** nebo vyšší
2. spustěte program **Rainin Dynamax RP-1 - control** k ovládání pumpy
3. připojte modifikovaný **převodník USB-RS422** k PC
4. připojte modifikovaný převodník USB-RS422 k vypnuté peristaltické pumpě **Rainin Dynamax RP-1**
5. **navazte komunikaci** s modifikovaným převodníkem USB-RS422 (ikona v pravém menu programu)
6. **zapněte** peristaltickou pumpu Rainin Dynamax RP-1 (na zadním panelu)
7. **navazte komunikaci** s peristaltickou pumpou Rainin Dynamax RP-1 (ikona v pravém menu programu)
8. je-li vše v pořádku, je uživatel informován popiskou v pravém menu programu
9. nyní můžete **ovládat** peristaltickou pumpu Rainin Dynamax RP-1, již byla komunikace zamčena (tlačítka pumpy nereagují)
10. je-li komunikace odemčena (ikonou v programu nebo stiskem tlačítka STOP na pumpě), je nutné opět zaslat příkaz na zamčení komunikace (v programu)
11. platí bod 8 a nastavení provedené chronologicky (byl-li manuálně změněn stav (například směr otáčení), tak tento stav trvá i po opětovném uzamčení komunikace)
12. komunikace s pumpou RP-1 je **ukročena** v momentě zavření okna programu, pumpa zůstává ve stavu, jaký byl nastaven před ukončením komunikace a převodník USB-RS422 je odpojen (je možné jej odpojit od PC)